Coastal Navigation – Mobile Robot Navigation with Uncertainty in Dynamic Environments

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

September 30th, 1998

Abstract

Ships often use the coasts of continents for navigation in the absence of better tools such as GPS, since being close to 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 generating trajectories that take into account both the information content of the environment, and the density of the people in the environment. These trajectories reduce the average positional certainty as the robot moves, reducing the likelihood the robot will become lost at any point. Our method was successfully implemented and used by the mobile robot Minerva, a museum tourguide robot, for a 2 week period in the Smithsonian National Museum of American History.

1 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 [MD94, DZ96, SD98, FBT98, KCK96].

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 ceilings. 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.

Our solution to these problems is inspired by 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[1] consists of the following:

- Modelling the information content of the environment. The model accounts both for sensor limitations and unmodelled, dynamic obstacles.
- Planning trajectories that account for the information content model of the environment and obstacle information in the map, respecting 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

[1] The term "coastal navigation was suggested by Thomas Christaller in a private communication about the work reported here.

of combining the information content with the path planner to generate plans that reduce the expected localization error. Finally we show experimental results.

The framework that we use for navigation is a probabilistic one. Figure 1 shows an overhead (bird's-eye view) of an example environment. This map is a probabilistic occupancy grid [ME85, Elf90], 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.

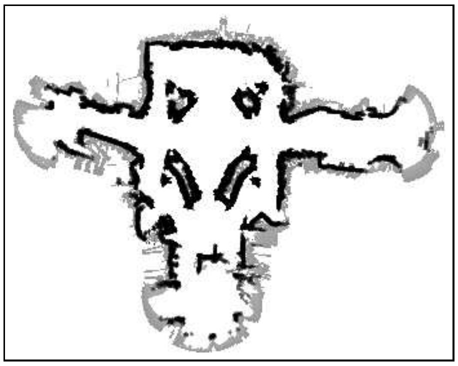


Figure 1: An example map of the National Museum of American History. The white areas correspond to open space, and the black areas are walls, or occupied space. The size of this map is 53m by 67m

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 [BCF+98] 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 sensors used to generate this map were two SICK 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.

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 primary result of this paper is that we were able to reduce the average positional uncertainty on a real robot, in a large, open and extremely dynamic environment (the museum). We will show experimental results for trajectories in the museum, including a detailed case of a specific trajectory. The coastal navigator in fact was in successful operation in the adverse conditions of the museum, over a long-term period of two weeks.

2 Previous Work

Developing motion planning algorithms based on positional uncertainty is not a new idea. Erdmann developed motion planning strategies with uncertainty [Erd84, Erd86] and probabilistic strategies. Nourbakhsh and colleagues [NPB95] developed probabilistic navigation techniques on DERVISH, a robot similar in many respects to Minerva.

Considerable work in in the field of partially observable Markov decision processes (POMDPs) [CKL94, KS96] 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. Markov localization, however, has been used successfully on multiple robot platforms [KCK96, FBT98].

Work has been done on trajectory generation with respect to positional uncertainty; Takeda et al. [TFL94] 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.

3 Modelling Information Content

The motivation for coastal navigation is generating trajectories for the mobile robot that reduce the likelihood of localization error. For example, localization error can arise when a mobile robot follows a path through a wide-open space, such as outdoors, or in a very large or crowded room. In these cases, all reference points are outside the range of the sensors, or even blocked. Therefore, the likelihood of the robot becoming lost as it moves through the open or crowded space is high.

Our solution is to have the robot identify how good regions of space are for localization, and alter the path planning accordingly. We have developed a planner that uses 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.

We first develop the general principles of the localization method and the information model in a statistical framework, before presenting the actual implementation of the algorithm.

3.1 Statistical Framework

The position, $\mathbf{x}$, of the robot is given as the location (x, y) and direction θ, defined over a space $\mathbf{X} = (X, Y, \Theta)$. Our localization method is a grid-based implementation of Markov localization [FBT98, KCK96]. This method represents the robot's belief in its current position using a 3-dimensional grid over $\mathbf{X} = (X, Y, \Theta)$, which allows for a discrete approximation of arbitrary probability distributions. The probability that the robot has a particular pose $\mathbf{x}$ is given by the probability $p(\mathbf{x})$.

Markov Localization

Let the robot's position be given by the initial probability distribution, $P_{\mathbf{x}}$, defined over $\mathbf{X} = (X, Y, \Theta)$. The robot acquires a sensor measurement s, for example a set of range data from a laser sensor. The localization process takes $P_{\mathbf{X}}$ and the sensor data, s, and returns the posterior probability distribution $P_{\mathbf{X}|\mathrm{s}} = p(\mathbf{x}|\,\mathrm{s})$, again defined over the space of poses of the robot, (X, Y, Θ).

The probability $p(\mathbf{x}|\,\mathrm{s})$ is given by Bayes' Rule:

$$p(\mathbf{x}|\,\mathrm{s}) = \frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})} \tag{1}$$

where $p(\mathbf{x})$ is the position distribution, and $p(\mathrm{s}|\,\mathbf{x})$ is computed from the sensor model and the robot's map of the environment.

The term $p(\mathrm{s})$ is the likelihood of observing sensor data s, and is computed from the prior position distribution, the sensor model and the environmental map.

$$p(\mathrm{s}) = \int_{\mathbf{X}} p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{2}$$

Entropy Computation

The entropy, $H(P_{\mathbf{X}})$, of a probability function, $P_{\mathbf{X}}$, provides a good measure of the certainty with which the robot is localized. The entropy of a probability distribution, $P_{\mathbf{X}}$, is computed over the space of all possible poses (X, Y, Θ) and is defined as:

$$H(P_{\mathbf{X}}) = -\int_{\mathbf{X}} p(\mathbf{x}) \log(p(\mathbf{x}))\, d\mathbf{x} \tag{3}$$

This measure can be considered as the "purity" of the probability distribution. If the distribution is highly focused at a single pose ($\mathbf{x} = x, y, \theta$), then the entropy will be low. If the distribution is spread over a wide space, then the entropy will be high. The effect that a particular set of sensing data has on the robot's belief in its position can therefore be measured in this way. Combining equations (1) and (3) gives the entropy of the posterior distribution after sensing:

$$\begin{aligned} H(P_{\mathbf{X}|\mathrm{s}}) &= -\int_{\mathbf{X}} p(\mathbf{x}|\,\mathrm{s}) \log(p(\mathbf{x}|\,\mathrm{s}))\, d\mathbf{x} \\ &= -\int_{\mathbf{X}} \frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})} \log\left(\frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})}\right) d\mathbf{x} \end{aligned} \tag{4}$$

Equation (4) gives the entropy of the posterior distribution, given a particular set of sensor measurements. Recall that $p(\mathbf{x})$ is the prior position distribution, $p(\mathrm{s}|\,\mathbf{x})$ is the probability of the sensor measurement conditioned on the position, computed from the sensor model and the environmental map. $p(\mathrm{s})$ is the prior distribution of the sensor measurement, given by equation (2).

A particular location in the environment can result, with different probability, in different sensor measurements. The entropy is therefore averaged from all possible sensor measurements, s, where each term is weighted by the likelihood of the sensor measurement. Equation (5) computes the expected value of the entropy over all posterior distributions P.

$$\begin{aligned} E(H(P_{\mathbf{X}|\mathbf{S}})) &= -\int_{\mathbf{S}} p(\mathrm{s}) \int_{\mathbf{X}} p(\mathbf{x}|\,\mathrm{s}) \log(p(\mathbf{x}|\,\mathrm{s})) d\mathbf{x} d\mathrm{s} \\ &= -\int_{\mathbf{S}} p(\mathrm{s}) \int_{\mathbf{X}} \frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})} \log\left(\frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})}\right) d\mathbf{x} d\mathrm{s} \\ &= -\int_{\mathbf{S}} \int_{\mathbf{X}} p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x}) \log\left(\frac{p(\mathrm{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathrm{s})}\right) d\mathbf{x} d\mathrm{s} \end{aligned} \tag{5}$$

$E(H(P_{\mathbf{X}|\mathbf{S}}))$ is the expected value of the entropy after firing the sensors, computed over all possible sensor measurements, given the initial position distribution $P_{\mathbf{X}}$.

For a particular pose distribution, we can compute the information content, I, of the robot's current position by computing the difference between the expected entropy of the positional probability conditioned on the sensor measurement, $E(H(P_{\mathbf{X}|\mathbf{s}}))$ and the entropy of the prior position distribution, $H(P_{\mathbf{X}})$:

$$I = E(H(P_{\mathbf{X}|\mathbf{S}})) - H(P_{\mathbf{X}}) \tag{6}$$

Note that equation (7) inverts the intuitive sense of information content; the higher the quantity I, the lower the information content.

Recall that the goal of measuring the information content of the environment, is to be able to construct a map of areas of the environment that have low and high information. The algorithm for constructing this map is the following procedure:

1. For each position $\mathbf{x}$ and initial probability distribution $P_{\mathbf{X}}$, generate all possible sensor measurements, $\mathbf{s}$ and the probability of these estimates, $p(\mathbf{s})$ as in equation (2).
2. For each sensor measurement $\mathbf{s}$, compute the entropy of the posterior probability distribution given by Markov localization as in equation (4)
3. Compute the expected value of the entropy as in equation (5), and take the difference from the initial entropy, as in equation (6).

In the above analysis, we have ignored the issue of the prior probability distribution of the robot's position. The entropy computation is heavily dependent on the robot's prior belief in its position, $p(\mathbf{x})$. Modeling robot navigation as a partially observable Markov decision process, or POMDP, would be one method for handling this dependency [CKL94]. However, the POMDP requires examining all possible prior probability beliefs and also all possible paths leading up to the prior probability belief. This process provides extremely accurate characterization of uncertainty. However, for planning, the computation is intractable, as it is exponential in the size of the environment. We have therefore made some simplifying assumptions in the implementation of the algorithm which dramatically reduce the complexity. One such simplification allows us to ignore the problem of the prior positional distribution.

3.2 Implementation

The first simplification we make immediately is to use a tracking assumption. The mobile robot has internal odometry which, while insufficient for long-term position estimation (as discussed), provides a reasonable estimate on the bounds and type of uncertainty of the robot position. Because the robot tracks its position using its odometry, we can in fact assume a Gaussian prior probability distribution centered at the assumed location of the robot, (x, y, θ), and limited to a small region of the environment. The Gaussian nature of the distribution is a result of the kinds of error that accumulate using odometry. It is this simplification that makes our POMDP-style approach tractable.

Also, since we use the same initial distribution at every point in the environment, it is not necessary to compute the change in entropy as in equation (6), but instead simply to compute the new entropy estimate.

We do not simulate every possible sensor measurement $\mathbf{s}$, but instead sample the sensor space, choosing only the most likely sensor data sets for a particular position. The samples we choose are maximum-likelihood sensor measurements, simulated using the map and the sensor model, from the cells in the 3×3 square centered at the assumed location of the robot. This particular sampling method has the effect of attributing all sensor errors to positional error in the robot – we assume that every sensor measurement corresponds correctly to some feature in the environment.

Furthermore, we do not in fact have a continuous distribution for the position of the robot, but a discrete grid. We also do not generate every possible sensor measurement, but use a sampling of the sensor measurement space. This reduces the integration into a summation in equation (5).

Finally, we can use the fact that our particular robot has $360°$ field of view, to eliminate the dependence on θ. Because the robot can see around its circumference completely, there is no positional certainty dependent on its orientation – it always knows its orientation exactly, given its location (x, y). Since the entropy is independent of θ in this case, we no longer need to integrate over it. It is important to note that in general, if the robot does not have rotationally-invariant perceptions, then θ cannot be ignored; indeed, coastal navigation is not very helpful if the ship is always pointed out to sea.

The above simplifications change the information content into the following equation:

$$I(x, y) = -\sum_{\mathbf{S}} \sum_{\mathbf{X}} p(\mathbf{s}|\,\mathbf{x})p(\mathbf{x}) \log\left(\frac{p(\mathbf{s}|\,\mathbf{x})p(\mathbf{x})}{p(\mathbf{s})}\right) \tag{7}$$

Figure 3 shows an example map of the 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.

4 Dynamic Environments

Entropy as described above is useful for determining the information content of a particular point in the environment, however the model assumes a static environment. In a dynamic environment, the data gathered by the sensors can be corrupted, for example by people blocking the

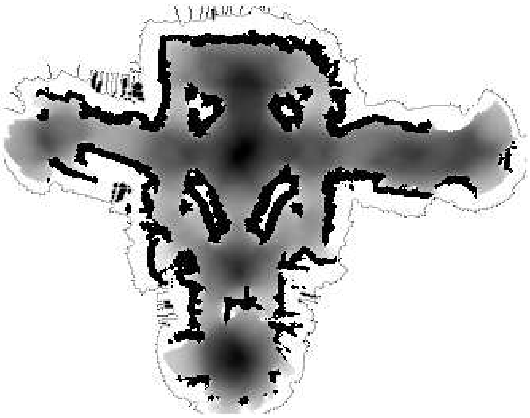


Figure 3: An example map of the entropy, or information content, of the National Museum of American History. The darker an area is, the less information content it contains. The blackest areas of the map are the walls.

proximity sensors. We therefore must also account for the likelihood that information can be corrupted.

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.

$$p^{corrupt}(s) = 1 - \gamma^{||s||} \tag{8}$$

$s : 0 \leq ||s|| \leq MaxRange$ is the particular range measurement, $||s||$ is its length and $\gamma : 0 \leq \gamma \leq 1$ is the probability that any particular point in the environment is occupied by a dynamic obstacle: for the case of museum, this is simply the number of people in the museum, divided by the area of the free space of the museum.

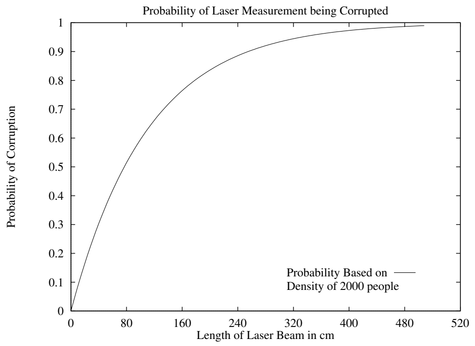


Figure 4: Probability of corruption of laser measurement, as a function of measured distance, given 2000 people in the museum.

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.

In order to alter the information content computation to account for this corruption model, we need to alter the effect that individual sensor measurements have on the total information content. One approach is to integrate the information content over all possible distributions of people in the museum, weighting the information by the likelihood of each distribution. This approach has the disadvantage of being computationally intractable. We therefore make a simplifying assumption that each component of the sensor measurement $s_i(x, y) : \mathrm{s}(x, y) = \{s_1(x, y), \ldots, s_n(x, y)\}$ is independent. This allows us to compute the expected value of $I(x, y)$ by averaging over the information content, $I_i(x, y)$ of each component of the measurement. We make this independence assumption for the sake of computational speed.[2]

Equation (7) gives the information based on data set s. For a typical laser range scan, s contains 360 measurements. We now consider the data as n individual successive measurements, $s_1 \ldots s_n$. The information content $I(x, y)$ can be computed for each s_i alone, giving $I_1(x, y) \ldots I_n(x, y)$, computed as in equation (7). The n measurements are averaged, weighting each measurement by the probability that the measurement was corrupted.

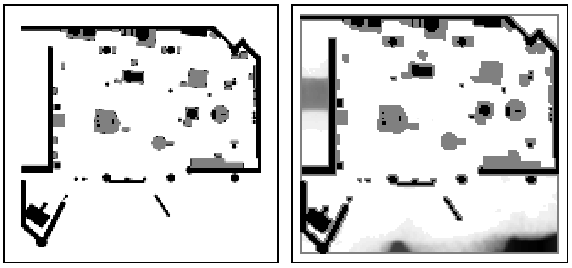
$$I(x, y) = \sum_{i=1}^{n} (I_i(x, y) \cdot p_i^{corrupt}(x, y)) \tag{9}$$

$I_i(x, y)$ is the information content at (x, y) as in equation (7), based only on the i-th sensor reading. The probability $p_i^{corrupt}(x, y)$ is the probability that the i-th sensor reading is corrupted, computed from the distribution given in equation (8).

For a laser range scanner, each measurement is taken at some angle θ_i around the circumference of the robot. The $I_i(x, y)$ is computed from the entropy using the prior distribution centered at (x, y) and the range measurement at θ_i. One effect of the model of people as discussed above is that, since longer laser measurements are more likely to become corrupted than shorter measurements, the information content of a longer laser measurements is lower than the information content of shorter beams. Thus, the best points in the environment are those that have many, close environmental features, such as walls. The worst points in the environment are those that have few environmental features that are far away. Between these two extrema are positions that contain complete, but highly-corruptible information, compared with positions in the map that contain incomplete, but more reliable information.

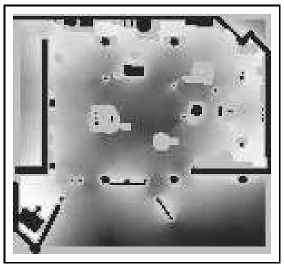
[2]In reality, the range to an obstacle at direction θ_i is highly correlated with the range θ_{i+1} in most environments. However, we can make reasonable conclusions about the information content of each position in the map nevertheless, and this method has the advantage of being computationally fast.

Figure 5 depicts the three stages of the information content development: the initial map, the information content based on a static environment, and the information content based on a dynamic environment. This particular environment is the Deutsches Museum in Bonn, Germany. The size of the environment is 19.8m by 21.3m, discretized at 15cm/grid cell. The computation for this environment took 55 minutes to generate on a Pentium II 266 MHz machine. The computation for the map of the NMAH depicted in figure 3 took 3 hours on the same machine.



(a) Museum map

(b) Information content for a static environment

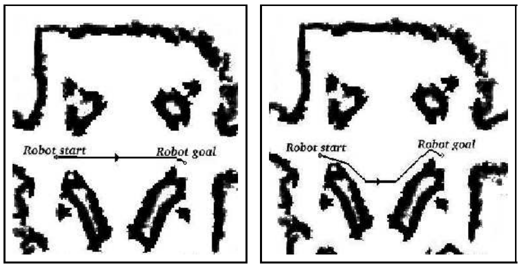


(c) Information content for a dynamic environment

Figure 5: The various stages of the information content map construction process, for the Deutsches Museum in Bonn, Germany. The black obstacles are visible to the laser range finder, the light-grey regions are obstacles not visible to the robot.

5 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 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 [Thr99], along



(a) Conventional trajectory

(b) Coastal trajectory

Figure 6: Example trajectories using the conventional and coastal planners, in the National Museum of American History, for the same start and goal positions. Note the motion of the robot along the wall for the coastal planner.

the path given by the list of cells (x_i, y_i) from start to goal:

$$Cost_{Total} = \sum_{X,Y} c(x_i, y_i) \tag{10}$$

The cost $c(x_i, y_i)$ is the cost of crossing cell (x_i, y_i), which increases with the probability that the cell is occupied, from some minimum cost associated with travel. The minimum $Cost_{Total}$ is found by dynamic programming (also known as Viterbi or Dijkstra method) [How60].

An example trajectory is shown in figure 6(a). 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.

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

$$Cost_{Total} = \sum_{X,Y} \lambda_1 c(x_i, y_i) + \lambda_2 I(x_i, y_i) \tag{11}$$

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

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

It should be emphasized that the computation of the information maps is a one-time operation for any particular environment. The information content is computed off-line, and used by the path planner to construct a single static cost function $c(x_i, y_i)$. This cost function is used by the dynamic programming search; a typical path for the Museum of American History took under 100ms to compute.

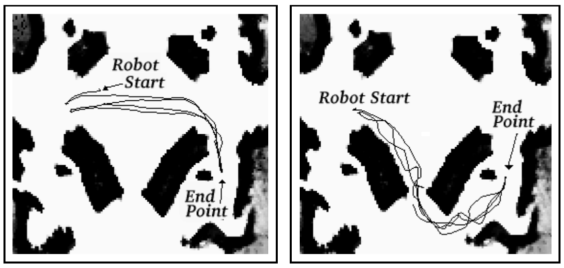
6 Experimental Results

Over the course of two weeks, our robot Minerva gave tours of exhibits in the National Museum of American History, shown in figure 1, using the coastal planner to generate trajectories between exhibits. The robot operated from 10am to 5:30pm daily for 14 days. The total distance by the robot covered was 44km, at an average speed of 38.8 cm/sec. The majority of this time was spent interacting with people; approximately 50,000 people passed through the museum during the two-week period. The main motivations for developing the coastal navigation technique were the large open space in the main operational area for Minerva and the many people gathered around the robot at any given time.

The sensor and localization data was recorded during the operation of Minerva, and some statistics were gathered to compare the performance of the coastal planner to the conventional planner. The conventional planner was also used for part of Minerva's operation, in order to allow comparison of the two navigation methods. The most useful statistic is the average entropy of the probability distribution of the robot's pose, as it travelled along the trajectories.

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..



(a) Conventional trajectory (b) Coastal trajectory

Figure 7: The coastal and conventional trajectories, for the same start and goal.

In the best case, the robot followed trajectories that had a measurably lower average entropy, which indicates the success of the coastal navigation. Table 1 shows the result of using a laser range finder with a 3m max range. The trajectories are given in figure 7. The same start and goal were given to the robot, using first the conventional and then the coastal planner. The robot travelled between the start and goal position 4 times, to generate trajectories of length 87.0m (conventional) and 109.8m (coastal).

Figure 8 shows the performance of the coastal navigation under different sensor abilities, in a static environment. As the sensor range increases to 50m, the sensor is able to use its entire field of view for localization at most points in the museum, so the conventional planner generates paths that are equally in localization ability as the coastal planner. The use of the sensors of different scales illustrates that there are different environmental conditions under which coastal navigation is more or less useful. This graph indicates that if the museum were even spacious, such as a typical outdoor environment, coastal navigation would be even more important.

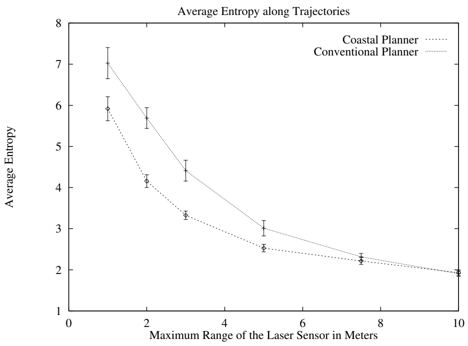


Figure 8: This graph shows the average entropy over trajectories of 87.0m and 109.8m in the National Museum of American History.

7 Conclusion

In this paper, we have presented a method of generating trajectories through environments where positional uncertainty is likely to accrue. Localization methods can often fail in environments that lack many reference points, for example wide-open spaces with walls outside the range of the sensors. Localization can also fail where the sensors are obstructed by dynamic obstacles, such as people. The solution is to generate trajectories that minimize the probability that the robot will fall victim to these problems, and become lost. The method draws on ship-based navigation, where ships lacking reliable global position estimation stay close to known landmarks along shores.

The algorithm operates in two parts. The first part generates a map of the environment that contains the information content of each position in the environment. This representation includes the likelihood of the sensor data to be corrupted by dynamic obstacles. 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.

This particular solution is a special case of a general class of POMDP problems. However, POMDP problems are computationally intractable for systems with large numbers of states. We reduce the complexity by making a number of assumptions such as the ability of the robot to track its position, and the kind of the positional error that accrues.

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. Another direction for future work lies in determining the planner parameters appropriately. The weights λ_1 and λ_2 were set empirically, as were the parameters of the geometric distribution describing the crowdedness of the environment. It would be useful to have the planner choose different parameters based on the perceived crowdedness of the environment, following more coastal trajectories in highly crowded environments, and more direct trajectories in mostly-static environments.

8 Acknowledgements

The authors would like to thank Tom Mitchell for his advice and support of this research. The assistance of Maren Bennewitz, Frank Dellaert, Dirk Haehnel, Chuck Rosenberg, Dirk Schultz and Jamieson Schulte was invaluable both in operating Minerva, and acting as dynamic obstacles. We would also like to thank the Lemelson Center for Innovation and Invention, and the National Museum of American History for the use of their space for these experiments.

This research was supported in part by Le Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Fonds FCAR). This research is sponsored in part by DARPA via AFMSC (contract number F04701-97-C-0022), TACOM (contract number DAAE07-98-C-L032), and Rome Labs (contract number F30602-98-2-0137). Additional financial support was received from Daimler Benz Research and Andy Rubin, all of which is gratefully acknowledged.

References

[BCF+98] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Gerhard Lakemeyer, Dirk Hähnel, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Proc. 15th Nat. Conf. on Artificial Intelligence*, Madison,WI, 1998.

[CKL94] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman. Acting optimally in partially observable stochastic domains. In *Proc. 12th Nat. Conf. on Artificial Intelligence*, Seattle, WA, 1994.

[DZ96] Gregory Dudek and Chi Zhang. Vision-based robot localization without explicit object models. In *Proc. Inter. Conf. of Robotics and Automation*, Minneapolis, MN, 1996. IEEE Press.

[Elf90] Alberto Elfes. *Autonomous Robot Vehicles*, chapter Sonar-based real-world mapping and navigation. Springer, Berlin, 1990.

[Erd84] Michael Erdmann. On motion planning with uncertainty. Technical Report 810, AI Lab, MIT, 1984.

[Erd86] Michael Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, 5(1):19–45, Spring 1986.

[FBT98] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 1998. to appear.

[How60] R. A. Howard. *Dynamic Programming and Markov Processes*. The MIT Press, Cambridge, MA, 1960.

[KCK96] Leslie Pack Kaelbling, Anthony R. Cassandra, and James A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1996.

[KS96] S. Koenig and R.G. Simmons. The effect of representation and knowledge on goal-directed exploration with reinforcement learning algorithms. *Machine Learning Journal*, 22:227–250, 1996.

[MD94] Paul MacKenzie and Gregory Dudek. Precise positioning using model-based maps. In *Proc. Int. Conf. on Robotics and Automation*, San Diego, CA, 1994. IEEE Press.

[ME85] Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 116–121, 1985.

[NPB95] Illah Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

[SD98] Robert Sim and Gregory Dudek. Mobile robot localization from learned landmarks. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, Victoria, BC, Octover 1998.

[TFL94] Haruo Takeda, Claudio Facchinetti, and Jean-Claude Latombe. Planning the motions of mobile robot in a sensory uncertainty field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(10):1002–1017, October 1994.

[Thr99] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 1:27–71, 1999. to appear.