Planning with Uncertainty in Position Using High-Resolution Maps

Juan Pablo Gonzalez

Thesis Proposal

CMU-RI-TR-06-45

Thesis Committee:
Anthony Stentz (chair)
Martial Hebert
Drew Bagnell
Steven LaValle, University of Illinois at Urbana

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

October 2006

© Carnegie Mellon University

Abstract

Navigating autonomously is one of the most important problems facing outdoor mobile robots. This task can be extremely difficult if no prior information is available, and would be trivial if perfect prior information existed. In practice prior maps are usually available, but their quality and resolution varies significantly.

When accurate, high-resolution prior maps are available and the position of the robot is precisely known, many existing approaches can reliably perform the navigation task for an autonomous robot. However, if the position of the robot is not precisely known, most existing approaches would fail or would have to discard the prior map and perform the much harder task of navigating without prior information.

Most outdoor robotic platforms have two ways of determining their position: a dead-reckoning system and Global Position Systems (GPS). The dead reckoning system provides a locally accurate and locally consistent estimate that drifts slowly, and the GPS provides globally accurate estimate that does not drift, but is not necessarily locally consistent. A Kalman filter usually combines these two estimates to provide an estimate that has the best of both position estimates.

While for many scenarios this combination suffices, there are many others in which GPS is not available, or its reliability is compromised by different types of interference such as mountains, buildings, foliage or jamming. In these cases, the only position estimate available is that of the dead-reckoning system which drifts with time and does not provide a position estimate accurate enough for most navigation approaches.

This proposal addresses the problem of planning with uncertainty in position using high-resolution maps. The objective is to be able to reliably navigate distances of up to one kilometer without GPS through the use of accurate, high resolution prior maps and a good dead-reckoning system. Different approaches to the problem are analyzed, depending on the types of landmarks available, the quality of the map and the quality of the perception system.

Contents

1 Introduction

Navigating autonomously is one of the most important problems facing outdoor mobile robots. This task can be extremely difficult if no prior information is available, and it would be trivial if perfect prior information existed. In practice prior maps are usually available, but their quality and resolution varies significantly.

When accurate, high-resolution prior maps are available and the position of the robot is precisely known, many existing approaches can reliably perform the navigation task for an autonomous robot. However, if the position of the robot is not precisely known, most existing approaches would fail or would have to discard the prior map and perform the much harder task of navigating without prior information.

Most outdoor robotic platforms have two ways of determining their position: a dead-reckoning system and Global Position Systems (GPS). The dead reckoning system provides a locally accurate and locally consistent estimate that drifts slowly, and the GPS provides globally accurate estimate that does not drift, but is not necessarily locally consistent. A Kalman filter usually combines these two estimates to provide an estimate that has the best of both position estimates.

While for many scenarios this combination suffices, there are many others for which GPS is not available, or its reliability is compromised by different types of interference such as mountains, buildings, foliage or jamming. In these cases, the only position estimate available is that of the dead-reckoning system which drifts with time and does not provide a position estimate accurate enough for most navigation approaches.

2 Related Work

2.1 Classical Path Planning with Uncertainty in Position

Classical path planning deals with the problem of finding the best path between two locations, assuming the position of the robot is known at all times. Because of the deterministic nature of the problem, and the fact that good heuristics can be easily found, the problem can be efficiently addressed using deterministic search techniques. In order to deal with uncertainty in position, classical path planning algorithms usually model uncertainty as a region of uncertainty that changes shape as states propagate in the search.

2.1.1 Indoors

Most of the prior work in classical path planning with uncertainty deals with indoor environments and as such represents the world as either *free* or *obstacle*. In most of the approaches from classical path planning uncertainty is modeled as a set that contains all the possible locations within the error bounds defined by an uncertainty propagation model.

The first approach to planning with uncertainty, by Lozano-Perez, Mason and Taylor [44], was called *pre-image back chaining.* This approach was designed to plan fine motions in the presence of uncertainty in polygonal worlds. Latombe, Lazanas and Shekhar [39] expanded this approach by proposing practical techniques to calculate *pre-images*, still within the limitations of binary, polygonal worlds. Latombe [40][38] has an extensive review of other similar approaches as of 1991.

Lazanas and Latombe [42] later expanded the *pre-image backchaining* approach to robot navigation. For this purpose they define the bounds in the error of the position of the robot as a disk, and they use an uncertainty propagation model in which the error in the position of the model increases linearly with distance traveled. The idea of a landmark is also introduced as a circular region with perfect sensing. The world consists of free space with disks that can be either landmarks or obstacles.

Takeda and Latombe [62][63], proposed an alternative approach to planning with uncertainty using a sensory uncertainty field. In their approach, the landmarks in the environments are transformed into a sensory uncertainty field that represents the landmark's ability to localize the robot. Then they use Dijkstra's algorithm to find a path that minimizes the uncertainty field or a combination of uncertainty and another objective function. The planner is limited to a polygonal representation of the world, and does not actually consider the uncertainty in position that the robot accrues as it executes the path.

Bouilly, Simeón and Alami [3][4] use a potential-field approach. In their approach the world is described by free space and polygonal obstacles and landmarks. They add the notion of localization with walls. The also introduce the notion of minimizing either the length of the resulting path or its uncertainty.

Fraichard and Mermond [10][20][21] extend the approach of Bouilly *et al* by adding a third dimension to the search (for heading) and computing non-holonomic paths for car-like robots. They also use a more elaborate model for uncertainty propagation in which they calculate separately the uncertainty caused by errors in the longitudinal and steering controls.

2.1.2 Outdoors

Planning paths for outdoor applications represents additional challenges with respect to planning for indoor environments. The main challenges are the varying difficulty of the terrain, the lack of structure, and the size of the environment.

The varying difficulty of the terrain requires a representation of the environment that is able to represent terrain types other than *free space* and *obstacles.* Outdoor terrain has many variations with different implications for robot navigation: there are paved roads, unpaved roads, grass, tall grass, hilly areas, etc. The non-binary nature of outdoor terrain also requires a different spatial representation for the world. While polygonal representations are adequate and efficient for indoor environments, they are not practical or efficient for varying terrain types. The most common way to represent this varying terrain is through a cost map. Cost maps are usually uniform grids of cells in which the cost at each cell represents the difficulty of traversing that cell. The main problem of uniform cost maps is that they can only represent the world up to the resolution of the cells in the map. Although there are other representations that avoid this problem, most cost maps are implemented on a uniform grid.

Haït, Siméon and Taïx were the first to consider the problem of planning with uncertainty in an outdoor environment. In the approach proposed in [27], the resulting path minimizes a non-binary cost function defined by a cost map, but the definition of uncertainty is limited to a fixed local neighborhood in which the planner checks for the validity of the paths. In [28] they expand their uncertainty model to be a disk with a radius that grows linearly with distance traveled, and they add landmarks as part of the planning process. Their approach performs a wave propagation search in a 2-D graph and attempts to minimize worst-case cost instead of path length. However, because their representation is limited to 2-D, it is unable to find solutions for problems in which uncertainty constraints require choosing a higher-cost path in order to achieve lower uncertainty. See section 4.1.2 for more details on the importance of modeling uncertainty as a third dimension.

2.2 Planning with Uncertainty Using POMDPs

A Partially Observable Markov Decision Process (POMDP) is a representation in which both states and actions are uncertain [9][32] therefore providing a rich framework for planning under uncertainty. To handle partial state observability, plans are expressed over *information states*, instead of world states, since the latter ones are not directly observable. The space of information states is the space of all beliefs a system might have regarding the world state and is often known as the *belief space*. In POMDPs, information states are typically represented by probability distributions over world states. POMDPs can be used for either belief tracking or policy selection.

When used for belief tracking, the solution of a POMDP is a probability distribution representing the possible world states. Most approaches to belief tracking perform only robot localization (see section 2.3.1), however a few perform planning based on the solution of the belief tracking POMDP. Nourkbakhsh *et al*, [45] quantize the world into corridors and interleave planning and execution by using a POMDP to keep track of all possible states, and then find the shortest path to the goal from the most likely location of the robot. Their results are implemented in the DERVISH robot, allowing it to win the 1994 Office Delivery Event of in AAAI'94.

Simmons *et al* [57] use a similar approach of interleaving planning and execution on Xavier, but quantize the position of the robot in one-meter intervals along corridors. From the most likely position reported by the POMDP they plan an A* path that accounts for the probability of corridors being blocked or turns being missed.

When used for policy selection the solution of a POMDP is a policy that chooses what action to attempt at each information state based on the state estimate and the results of the previous actions. In this case, the optimal solution of a POMDP is the policy that maximizes the expected reward considering the probability distributions of states and actions. While the model is sufficiently rich to address most robotic planning problems, exact solutions are generally intractable for all but the smallest problems [48].

Because POMDPs optimize a continuous value function they could easily handle the continuous cost worlds typical of outdoor environments. However, because of their complexity, they have only been used for indoor environments where the structure and size of the environment significantly reduce the complexity of the problem. In spite of the reduced state space of indoor environments, POMDPs still require further approximations or simplifications in order to find tractable solutions.

Very few POMDP approaches can handle large-enough environments that would be suitable for outdoor environments. One of the leading approaches to extend POMDP's to larger environments is Pineau's Point-Based Value Iteration (PBVI) [47][48]. This approach selects a small set of belief points to calculate value updates, enabling it to solve problems with 10^3-10^4 states, at least an order of magnitude larger than conventional techniques. However, this algorithm still takes several hours to find a solution.

Roy and Thrun implemented an approach called Coastal Navigation [50][52] that models the uncertainty of the robot's position as a state variable and minimizes the uncertainty at the goal. They model the uncertainty through a single parameter which is the entropy of a Gaussian distribution and then use Value Iteration to find an optimal policy in this compressed belief space. Their algorithm has a lengthy pre-processing stage but is able to produces results in a few seconds after the initial pre-processing state. The total planning time (including the pre-processing stage) can take from several minutes to a few hours [53].

Roy and Gordon [51] using Exponential Family Principal Component Analysis (E-PCA) take advantage of belief space sparsity. The dimensionality of the belief space is reduced by exponential family Principal Components Analysis, which allows them to turn the sparse, high-dimensional belief space into a compact, low-dimensional representation in terms of learned features of the belief state. They then plan directly on the low-dimensional belief features. By planning in a low-dimensional space, they can find policies for POMDPs that are orders of magnitude larger than what can be handled by conventional techniques. Still, for a world with 10^4 states E-PCA takes between two and eight hours to find a solution, depending on the number of bases used.

2.3 Robot Localization

Localization is a fundamental problem in robotics. Using its sensors, a mobile robot must determine its localization within some map of the environment. There are both passive and active versions of the localization problem: [41]

2.3.1 Passive Robot Localization

In passive localization the robot applies actions and its position is inferred from the sensor outputs collected during the traverse.

Kalman Filter

The earliest and most common form of passive robot localization is the Kalman filter [33], which combines relative (dead-reckoning) and absolute position estimates to get a global position estimate. The sources for the absolute position estimates can be landmarks, map features, Global Positioning System (GPS), laser scans matched to environment features, etc.

The Kalman Filter is an optimal estimator for a linear dynamic process. Each variable describing the state of the process to be modeled is represented by a Gaussian distribution and the Kalman filter predicts the mean and variance of that distribution based on the data available at any given time. Because the Gaussian distribution is unimodal, the Kalman filter is unable to represent ambiguous situations and requires an initial estimate of the position of the robot. Approaches like the Kalman filter that can only handle single hypotheses and require and initial estimate of the position of the robot are considered *local approaches*.

If the process to be estimated is not linear, as often happens in real systems, the standard Kalman filter cannot be used. Instead, other versions of the Kalman filter such as the Extended Kalman Filter need to be implemented. The Extended Kalman filter linearizes the system process model every time a new position estimate is calculated, partially addressing the non-linearities of the system. However the estimate produced by the Extended Kalman filter is no longer optimal. See Gelb [23] for an in-depth analysis of the theory and practical considerations of Kalman Filters.

In spite of its limitations, Kalman filters are the most widely used passive localization approach, because of its computational efficiency and the quality of the results obtained when there is a good estimate of the initial position and unique landmarks are available.

Map Matching

One of the earliest forms of passive localization is map matching, which originated in the land vehicle guidance and tracking literature [22][29][64]. The vehicle is usually constrained to a road network and its position on the road network is estimated based on the odometry, heading changes, and the initial position. Because an estimate of the initial position is needed, this method is also considered a *local approach.* The main application of map matching is to augment GPS in tracking the position of a vehicle.

While the first approaches were limited to grid-like road networks, later approaches such as [31] extended the idea to more general road geometries by using pattern recognition techniques. These algorithms attempt to correlate the pattern created by the recent motion of the vehicle with the patterns of the road network. El Najaar [14] uses belief theory to handle the fusion of different data sources and select the most likely location of the vehicle. Scott [54] models map-matching as an estimation process within a well defined mathematical framework that allows map information and other sources of position information to be optimally incorporated into a GPS-based navigation system. Abbott [1] has an extensive review of the literature in map-matching as well as an analysis of its influence on the performance of the navigation system.

Markov Localization

Markov localization uses a POMDP to track the possible locations of the robot (belief tracking). Tracking the belief state of a POMDP is a computationally intensive task, but unlike planning with POMDPs it is tractable even for relatively large worlds. POMDPs are a *global approach* to localization. They can track multiple hypotheses (modes) about the belief space of the robot and are able to localize the robot without any information about the initial state of the robot. They also provide additional robustness to localization failures.

Nourkbakhsh *et al*, [45] introduced the notion of Markov localization, quantizing the world into corridors and using a POMDP to keep track of all possible states. Simmons *et al* [57] use a similar approach on Xavier, but quantize the position of the robot in one-meter intervals along corridors.

Burgard *et al* [5][7][15] introduced grid-based Markov localization. In their approach, they accumulate in each cell of the position probability grid the posterior probability of this cell referring to the current position of the robot. This approach is able to take raw data from range sensors and is therefore able to take advantage of arbitrary geometric features in the environment. Its main disadvantage is that it needs to store a full 3-D grid containing the likelihoods for each position in x, y and θ. The approach is experimentally validated in worlds of about 200x200 cells, but should work in much larger worlds.

Dellaert *et al* [10][18] introduced Monte Carlo Localization. In this approach the probability density function is represented by maintaining a set of samples that are randomly drawn from it (a *particle filter*) instead of a grid. By using a sampling-based representation they significantly improve speed and space efficiency with respect to grid-based approaches, while maintaining the ability to represent arbitrary distributions. It is also more accurate than grid-based Markov localization, and can integrate measurements at a considerably higher frequency.

Markov localization is a powerful and mature technique for robot localization. It is a reliable and fast way to globally localize a robot. However, it is a passive technique and it cannot control the trajectory of the vehicle to choose a path that either guarantees localization or satisfies other mission objectives.

2.3.2 Active Robot Localization

In active localization a plan must be designed to reduce the localization uncertainty as much as possible. Active localization does not plan trajectories that improve localization while simultaneously achieving some goal, but instead dictates the optimal trajectory only for discovering the true location of the robot [15].

Non Deterministic

Non-deterministic approaches to active robot localization model the position of the robot as a set of states. The localization task consists of reducing the set to a singleton (a single state). Most approaches assume that the initial position is unknown and there is no uncertainty in sensing or acting.

Koenig and Simmons use real-time heuristic search (Min-Max Learning Real Time A* - LRTA*) [35][36], which is an extension of LRTA* to handle non-deterministic domains. They are able to produce worst-case optimal results in a maze by modeling the task as a large, non-deterministic domain whose states are sets of poses.

If the robot has a range sensor and a compass then the environment can be modeled as a polygon and use visibility tests to identify hypotheses about the location of the robot followed by a hypothesis elimination phase. Dudek *et al* [10] proved that localizing a robot in this way with minimum travel is NP-hard, and proposed a greedy approach that is complete but that can have very poor performance. Rao *et al* [49] builds on that result and uses the best of a set of randomly selected points to eliminate hypotheses, which produces much better average performance.

O'Kane and La Valle [46] showed that the localization problem in a polygonal world could be solved in a suboptimal fashion with just a compass and a contact sensor. They also showed that this was the minimal robot configuration that would allow localization is such worlds.

Probabilistic

Probabilistic methods for active localization model the distribution of possible states as a probability density function, and then find a set of actions to localize the robot. The POMDP-based planning approaches described in section 2.2 could in theory be used to solve this problem, but they also have the limitations caused by the complexity of POMDPs.

The approach proposed by Burgard *et al* [8][17] is similar to the one by Roy and Thrun described in section 2.2. They use entropy as a single-parameter statistic that summarizes the localization of the robot and perform value iteration selecting actions that minimize entropy at each step. Even though the solution uses a non-sufficient statistic to describe the probability distribution and uses greedy action selection, the algorithm performs very well and is able to work in reasonably large worlds. The approach, however, requires a long pre-processing stage in which the likelihoods of all sensor readings are calculated.

2.4 Active Exploration and SLAM

Simultaneous Localization and Mapping (SLAM), attempts to build a map of an unknown environment while simultaneously using this map to estimate the absolute position of the vehicle. While in general SLAM is a passive approach in which the robot moves and data is collected to build the map and localize the robot, there exists a complementary approach called *active exploration*. In *active exploration* the objective is to find an optimal exploration policy to reduce the uncertainty in the map. As in active localization the optimal solution is intractable, but suboptimal solutions often produce satisfactory results.

One approach to the active localization problem is to attempt to maximize the information gain throughout the map by using gradient descent on the information gain (or change in the relative entropy) [5][67]. The main problem with this approach is that is only locally optimal and is subject to local minima.

Staunchis and Burgard [61] describe one of the few global exploration algorithms. Their approach uses *coverage maps*, an extension of occupancy maps in which each cell represents not only the occupancy of the cell but also a probabilistic belief about the coverage of the cell. In this way they are able to compute the information available at all locations in the environment which allows them to calculate a maximally informative location.

Sim and Roy [56] propose using the *a-optimal* information measure instead of the *d-optimal* information measure (relative entropy), as the measure for information gain. Whereas the *d-optimal* information measure uses the product of the eigenvalues of the distribution, the *a-optimal* information measure uses the sum of the eigenvalues. They then use pruned breadth-first search over all robot positions to search for the expected sequence of estimates the lead to the maximum information gain. While their approach is not optimal, it produces a significantly more accurate map than the map obtained using *d-optimal* information gain.

3 Problem Statement

This proposal addresses the problem of planning paths with uncertainty in position using high-resolution maps. The objective is to be able to reliably navigate autonomously in an outdoor environment without GPS through the use of accurate, high resolution prior maps and a good dead-reckoning system.

The initial position of the robot is assumed to be known within a few meters and the initial heading is also assumed to be known within a few degrees. The dead reckoning estimate in the position of the robot is assumed to drift slowly, at a rate lower than 10% of distance traveled, which is typical of an outdoor robot with wheel encoders and a fiber-optic gyro.

We assume an accurate, high-resolution map that allows the identification of landmarks and the approximate estimation of terrain types by automatic or manual methods. The high-resolution map is translated into a cost grid, in which the value of each cell corresponds to the cost of traveling from the center of the cell to its nearest edge. Non-traversable areas are assigned infinite cost and considered obstacles. We also assume that the robot has an on-board perception system able to reliably detect landmarks that are within a radius R from the robot and able to avoid small obstacles not present in the prior map. The uncertainty when detecting landmarks is assumed to be less than the detection range of the vehicle.

The resulting path should minimize the expected value of the objective function along the path, while ensuring that the uncertainty in the position of the robot does not compromise its safety or the reachability of the goal.

Additionally we will explore in isolation the possibility of relaxing some of the assumptions mentioned above.

4 Technical Approach

Planning with uncertainty in position in outdoor environments is a difficult and computationally expensive problem that presents a number of important challenges such as the varying difficulty of the terrain, the lack of structure, and the size of the environment. The most general solution to planning paths with uncertainty in position requires finding the optimal policy for a POMDP, which is intractable for all but the smallest problems.

We present an alternative approach that takes advantage of the low drift rate in the inertial navigation system of many outdoor mobile robots. The planner uses a Gaussian distribution to model position uncertainty and uses deterministic search to efficiently find resolution-optimal paths that minimize expected cost while considering uncertainty in position.

We model the world as a continuous-cost map derived from a high-resolution prior map. This model allows us to model the varying difficulty of the terrain typical of outdoor environments. We model uncertainty as an additional dimension that constrains the feasible solutions to the problem and use deterministic search techniques such as A*, heuristics, lazy evaluation and state dominance to efficiently find solutions in this larger state representation.

Our approach also uses landmarks to reduce uncertainty as part of the planning process. Landmarks are carefully chosen such that they can be reliably detected and their detection can be modeled as a deterministic event. Because landmarks that can be reliably detected such as trees and electric poles are not unique, we use an estimate of the position of the robot in order to disambiguate them and prevent aliasing as part of the planning process.

This section is structured as follows: we first analyze the uncertainty propagation for a mobile robot modeled as a unicycle. We use this analysis to justify the use of a Gaussian model for uncertainty propagation and to further reduce this model to an isometric Gaussian in order to reduce the number of parameters required. Next we analyze the importance of prior maps and how to extract cost estimates from them. We then use the isometric Gaussian model to develop a path planner that minimizes expected cost while considering uncertainty in position assuming very limited perception. Next, we add landmark detection capabilities and include landmarks in the planning process: we start by analyzing point landmarks and then increase the complexity of the landmark detection task by increasing the density of the landmarks until they can be treated as linear features rather than individual features. Finally, we explore the implications of relaxing the assumptions of perfect maps and perfect sensing.

4.1 Work to Date

4.1.1 Background

Analysis of Uncertainty Propagation

The first-order motion model for a point-sized robot moving in two dimensions is:

$$\begin{aligned}\dot{x}(t) &= v(t)\cos\theta(t) \\ \dot{y}(t) &= v(t)\sin\theta(t) \\ \dot{\theta}(t) &= \omega(t)\end{aligned} \tag{1}$$

where the state of the robot is represented by x(t), y(t) and θ(t) (x-position, y-position and heading respectively), and the inputs to the model are represented by v(t) and ω(t) (longitudinal speed and rate of change for the heading respectively). Equation (1) can also be expressed as:

$$\dot{\mathbf{q}}(t) = f(\mathbf{q}(t), \mathbf{u}(t)) \tag{2}$$

where $\mathbf{q}(t) = (x(t), y(t), \theta(t))$ and $\mathbf{u}(t) = (v(t), \omega(t))$.

A typical sensor configuration for a mobile robot is to have an odometry sensor and an onboard gyro. We can model the errors in the odometry and the gyro as errors in the inputs where $w_v(t)$ is the error in $v(t)$ (error due to the longitudinal speed control), and $w_\omega(t)$ is the error in $\omega(t)$ (error due to the gyro random walk).

Incorporating these error terms into (1) yields:

$$\begin{aligned}\dot{x}(t) &= (v(t) + w_v(t))\cos\theta(t) \\ \dot{y}(t) &= (v(t) + w_v(t))\sin\theta(t) \\ \dot{\theta}(t) &= \omega(t) + w_\omega(t)\end{aligned} \tag{3}$$

or, in discrete-time:

$$\begin{aligned}x(k+1) &= x(k) + (v(k) + w_v(k))\cos\theta(k)\Delta t \\ y(k+1) &= y(k) + (v(k) + w_v(k))\sin\theta(k)\Delta t \\ \theta(k+1) &= \theta(k) + (\omega(k) + w_\omega(k))\Delta t\end{aligned} \tag{4}$$

Using the extended Kalman filter (EKF) analysis for this system, which assumes that the random errors are zero-mean Gaussian distributions, we can model the error propagation as follows:

$$\mathbf{P}(k+1) = \mathbf{F}(k)\cdot\mathbf{P}(k)\cdot\mathbf{F}(k)^T + \mathbf{L}(k)\cdot\mathbf{Q}(k)\cdot\mathbf{L}(k)^T \tag{5}$$

where

$$\mathbf{P}(k) = E(\hat{\mathbf{q}}(k)\hat{\mathbf{q}}(k)^T) \qquad \mathbf{Q}(k) = \frac{1}{\Delta t}\begin{pmatrix}{\sigma_v}^2 & 0 \\ 0 & {\sigma_\omega}^2\end{pmatrix} \tag{6}$$

$$\mathbf{F}_{ij} = \frac{\partial f(q_i(k), u_j(k))}{\partial q_i}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & -v(k\Delta t)\sin(\theta(k)) \cdot \Delta t \\ 0 & 1 & v(k\Delta t)\cos(\theta(k)) \cdot \Delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

$$\mathbf{L}_{ij} = \frac{\partial f(q_i(k), u_j(k))}{\partial u_j}$$

$$\mathbf{L} = \begin{pmatrix} \cos(\theta(k)) \cdot \Delta t & 0 \\ \sin(\theta(k)) \cdot \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \tag{8}$$

Although there are no general closed-form solutions for these equations, Kelly [31] calculated closed-form solutions for some trajectories, and showed that a straight line trajectory maximizes each one of the error terms. We can use the results for this trajectory as an upper bound on the error for any trajectory. For a straight trajectory along the x-axis keeping all inputs constant, the error terms behave as follows. The error due to the longitudinal speed control w_v is reflected in the x direction, and is given by $\sigma_x^2 = \sigma_v^2 \cdot vt$, or $\sigma_x = \sigma_v \sqrt{vt}$. The error due to the gyro random walk w_ϖ is reflected in the y direction, and is given by $\sigma_y^2 = \sigma_\omega^2 \cdot v^2 t^3$, or $\sigma_y = \sigma_\omega \cdot v \cdot t^{3/2}$.

Additionally, there are errors in the initial position of the robot. Errors in x and y do not increase unless there is uncertainty in the model or in the controls. Errors in the heading angle, however, propagate linearly with *t*. For small initial angle errors, the approximation $\sin\theta \approx \theta$ can be used to obtain the expression $\sigma_y = \sigma_{\theta_o} .vt$. As long as the total heading error is small, the first order approximation of the EKF will be a good approximation of the error propagation.

The dominant terms in the error propagation model depend on the navigation system and on the planning horizon for the robot. A typical scenario for a mobile robot with good inertial sensors is to have a planning horizon of up to 3km, at a speed of 5 m/s, with longitudinal control error of 10% of the commanded speed $(\sigma_v = 0.1v = 0.5\, m/s)$ and gyro drift (random walk) of $20°/h/\sqrt{Hz}$ $(\sigma_\omega = 0.005°/s \ = 8.72 \cdot 10^{-5} rad/s)$. In this scenario, the dominant term is the error in the longitudinal control. However, if there is uncertainty in the initial angle, this term becomes the dominant one for errors as small as 0.5 degrees (See Figure 1).

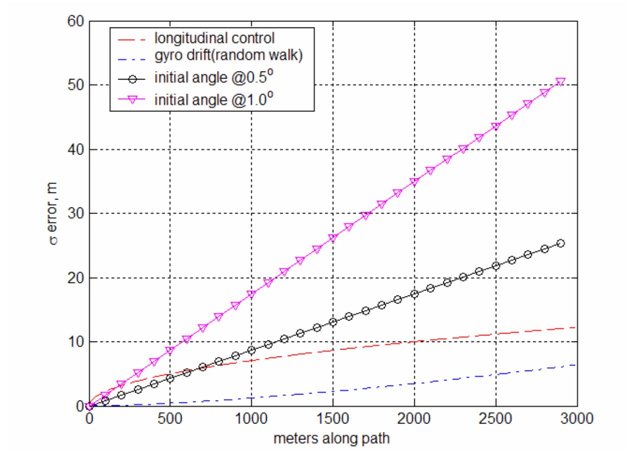


Figure 1. Comparison between different values of initial angle error and longitudinal control error

If we were to model the uncertainty propagation with a single parameter, the most appropriate model would be that of the dominant term. In this case, that dominant term is the error in the initial heading, which varies linearly with distance traveled.

Figure 2 shows the error propagation combining all the types of error (with the values mentioned above, and an initial heading error of 0.5 degrees) for a straight trajectory. The figure uses the results of a Monte Carlo simulation as a reference and it also shows the 2σ contours for the full EKF model and the 2σ contours for a single-parameter (isometric) Gaussian. Figure 3 shows the same analysis for a random trajectory. Notice how the single-parameter Gaussian is now a looser bound for the error propagation, and how the full EKF model is still a good approximation. See [25] for a more detailed analysis of the different modes of error propagation.

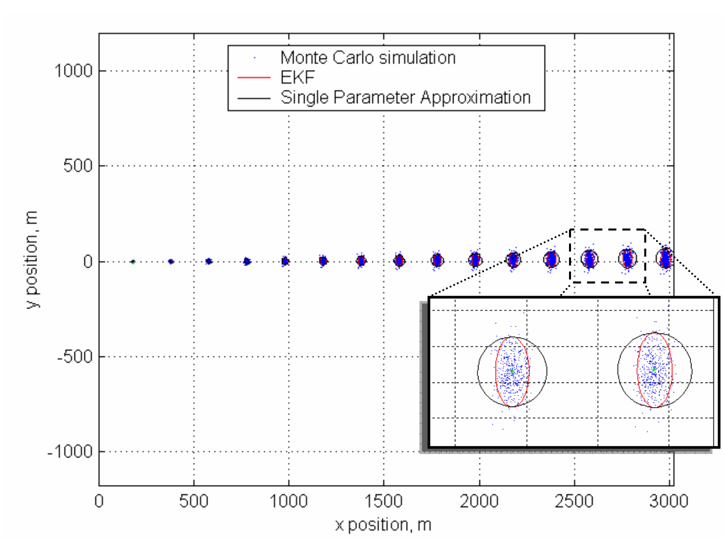


Figure 2. Error propagation for a straight trajectory with all error sources combined and inset showing detail. The blue dots are the Monte Carlo simulation, the red ellipse is the EKF model (2σ), and the black circle is the single parameter approximation (2σ)

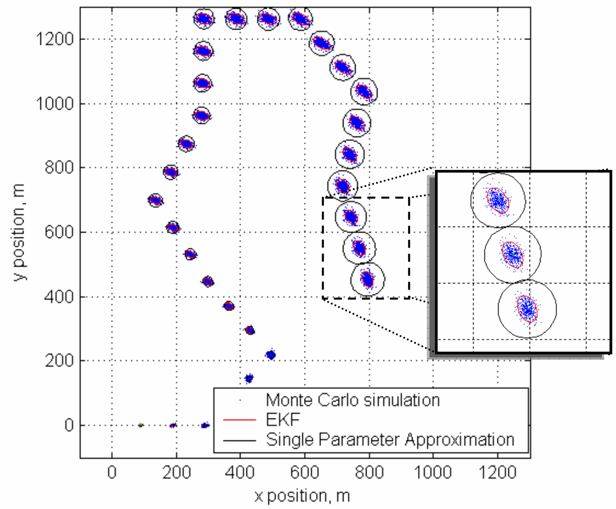


Figure 3. Error propagation for a random trajectory with all error sources combined and inset showing detail. The blue dots are the Monte Carlo simulation, the red ellipse is the EKF model (2σ), and the black circle is the single parameter approximation (2σ).

Prior Maps

The task of navigating autonomously can be extremely difficult if no prior information is available, and would be trivial if perfect prior information existed. In practice prior maps are usually available, but their quality and resolution varies significantly.

When accurate, high-resolution prior maps are available and the position of the robot is precisely known, many existing approaches can reliably perform the navigation task for an autonomous robot. However, if the position of the robot is not precisely known, most existing approaches would fail or would have to discard the prior map and perform the much harder task of navigating without prior information.

The approach presented here requires prior maps to estimate the cost to traverse different areas and to provide landmarks for navigation.

Cost Map

The cost map is the representation of the environment that the planner uses. It is represented as a grid, in which the cost of each cell corresponds to the cost of traveling from the center of the cell to its nearest edge. Non-traversable areas are assigned infinite cost and considered obstacles.

The resulting path calculated by the planner minimizes the expected value of the cost along the path, while ensuring that the uncertainty in the position of the robot does not compromise its safety or the reachability of the goal.

The procedure to create a cost map from a prior map depends on the type of prior map used. If elevation maps are available, cost is usually calculated from the slope of the terrain. When only aerial maps are available, machine learning techniques such as those in [55][59][59] can be used.

Figure 4 shows the cost map for the test area used in the experimental results presented here. The cost map was created by training a Bayes classifier and adding manual annotations to the resulting map. The table below shows the cost assigned to the different terrain types.

TABLE I

COST VALUES FOR DIFFERENT TERRAIN TYPES.

Terrain Type	***Cost***
Paved Road*	5
Paved Road 2	10
Dirt Road	15
Grass	30
Trees	40
Water	250
Buildings*	255

* Items manually labeled.

Figure 4. Cost map: lighter regions represent lower cost, and darker regions represent higher cost. Green areas are manually labeled buildings.

Map Registration

A prior map that is not correctly registered to the position of the vehicle is of little use for most planning approaches. The error in map registration usually comes from two main sources: error in the estimation of the position of the vehicle, and error in the estimation of the position of the map. The approach presented here uses the prior map as the reference for all planning and execution. Since the planner considers uncertainty in position, the error in map registration can be modeled as being part of the error in the position of the robot, therefore making use of the information of the map in a way that includes the total uncertainty in the position of the robot

Landmarks

Landmarks are features that can be identified in the prior map and that can be detected with the on-board sensors. The selection of landmarks depends on the quality and type of the prior map, the characteristics of the environment and the detection capabilities of the robot.

Landmarks can come from a separate database of features, or can be extracted directly from the prior map if the resolution of the map is high enough. In our approach, we use aerial maps with a resolution of 0.3 meters per cell. At this resolution, many features are clearly visible and can be identified using manual labeling. Automatic extraction is also possible for some types of features, but the state-of-the-art for automatic feature detection does not yet allow for reliable extraction of most features.

We classify landmarks based on their appearance on a map. *Point landmarks* are geographic or man-made features such as tree trunks and electric poles that can be represented in a map by a point. *Linear landmarks* are geographic or man-made features such as walls and roads that can be represented in a map by a line or set of lines.

While some landmarks can be easily classified as point or linear landmarks, many can be seen as either one, depending on the characteristics of the feature and the detection range of the vehicle. A house with four walls, for example, can be seen as four linear landmarks (the straight parts of the walls) plus four point features (the four corners). Furthermore, because the corners differ by a rotation of more than 90 degrees, they can often be seen as four separate types of point landmarks. A line of trees can be seen a either a set of point landmarks or a single linear landmark, depending on the separation of the trees and the detection range of the vehicle.

Figure 5 shows a small section of our test area with landmarks manually labeled. Features 1 through 7 are point landmarks (electric poles); features 8 through 11 are linear landmarks (walls of a building), and features 12 through 15 are less obvious point landmarks (corners of a building).

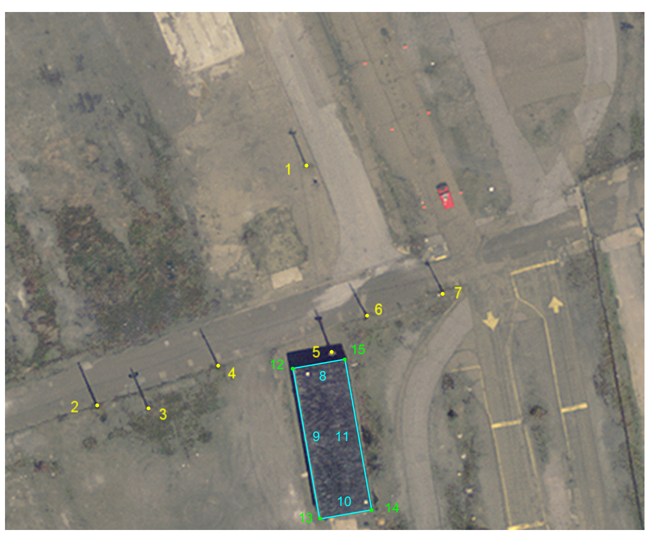


Figure 5. Detail of test area showing features of interest

Perception

One of the most challenging aspects of autonomous navigation is perception in unstructured or weakly structured outdoor environments such as forests, small dirt roads and terrain covered by tall vegetation [11]. While there has been great progress in perception in recent years, perception is still an area where, in general, only limited reliability has been achieved.

While general terrain classification and feature identification still have significant shortcomings, the detection of simple features such as tree trunks and poles can be performed in a more reliable way [37][66].

Simple features, however, are insufficient to provide unique location information because of aliasing and the fact that they are not unique. But we can make them unique and more reliable by combining them with an estimate of the position of the robot. By combining prior map information and planning with perception are able to improve the performance of the perception in the same way that using prior maps improves the navigation capabilities of robotic system.

For planning purposes the vehicle is assumed to have a range sensor with a maximum detection range R and 360^o field of view. We use electric poles as landmarks since they are widely available in our test location and can be reliably detected at distances of up to 10 meters. With little modification the approach could be modified to detect tree trunks and other similar features.

For execution purposes the vehicle is assumed to be able to detect small obstacles not present in the prior map, and most importantly, to be able to reliably detect the landmarks in the map. We use a simplified version of the approach by Vandapel [66]. Our approach finds areas that have high variance in elevation and low horizontal variance (narrow and tall objects). Although in theory any object with this description would be detected, the height threshold is such that in practice only electric poles are usually detected. In order to uniquely identify each landmark, we only attempt to detect a landmark when our position estimate indicates that we are within its *unique detection region.* See section 4.1.3 for more details.

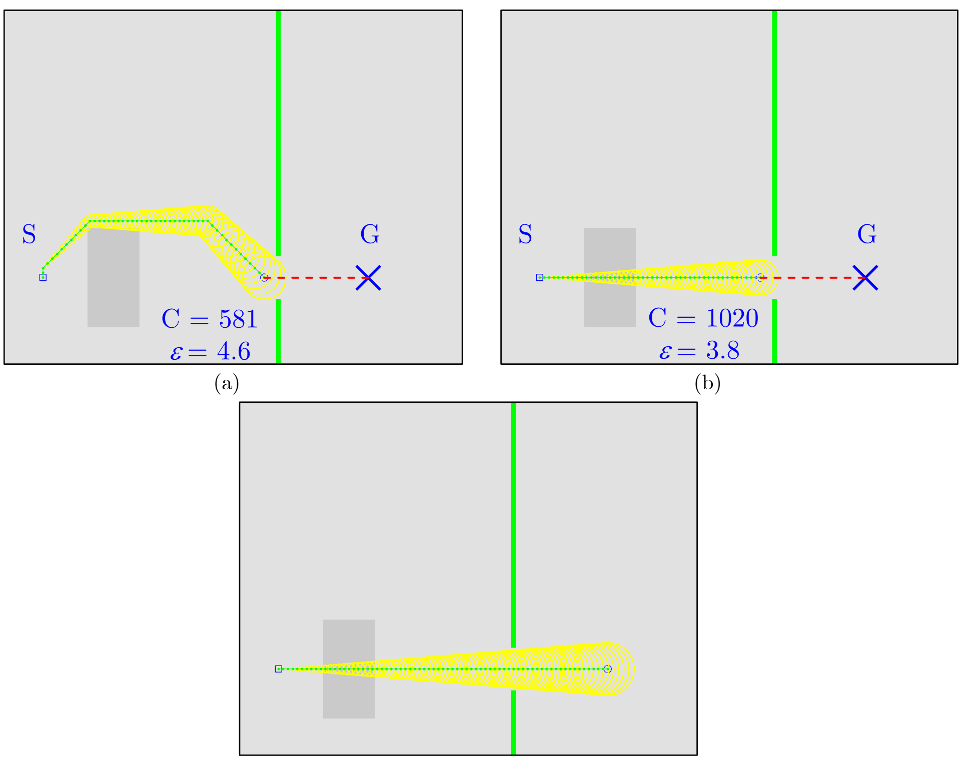
4.1.2 Planning with Uncertainty in Position without Using Landmarks

Planning with uncertainty in position for indoor environments is usually approached as a special case of shortest path planning. Because a binary representation of the world (*free space* or *obstacles*) is usually sufficient for indoor navigation, the best path to follow is usually the shortest path, which is often the path that also minimizes uncertainty in position. This path can be found using 2-D planning approaches such as those described in section 2.1.1.

For outdoor environments, however, the varying difficulty of the terrain requires a representation of the environment that is able to represent terrain types other than *free space* and *obstacles,* such as a cost map. When cost maps are used as the objective function to be minimized by a planner, the lowest cost path is no longer the same as the shortest path or the path with lowest uncertainty. While in binary worlds uncertainty is directly proportional to the objective function that is being minimized (path length), in continuous cost worlds it is not. A 2-

D representation, it is unable to find solutions for problems in which uncertainty constraints require choosing a higher-cost path in order to achieve lower uncertainty. To represent the constraint imposed by uncertainty in the planning problem, it is therefore necessary to add at least one more dimension to the planning problem.

Figure 6 shows an example that highlights the importance of adding a third dimension to represent uncertainty. It describes a sample world with a long obstacle (green) that has a small opening. While most of the space has low cost (5), there is a small region near the goal with higher cost (30). Figures (a) and (b) show an intermediate step in planning a path from the start location (S) to the goal location (G). Figure (a) shows the lowest expected cost path from S to the intermediate state, and (b) shows the lowest uncertainty path, which is more expensive. A 2-D planner attempting to minimize expected cost would only be able to represent the state from (a). While this is often correct, if the larger uncertainty of that state prevents the path from going through the opening in the obstacle, the intermediate state from (a) is no longer the better choice. The intermediate state shown in (b), in spite of having higher cost, is the optimal alternative: it is the only way for the path to fit through the opening in the obstacle and find the lowest cost path shown in (c). Only by modeling uncertainty as an additional dimension is it possible to find the lowest expected cost path in such continuous cost domains.



(a) (b) (c)

Figure 6. Importance of modeling uncertainty as a separate dimension. Light gray areas are low cost (5), dark gray areas are higher cost (30), and green areas are non-traversable (obstacles). (a) Intermediate state with low cost but high uncertainty. (b) Intermediate state with low uncertainty but high cost. (c) optimal path from start to goal

Simplified Error Propagation Model

In order to keep the planning problem tractable and efficient, we use a single-parameter error propagation model. We approximate the probability density function (pdf) of the error with an isometric Gaussian distribution, centered at the most likely location of the robot at step k:

$$\begin{aligned} \mathbf{q} &= (x, y) \\ \mathbf{q} &: N({\mathbf{q}_c}^k, \sigma^k) \end{aligned} \tag{9}$$

where $\mathbf{q}_c^k$ is the most likely location of the robot at step k, and $\sigma^k = \sigma_x^k = \sigma_y^k$ is the standard deviation of the distribution.

Let us define:

$$\varepsilon^k = 2 \cdot \sigma^k \tag{10}$$

We can then model the boundary of the uncertainty region as a disk centered at $\mathbf{q}_c^k$ with a radius ε^k that corresponds to the 2σ contour for the underlying Gaussian distribution. Because the position of the robot is now a probability distribution, the new representation for the robot's position is

$$p(\mathbf{q} \mid {\mathbf{q}_c}^k, \varepsilon^k) \tag{11}$$

where $\mathbf{q}$ is a random variable representing the position of the robot, ${\mathbf{q}_c}^k$ is the most likely location of the robot.

Having a probability distribution over possible robot positions rather than a fixed location means that we are planning over a belief space of the robot [1]. The belief space for this simplified model can be represented by the augmented state variable

$$\mathbf{r} = (\mathbf{q}, \varepsilon) \tag{12}$$

hence defining a 3-D configuration space where x and y are the first two dimensions, and ε is the third dimension.

As shown in section 4.1.1, this model is a conservative estimate of the true error propagation model and, depending on the type of error that is dominant in the system, can provide an accurate approximation of the true model. Since the dominant term in the error propagation for our planning horizon is linear with distance traveled, we use the following model to propagate the uncertainty:

$$\varepsilon(\mathbf{q}_c^k) = \varepsilon(\mathbf{q}_c^{k-1}) + \alpha_u d(\mathbf{q}_c^{k-1}, \mathbf{q}_c^k) \tag{13}$$

where α_u is the uncertainty accrued per unit of distance traveled, $\mathbf{q}_c^{k-1}$ is the previous position along the path, and $d(\mathbf{q}_c^{k-1}, \mathbf{q}_c^k)$ is the distance between the two adjacent path locations $\mathbf{q}_c^{k-1}$ and $\mathbf{q}_c^k$.

Equivalently, we can define the uncertainty at location $\mathbf{q}_c^k$ as:

$$\varepsilon(\mathbf{q}_c^k) = \varepsilon(\mathbf{q}_c^0) + \alpha_u D(\mathbf{q}_c^k, \mathbf{q}_c^0) \tag{14}$$

where $\mathbf{q}_c^0$ is the most likely initial location of the robot, and $D(\mathbf{q}_c^k, \mathbf{q}_c^0)$ is the total distance traveled along the path from $\mathbf{q}_c^0$ to $\mathbf{q}_c^k$. The uncertainty rate α_u is typically between 0.01 and 0.1 (1% to 10%) of distance traveled.

By modeling the error propagation in this manner, we are assuming that the dominant term is the uncertainty in the initial angle. Even though we are not explicitly modeling θ as a state variable, the effects of uncertainty in this variable are accounted for in the uncertainty propagation model for $\mathbf{q}=(x,y)$.

State Propagation

In order to use a deterministic planner to plan over this space we need to define the transition cost between adjacent cells. In our 3-D configuration space, we are interested in calculating the cost of moving between the configuration $\mathbf{r}^k$ (at path step k) and an adjacent configuration $\mathbf{r}^{k+1}$ (at path step k+1). This is equivalent to calculating the expected cost of going from a most likely workspace location ${\mathbf{q}_c}^k$, with uncertainty ε^k to an adjacent most likely workspace location ${\mathbf{q}_c}^{k+1}$ with uncertainty ε^{k+1} (See [26] for details):

$$C(\mathbf{r}^k, \mathbf{r}^{k+1}) = E\left[C\left(({\mathbf{q}_c}^k, \varepsilon^k), ({\mathbf{q}_c}^{k+1}, \varepsilon^{k+1})\right)\right] \tag{15}$$

Equivalently,

$$E\left[C\left(({\mathbf{q}_c}^k, \varepsilon^k), ({\mathbf{q}_c}^{k+1}, \varepsilon^{k+1})\right)\right] = \sum_{\forall i}\sum_{\forall j} C_o\left({\mathbf{q}_i}^k, {\mathbf{q}_j}^{k+1}\right) \cdot p\left({\mathbf{q}_i}^k, {\mathbf{q}_j}^{k+1} \mid {\mathbf{q}_c}^k, \varepsilon_k, {\mathbf{q}_c}^{k+1}, \varepsilon^{k+1}\right) \tag{16}$$

where ${\mathbf{q}_i}^k$ is each of the i possible states at path step k, ${\mathbf{q}_j}^{k+1}$ is each of the j possible states at path step k+1, and $C_o({\mathbf{q}_i}^k, {\mathbf{q}_j}^{k+1})$ is the deterministic cost of traveling from ${\mathbf{q}_i}^k$ to ${\mathbf{q}_j}^{k+1}$ (see Figure 7)[1].

Since we are assuming a low uncertainty rate ($\alpha_u < 0.1$), we can make additional simplifications that transform (16) into:

$$E\left[C\left(({\mathbf{q}_c}^k, \varepsilon^k), ({\mathbf{q}_c}^{k+1}, \varepsilon^{k+1})\right)\right] = a \cdot C_E(\mathbf{q}_c^k, \varepsilon_k) + b C_E(\mathbf{q}_c^{k+1}, \varepsilon^k) \tag{17}$$

where a and b are constants determined by the relative position of ${\mathbf{q}_c}^k$ and ${\mathbf{q}_c}^{k+1}$, and

$$C_E(\mathbf{q}_c^k, \varepsilon^k) = \sum_{\forall i} C_o\left(\mathbf{q}_i^k\right) \cdot p\left(\mathbf{q}_i^k \mid \mathbf{q}_c^k, \varepsilon^k\right) \tag{18}$$

is the expected cost of traversing cell $\mathbf{q}_c^k$ if the uncertainty at this location is ε^k. Therefore,

$$C(\mathbf{r}^k, \mathbf{r}^{k+1}) = a \cdot C_E(\mathbf{q}_c^k, \varepsilon_k) + b C_E(\mathbf{q}_c^{k+1}, \varepsilon^k)\,. \tag{19}$$

---

[1] As mentioned previously, we assume a discretized grid of states corresponding to a known map.

The planner used for planning with uncertainty is a modified version of A* in 3-D in which the successors of each state are calculated only in a 2-D plane, and state dominance is used to prune unnecessary states.

The planner works as follows: we have a start location $\mathbf{q}^0$ with uncertainty ε^0, an end location $\mathbf{q}^f$ with uncertainty ε^f, and a 2-D cost map C. We form the augmented state variable $\mathbf{r}^0 = (\mathbf{q}^0, \varepsilon^0)$ and place it in the OPEN list. States in the OPEN list are sorted by its expected total cost to the goal:

$$C_{ET}(\mathbf{q}^k, \varepsilon^k) = C_E\left((\mathbf{q}^0, \varepsilon^0), (\mathbf{q}^k, \varepsilon^k)\right) + h_E(\mathbf{q}_k, \mathbf{q}_f) \tag{20}$$

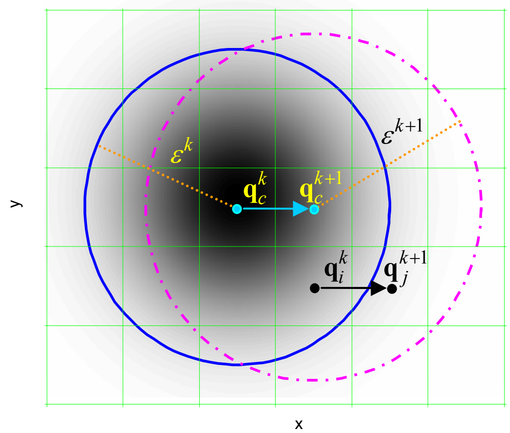


Figure 7. $p\left(\mathbf{q}_i^k \mid \mathbf{q}_c^k, \varepsilon_k\right)$ and the state transitions from $\mathbf{r}^k = (\mathbf{q}_c{}^k, \varepsilon^k)$ to $\mathbf{r}^{k+1} = (\mathbf{q}_c{}^{k+1}, \varepsilon^{k+1})$

where $C_E\left((\mathbf{q}^0, \varepsilon^0), (\mathbf{q}^k, \varepsilon^k)\right)$ is the expected cost of the best path from the start location $\mathbf{r}^0 = (\mathbf{q}^0, \varepsilon^0)$ to the current state $\mathbf{r}^k = (\mathbf{q}^k, \varepsilon^k)$, and $h_E(\mathbf{q}_k, \mathbf{q}_f)$ is the heuristic of the expected cost from the current state to the goal . The heuristic used is a function of the Euclidean distance between $\mathbf{q}_k$ and $\mathbf{q}_f$.

The state $\mathbf{r}^k$ with lowest expected total cost to the goal is popped from the OPEN list. Next, $\mathbf{r}^k$ is expanded. To determine if a successor $\mathbf{r}_j^{k+1} = (\mathbf{q}_j^{k+1}, \varepsilon_j^{k+1})$ should be placed in the OPEN list, we use state dominance as follows: a state $\mathbf{r}_j^{k+1}$ is placed in the OPEN list if

- No states with (x,y) coordinates $\mathbf{q}_j^{k+1}$ have been expanded, or

- No other state in the OPEN with the same (x,y) coordinates $\mathbf{q}_j^{k+1}$ that has lower expected cost from the start $\mathbf{r}_0$ and lower uncertainty than ε_j^{k+1}

In other words, a state is only expanded if it can provide a path with lower uncertainty or lower cost from the start location. Additionally, since ε is a function of $\mathbf{q}$, the successors of $\mathbf{r}^\mathbf{k}$ may be calculated in the 2-D workspace defined by $\mathbf{q}^\mathbf{k}$, instead of the full 3-D configuration space defined by $\mathbf{r}^\mathbf{k}$ (See Figure 8).

As the states are placed in the OPEN list, their uncertainty is updated using (13), and the expected cost of $\mathbf{r}_j^{k+1}$ is updated according to (17) and (20). However, if any states within 2σ in the uncertainty region of $\mathbf{r}_j^{k+1}$ are labeled as obstacles then the expected total path cost of $\mathbf{r}_j^{k+1}$ is set to infinity, thereby preventing any further expansion of that state.

As in A*, the process described above is repeated until a cell $\mathbf{r}^{k+1}$ with the same (x,y) coordinates as the goal position ($\mathbf{q}^{k+1} = \mathbf{q}^f$), and uncertainty less than the target uncertainty ($\varepsilon^{k+1} \leq \varepsilon^f$) is popped off the OPEN list. The path connecting the backpointers from $\mathbf{r}^{k+1}$ to $\mathbf{r}^0$ is the optimal path between $\mathbf{q}^0$ and $\mathbf{q}^f$ with a final uncertainty lower than ε^f. If the OPEN list becomes empty and no such state has been found, then there is not a path between $\mathbf{q}^0$ and $\mathbf{q}^f$ such that the final uncertainty is lower than ε^f.

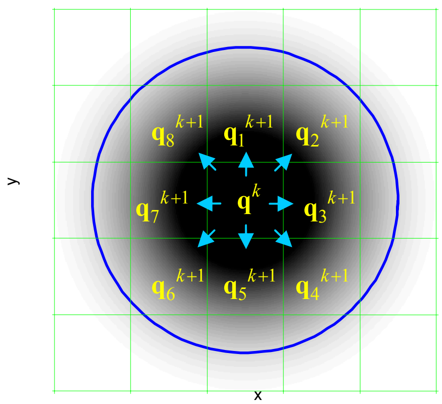


Figure 8. Successors of state $\mathbf{q}^k$

Simulation Results

The following is an example of our planner applied to the problem of finding the best path between two locations on opposite sides of Indiantown Gap, PA. The required task is to find the path with lowest cost from the start to the end location, with a final uncertainty smaller than 500 m, and which avoids all known obstacles in the map (within 2σ). The cost map used is defined as the mobility cost of traveling through each cell in the map, and is proportional to the slope of the terrain. Slopes greater than 30° are considered obstacles. This example shows the results of using our planner to calculate a path with uncertainty rates of 0%, 2% and 4% of distance traveled, and uses the results of a Monte Carlo simulation to illustrate the importance of planning with uncertainty.

In Figure 9(a) we can see the results when uncertainty is not being considered. Figure 9 (b) shows the resulting path when we use our planner and a propagation model with an uncertainty rate of 2%. Figure 9 (c) shows the resulting path when the uncertainty rate is increased to 4%.

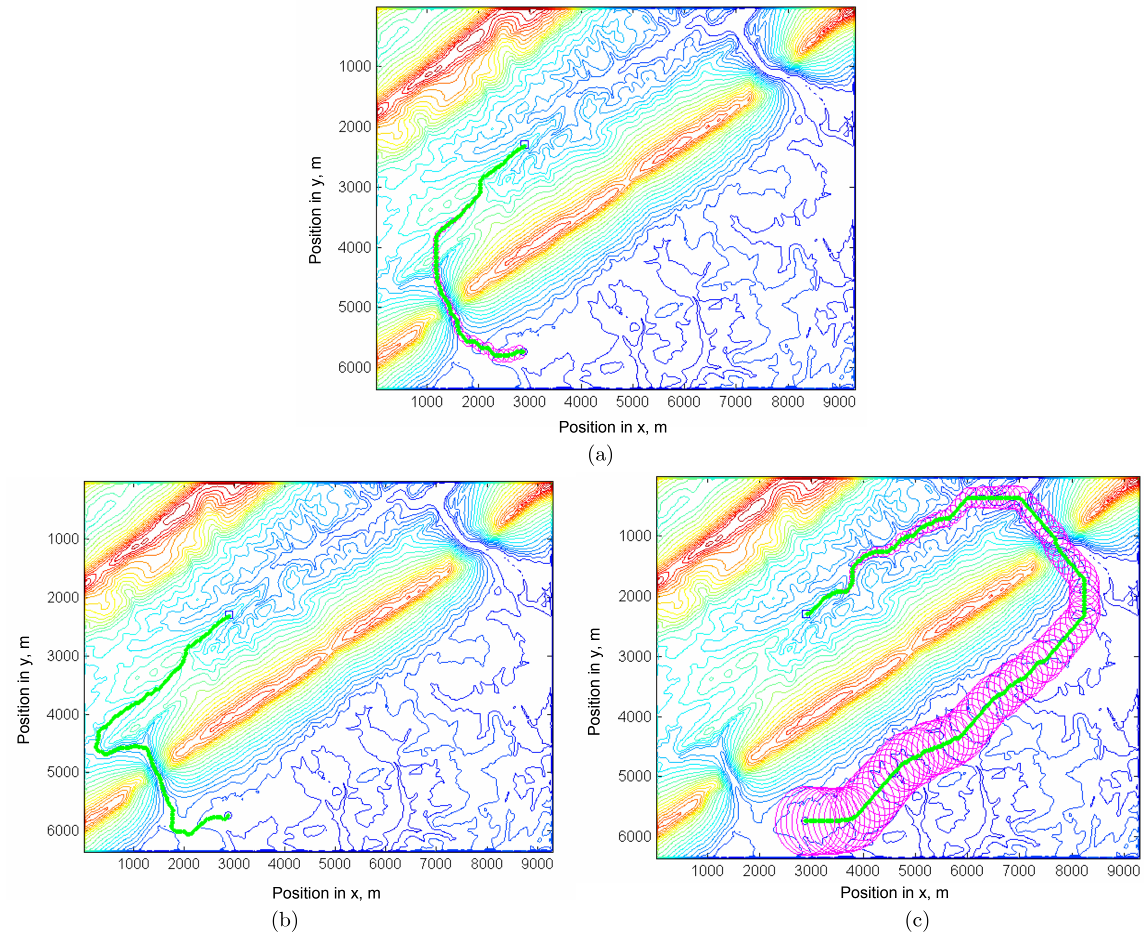


Figure 9. Comparison between paths found without uncertainty (a) and with uncertainty rates of 2% (b) and 4%. (c)

In order to understand the importance of planning with uncertainty, a set of Monte Carlo simulations were run for each of the previous three scenarios with and without considering uncertainty:

- Planning without considering uncertainty: we planned a path without uncertainty, and simulated executions with uncertainty rates of 0%, 2% and 4%.
- Planning considering uncertainty: we planned and simulated executions with uncertainty rates of 0%, 2% and 4%.

In both cases, we calculated the probability of collision as the percentage of simulations that resulted in a trajectory through obstacles. Table II shows the results of the simulations.

When there is no uncertainty (for example, if there is good GPS coverage throughout the path), the planner that considers uncertainty produces the same path as the planner without uncertainty (as expected).

If there is a 2% uncertainty rate in the motion model, and we fail to account for it, the average cost of the resulting path is 27705, and the probability of collision is 11% (The expected cost of this path as reported by the planner with no uncertainty is 9337). If we use our planner with uncertainty, the average cost is 19052 (30% lower). The expected cost calculated by the planner (18679), is also much closer to the average cost of the simulations, and the probability of collision is significantly lower (5%).

If there is a 4% uncertainty rate in the motion model but we do not account for it, the average cost of the resulting path is 37726, while the expected cost as calculated by the planner is still 9337. The probability of collision is now 36%. When we use the planner that considers uncertainty to solve this case, the average cost is 49727, the expected cost 42677, and the probability of collision 5%. The collision checking criterion for the planner with uncertainty is 2σ, which implies that paths with probabilities of collision of more than 15% would be rejected. Under this criterion, the path returned by the planner without uncertainty considerations is not a feasible path (36% probability of collision).

TABLE II

COMPARISON BETWEEN PLANNER WITH AND WITHOUT UNCERTAINTY

	Planner without uncertainty			Planner with uncertainty		
Uncert. rate	Expected cost	Average cost	Prob. of collision	Expected cost	Average cost	Prob. of collision
0%	9337	9353	0%	9337	9353	0%
2%	9337	27705	11%	18679	19052	5%
4%	9337	37726	36%	42677	49727	5%

Performance

The worst-case space complexity of the current implementation of the planner is $O(n_x \cdot n_y \cdot n_u)$, where n_x, n_y and n_u are the dimensions along the x, y and u directions. The average time complexity is $O(Q+R)$, where $Q = \alpha_u \cdot (n_x \cdot n_y)^2 \cdot n_u$ is the number of operations required to calculate the expected cost, $R = n_x \cdot n_y \cdot n_u \cdot \log(n_x \cdot n_y \cdot n_u))$ is the number of operations required by A* to calculate the path, and α_u is the uncertainty rate. For $\alpha_u > 0$, the Q term dominates the time complexity of the algorithm. If $\alpha_u = 0$, or the calculation of the expected value is performed beforehand, then the R term becomes the dominant one.

Our algorithm takes significant advantage of several tools available for deterministic search, namely heuristics, lazy evaluation and state dominance. In practice, the search space for the algorithm is not a the full 3-D volume defined by the state space representation $n_x \cdot n_y \cdot n_u$, but a thin volume $n_x \cdot n_y \cdot \overline{n}_u$, where $\overline{n}_u$ is the average number of propagations along the uncertainty axis (the thickness of the search space). $\overline{n}_u$ is typically much smaller than the size of the uncertainty dimension (n_u). Because the search space is just a fraction of the complete search space and expected costs are only calculated when needed (lazy evaluation), the average performance of the algorithm is much faster than that of algorithms that calculate all states in the search space.

Figure 10 shows the planning time for the algorithm for a batch of 1800 simulations in 10 random fractal worlds, with α_u varying between 1% and 10%, and $n_x = n_y$ varying from 50 to 250 cells. We can see a strong dependence on n_x, n_y, and α_u, and a significantly weaker dependence on n_u. Figure 11 shows the average number of uncertainty levels per state ($\bar{n}_u$) vs. the actual number of uncertainty levels (n_u). We can see that even though $\bar{n}_u$ does increase with n_u, it is always a small fraction of n_u. For $n_u = 100$ the average value of $\bar{n}_u$ is 3.4, and the maximum value is 7.9.

Our algorithm requires significantly fewer state expansions than processing all the states in the search space. Depending on the size of the world and the uncertainty rate the speed-up factor of our algorithm is between 8 and 80, depending on the size and characteristics of the world (see Figure 12).

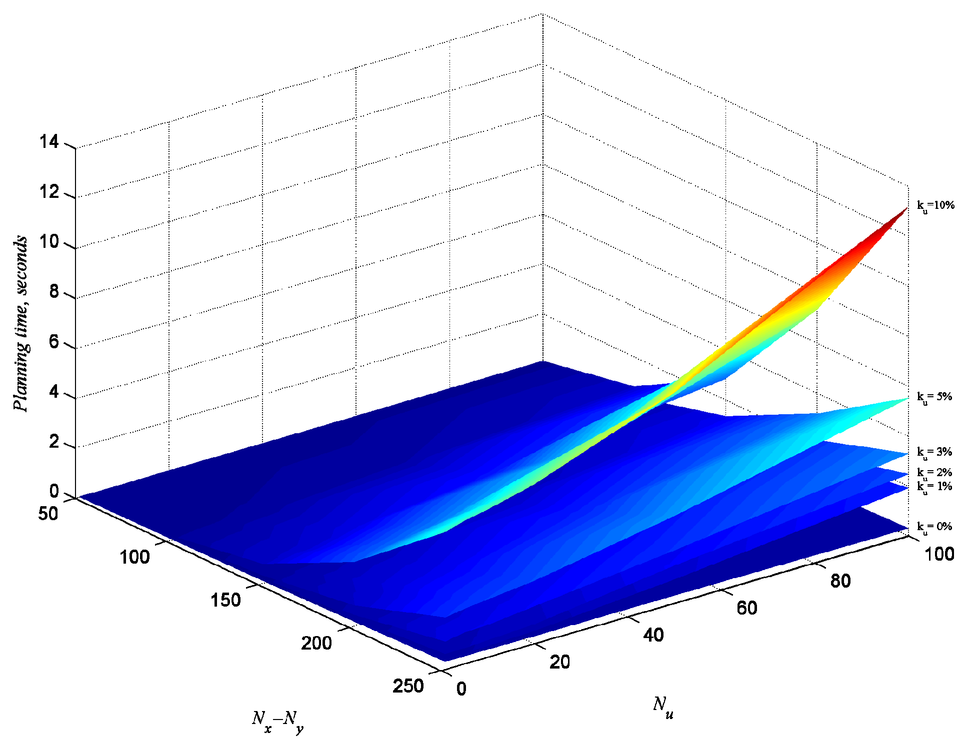


Figure 10. Planning time vs. world size (N_x=N_y) and number of uncertainty levels (N_u) for different uncertainty rates.

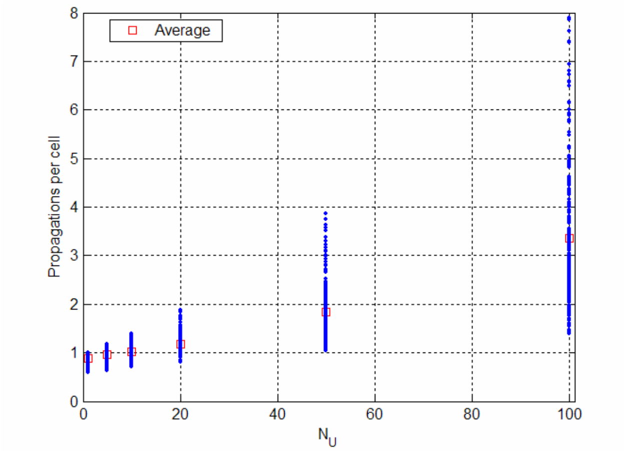


Figure 11. Propagations per cell ($\overline{n}_u$) vs. number of uncertainty levels n_u. The red squares indicate the mean value at for each value of n_u.

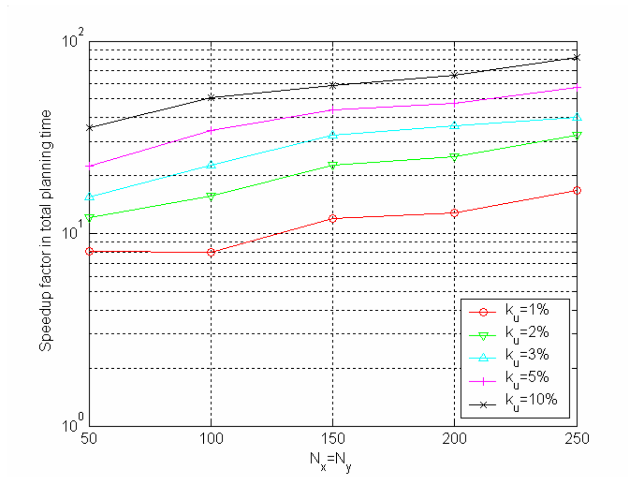


Figure 12.Speed up factor in total planning time compared to preprocessing all states

Discussion

We have introduced an efficient path planner that calculates resolution-optimal paths while considering uncertainty in position. The resulting paths are safer and have lower average costs than paths calculated without considering uncertainty.

Our approach models a stochastic problem as a deterministic search over a simplified belief space, enabling the use of a deterministic planner such as A* to solve the planning problem. Because of the characteristics of the search space and the motion model, there are significant performance gains from the use of state dominance and lazy evaluation

The main two limitations of the approach presented are that it does not use any perception information to reduce the error growth, and that it has a very restrictive probability distribution for the position of the robot. These limitations will be addressed in the following two subsections.

4.1.3 Planning with Uncertainty in Position Using Point Landmarks

If landmarks are available, the robot should use them to reduce the uncertainty in its position. Most approaches to planning with uncertainty in position using landmarks assume landmarks are unique, therefore finding a landmark immediately resolves the position of the robot. In outdoor environments, however, reliably identifying and detecting unique landmarks is very difficult. A more tractable problem is that of finding non-unique landmarks such as trees, electric poles, etc. Non-unique landmarks abound in outdoor environments, and some of them can be reliably identified in prior maps and with on-board sensors [66].

If we know that our position is within certain error distribution, the number of features that are be visible within a given detection range are significantly fewer. And if we choose our features and our positions well, we can often make sure that there is only one feature within the detection range of the robot, in which case the feature detected becomes a unique feature.

The key idea is to identify those areas in which a given point feature can be uniquely identified. We call these regions *unique detection* regions. Assuming flat terrain, 360º field of view, and a detection range R, the detection region for a point feature i such as an electric pole is a circle of radius R. If the robot is located within this region we can guarantee that only feature i can be detected.

If there is no overlap between detection regions, each circle would be a *unique detection* region. However, if there are other features within a $2R$ radius of the feature, the other features would reduce the unique detection region of the original feature. These overlapping regions would be the unique detection regions for groups of two or more features. However, multiple features are harder to detect than individual ones because of occlusions and visibility constraints. For this reason, the current approach only uses *unique detection* regions generated by single features.

Figure 13 shows the same area as Figure 5, with the unique detection regions for the electric poles in the area highlighted for a detection range R=10 meters. The dark (blue) shading shown in feature number 1 indicates *unique detection* regions. In this case, since there are no other features in a $2R$ radius the whole circular detection region is a *unique detection* region. The light shading (red) shown in parts of the detection regions of all the other features indicates a part of the detection region where more than one feature can be detected, therefore excluding that area from the *unique detection* region.

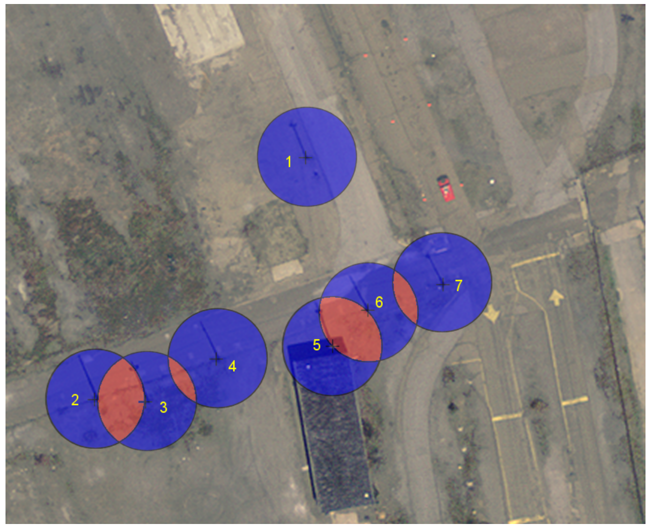


Figure 13. Unique detection regions for electric poles.

We could repeat this procedure for groups of two, three and more point features. If we were to use pairs of point features as another type of feature, the red regions in Figure 13 would be the *unique detection* regions for pairs of electric poles. However, detecting more than one feature at a time is less reliable than detecting individual features because of occlusions. For this reason, even though our algorithm can use of groups of point features, the results presented here only use individual point features as landmarks.

If all the possible locations for a configuration $\mathbf{r}^k$ are inside a *unique detection* region, we can guarantee that the feature that created the region can be detected, and that no other features will be visible within the field of view of the robot.

For practical purposes we make the simplifying assumption that a circle with radius $\varepsilon^k = 2 \cdot \sigma^k$ completely contains all possible locations on (x,y) of a given state $\mathbf{r}^k$. Therefore, if a circle of radius ε^k centered at q_c^k is completely contained within a *unique detection* region i, we can guarantee that feature i will be detected. This approximation allows us to model the detection of landmarks in a deterministic way, therefore allowing the use of deterministic search for this part of the state propagation as well. This assumption is only valid if we can *reliably* detect landmark i.

When a landmark is detected, the next state in the state propagation is set to have the same mean as the state before the detection, which is the most likely location for the robot at that step. The uncertainty at this step will be set to a small value ε_L.

While in general a single landmark detected does not reduce the uncertainty in all directions, because of the small heading error assumed for our problem, it is correct to assume that the uncertainty in x and y will be reduced to a small value. The uncertainty in heading, however, remains the same as before the detection.

Simulation Results

Figure 14 shows a sample cost map with some landmarks. Shades of gray indicate different costs in the cost map: areas with lighter color have lower cost, and areas with darker color have higher cost. Solid green areas are obstacles. The start location is a small square on the left, and the goal is a small circle on the right. As a reference, this figure also shows the shortest path that guarantees reachability of the goal for this cost map. The uncertainty rate is 10% of distance traveled. The yellow circles indicate the $\varepsilon = 2\sigma$ contours of the error distribution at each step along the path. The expected cost for this path is 8586.9 and the uncertainty at the goal is $\varepsilon_f = 2\sigma_f = 3.8$ m.

The following figures show the paths found by our approach under different constraints for uncertainty at the goal. They also illustrate the advantages of minimizing the expected cost of the path instead of minimizing the path length or the uncertainty of the path.

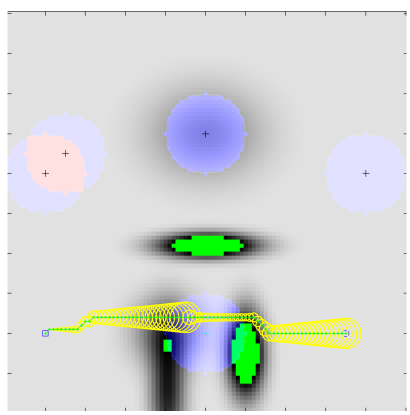


Figure 14. Planning with uncertainty rate ku=10% and using landmarks for localization (shortest path).

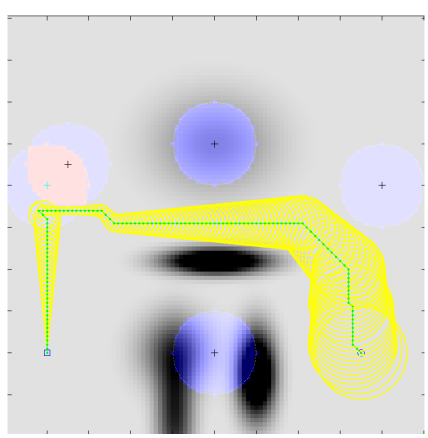


Figure 15. Planning with uncertainty rate k_u=10% and maximum uncertainty at the goal of 12 m.

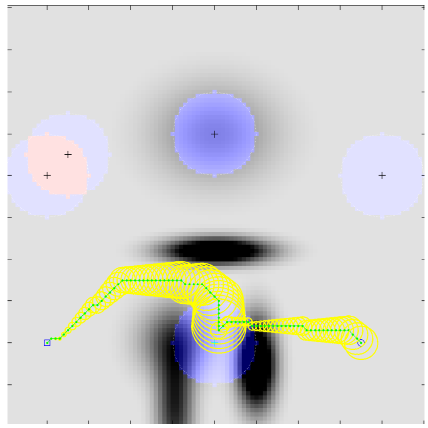


Figure 16. Planning with uncertainty rate k_u=10% and maximum uncertainty at the goal of 3.8 m

Figure 15 shows the lowest expected cost path with an uncertainty rate k_u=10% if the maximum uncertainty allowed at the goal is 12 m. The path found has an expected cost of 1485 (82% lower than the shortest path) and the uncertainty at the goal is 11 m. Because the uncertainty allowed at the goal is large, the planner has enough freedom to look for a low cost path, even if a low cost path is longer and has higher uncertainty. Only one of the localization regions can provide an improvement in the total cost, therefore the planner only includes that landmark in the final path. The planner also avoids the aliased region between the two landmarks on the left, and localizes only with the leftmost landmark.

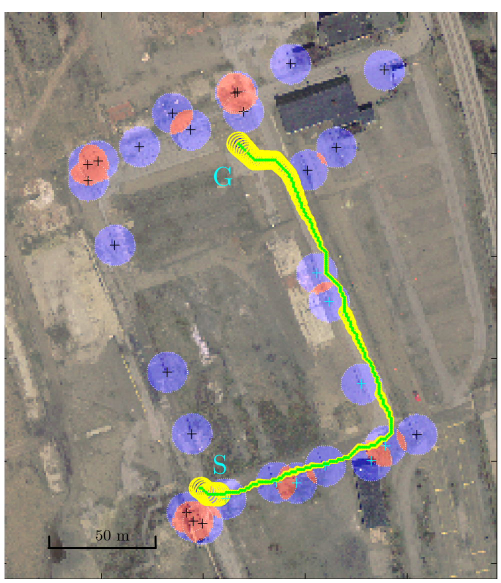
If the maximum uncertainty allowed at the goal is small, the planner trades off lower cost solutions in order to satisfy the uncertainty constraint. Figure 16 shows the lowest expected cost path when the maximum uncertainty allowed at the goal is reduced to 3.8 m. Even with a maximum uncertainty at the goal equal to that of the shortest path there can be significant advantages in minimizing the expected cost instead of the uncertainty or the path length. The expected cost is now 4710 (still 47% lower than the shortest path) and the final uncertainty is ε =3.8 m. Although the last segment of the path is the shortest path for that segment, the first segment is able to look for a less expensive path than the shortest path and the resulting path is significantly less costly than the shortest path.

Field Tests

In order to validate the results experimentally the following field test was carried out on the e-gator autonomous vehicle shown in Figure 17: a path was planned between a location S and a location G, assuming initial uncertainty σ=2.5m, uncertainty rate of 5% of distance traveled and maximum uncertainty of 10 m, using electric poles for localization (Figure 18). Notice how the path follows a road in order to minimize the expected cost along the path (instead of just minimizing the length of the path). The path also visits detection regions as needed to maintain a low cost path within the uncertainty constraints, and avoids narrow areas that could not be safely avoided if the position of the robot is not accurately known.

Figure 17. E-gator autonomous vehicle used for testing and electric poles used for localization at test site. The vehicle equipped with wheel encoders and a KVH E- core 1000 fiber-optic gyro for dead reckoning, and a tilting SICK ladar and onboard computing for navigation and obstacle detection

Also notice that the final uncertainty of the path is relatively high ($\varepsilon = 5.4$ m). This is because the maximum uncertainty allowed was set to 10 meters, and the planner will only try to reduce the uncertainty if the reduction in uncertainty will reduce the expected cost of the path. In the last segment the path is going through a large paved area and there is no increase in cost because of the higher uncertainty. For this reason, the planner does not try to detect any features in last 50 meters of the path.



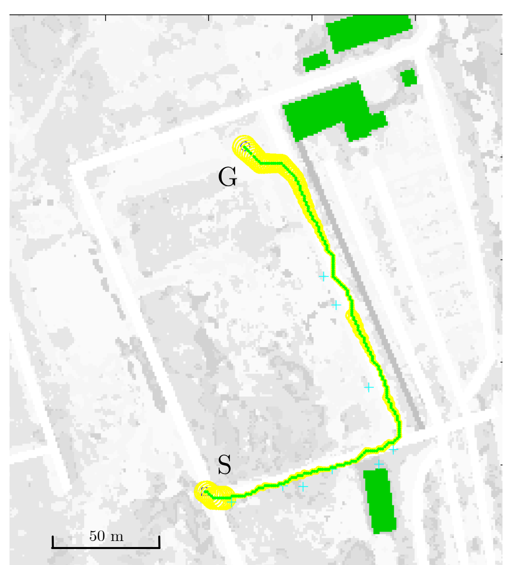


Figure 18. Path planned assuming initial uncertainty σ=2.5m, uncertainty rate of 5% of distance traveled and maximum uncertainty of 10 m. The expected cost of the path is 3232, and the final uncertainty is σ=2.7m. Left: aerial image and unique detection regions. Right: cost map used.

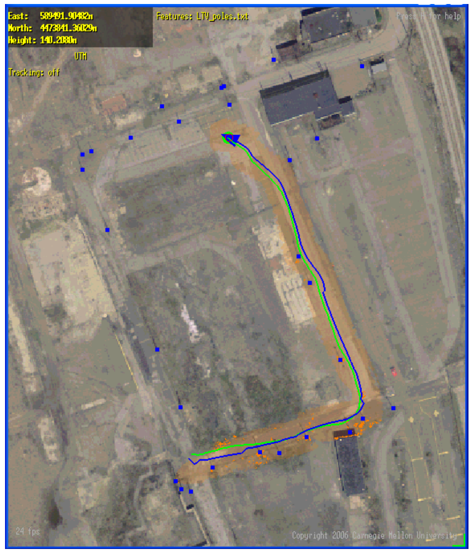


Figure 19. Path planned and executed without GPS. Blue dots show the location of landmarks. The blue line is the position estimate of the Kalman filter on the robot and the green line is the position reported by a WAAS differential GPS with accuracy of approximately 2 meters (for reference only).

Figure 19 shows the path executed by the robot. The blue line is the position estimate of the robot according to the onboard Kalman filter that combines the dead reckoning sensors and the landmark detections (no GPS). The green line shows the position estimate according to the GPS (shown as a reference only). Notice how the blue line stays very close to the GPS estimate, and jumps in a few places after detecting a landmark.

Discussion

The approach presented here allows robust and efficient navigation without GPS. it uses landmarks in the environment that have been manually identified in a high-resolution prior map to reduce the uncertainty in the robot's position as part of the planning process. The resulting path minimizes the expected cost along the route considering the uncertainty in the position of the robot. We have also shown experimental results of the system, showing navigation capabilities similar to those of a robot equipped with GPS. The current version of the algorithm uses light poles as its landmarks, and assumes that the landmarks will always be detected in both planning and execution.

Although algorithms in the literature (such as the one of Lazanas and Latombe [42]) claim optimality when using landmarks in a similar fashion, the optimality of the algorithm is significantly limited by the representation and the approximations made. The most limiting approximation is the inability to use landmarks when the detection range is smaller than the uncertainty of the robot. Figure 20 shows an example where by performing a local search for a landmark L it is possible to find a lower cost path than the one found by our algorithm. In order to find this solution, the algorithm would need to have the ability to represent the local search for a landmark. While this is an important limitation it only affects the optimality of the solution when there are no other alternatives for the path and the uncertainty when reaching the landmark is slightly larger than the detection range. If the uncertainty at the landmark is much larger than the detection range, the solution found by the local search would be too complex and unreliable to be a feasible one.

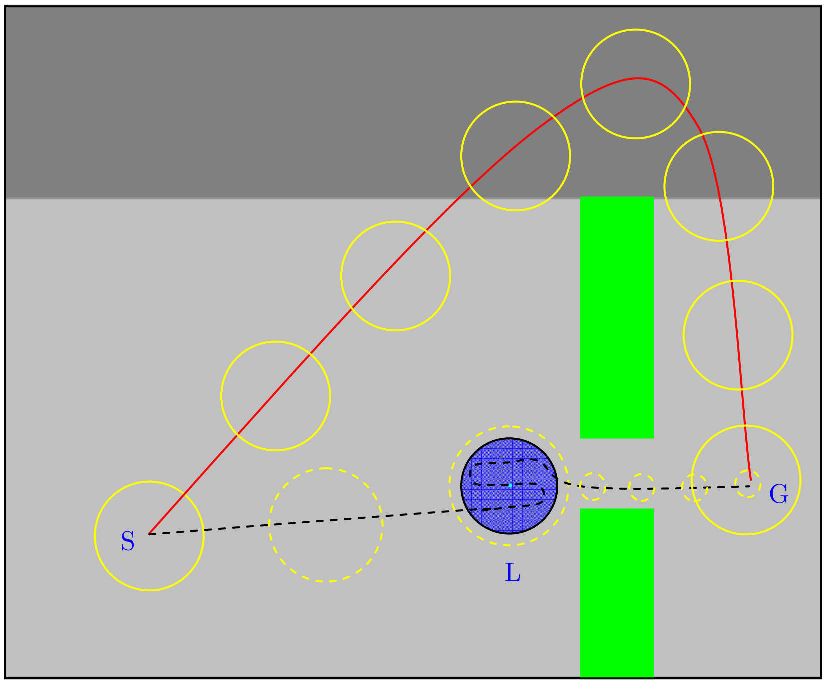


Figure 20. Limitations of the current approach. Light gray areas are low cost regions, dark gray areas are high cost regions, green regions are obstacles. The yellow circles represent the uncertainty at each step, the blue circle is the unique detection region for landmark L. The solid line is the path that our approach would find, and the dotted line is an approach that has lower cost while still guaranteeing reachability of the goal.

Another limitation is that the approach proposed here finds the path that has the lowest expected cost and guarantees the reachability of the goal within the given error bounds. However, in some scenarios, the best approach would be to have a policy instead of a path. A policy would consider the detection of features as a non-deterministic event and would produce a set of actions to be performed depending on the outcome of the detection of features. Because of this added flexibility a policy could have a lower expected cost than the path found by our approach. But finding an optimal policy would require solving a POMDP, which would be intractable to solve or would take significantly longer to plan even with an approximate solution as in [50]

4.2 Proposed Work

4.2.1 Linear Landmarks

The main limitation of using point features for localization is that they can only be reliably detected when the uncertainty in the position of the robot is smaller than the sensor range of the vehicle. Typical sensing ranges for features such as trees and electric poles are in the order of tens of meters, which only allows for a few hundred meters of travel between landmarks. For example, with a sensing range R=10 m and an uncertainty rate k_u=10%, the maximum distance that the robot can travel without finding a landmark is $D_{\max} = R / k_u = 100\,m$. If the landmarks are spaced more than $D_{\max}$ the planner cannot use them to reduce the uncertainty in the position because their detection becomes unlikely.

In contrast, other types of landmarks can be reliably detected over greater ranges. A linear landmark such as a wall, for example, can be detected reliably over all its extent. A 100 meter wall could be detected even with an uncertainty of 50 meters. Linear landmarks are also often more widely available than point landmarks: man-made structures such as walls and roads are linear landmarks that can be easily identified in aerial images. Some geographic features such as rivers and ridges also constitute linear landmarks that can be easily identified from aerial images.

Features such a tree lines can be either point or linear landmarks, depending on the size of the robot, the scale of the map, and the separation between the features. Features with a separation smaller than the range of the vehicle can be treated as either linear features or point features. If the uncertainty of the robot when approaching the features is low, they can be used as point features and can provide complete localization. If the uncertainty of the robot when approaching the features is high, they can be used as linear features, therefore extending the range of the vehicle and allowing a longer range detection than its use as point features would allow.

Error Propagation Model

In order to use linear landmarks for planning with uncertainty a different error propagation model is required. Instead of the single-parameter simplified error propagation model described before, the full uncertainty propagation model from (5) is now required. However, this model has 9 parameters for the covariance matrix P, of which 6 are independent. A complete planner that included P in its state space would require 8 dimensions (6 from P plus x and y), which is beyond the practical limits of deterministic planners.

An alternative approach is to represent the uncertainty in P by a single number and use an approach similar to the one for point landmarks. Because not all of the dimensions of P are represented, the planner is no longer complete or optimal. However, preliminary results using entropy as the number that represents the uncertainty in P have produced very good results.

Localization with Linear Landmarks

Although there are many types of linear landmarks, we will only focus in straight linear landmarks in order to simplify the problem and to maintain the Gaussian distribution assumption. In the limit, however, these straight piecewise linear landmarks can approximately represent any curve.

The information provided by this type of linear landmarks is not as rich as that of point landmarks. A linear landmark provides accurate information along one direction (perpendicular to the feature), but very little –if any- information along the direction parallel to the feature. They can also provide information about heading, but this information is often very noisy. Since the heading errors assumed by our approach are very small, we will not assume that linear features provide meaningful heading information.

In order to calculate the resulting mean and covariance using a linear feature to reduce uncertainty we proceed as follows. When the 2σ contour of a state that is being expanded approaches a linear landmark, we calculate the projection of the state distribution $p(x,y,\Sigma)$ onto the 2-D plane described by expanding the feature to all values of θ within 90 degrees of the direction normal to the landmark (at the point of contact, the feature looks identical for all values of θ). Figure 21 shows an example with the 3-D ellipsoid defined by the 2σ bounds of the covariance matrix:

$$\Sigma_{xy\theta} = \begin{bmatrix} 5^2 & 0 & 0 \\ 0 & 5^2 & 0 \\ 0 & 0 & (2.5\frac{\pi}{180})^2 \end{bmatrix} \tag{21}$$

as it approaches a linear feature located at x=10. The feature creates a plane at x=10, and the covariance matrix is then projected onto that plane. In this case, the projection is equivalent to marginalizing $p(x,y,\Sigma)$ over all values of x.

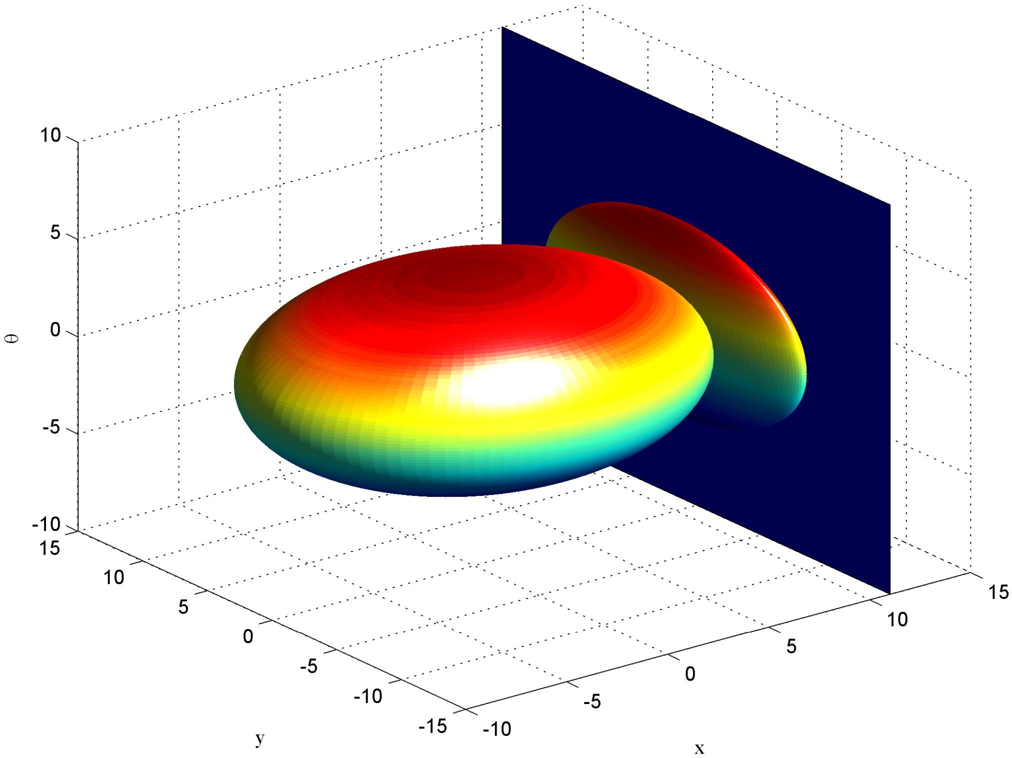


Figure 21. Localization with a linear feature: projection of the ellipsoid representing the 3-D covariance matrix in x,y and θ onto the plane defined by a linear landmark.

Figure 22 shows the same localization process on the xy plane. It shows how the uncertainty in x is reduced, but it does not represent the uncertainty in θ.

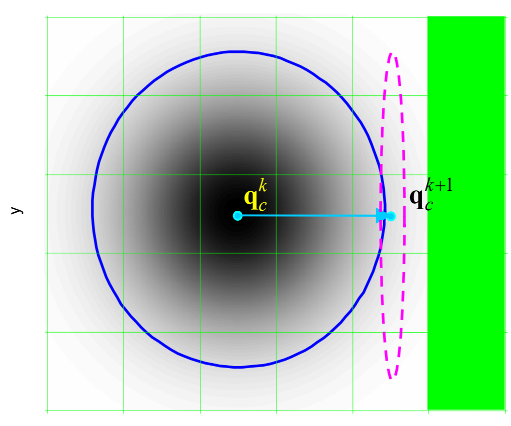


Figure 22. Localization with linear feature

Figure 23 shows a sample world, a modified version of the “peg in the hole” problem, and the path found by a preliminary version of a planner applying the principles mentioned above.

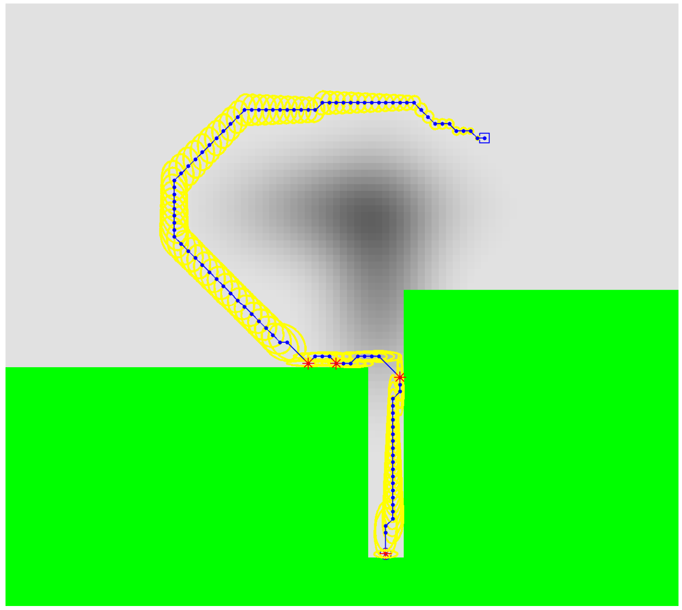


Figure 23. Modified version of the “peg in the hole” problem and the solution found by an early version of the planner using linear landmarks (the obstacles). Light gray areas are low cost regions, dark gray areas are higher cost regions and green areas are linear landmarks (walls) which are obstacles. The red stars indicate the locations where the planner uses the landmarks to reduce its uncertainty.

Discussion

Adding the ability to localize with linear landmarks significantly improves the localization abilities of the planner. This is especially true in urban settings, since linear features are most common in man-made environments.

Having a more accurate and flexible error propagation model also allows the planner to find solutions in situations where the single-parameter approximation is too limiting, such as in narrow corridors or when the initial uncertainty is not symmetric.

However, by planning in a space that does not represent the full belief space for the problem makes the planner suboptimal and incomplete. Although preliminary results show sound and near-optimal results, more analysis is required to characterize the instances in which the planner will fail.

4.2.2 Relaxing Assumptions

The approach presented here depends on a good-quality prior map and a reliable detection system. While the path found by the planner will still be a sound path if the quality of the prior map or the detection system degrades, it will no longer be optimal, and in some cases it could lead to getting lost if a landmark is not found.

The following two extensions deal with the cases in which the cost estimates need to be adjusted after the execution of the path has started because of inaccuracies in the prior map, and when the landmarks cannot be reliably detected either because of map errors or because of limitations in the detection system.

Because these extensions are unlikely to work with the planner that localizes with linear landmarks, they will be explored in isolation, using only the planner that localizes with point features.

Inaccurate Mobility Map

One of the most common causes for inaccurate cost estimates is the difference in resolution between the prior map and the onboard sensor map. While large terrain features will likely be present in both maps, smaller obstacles are often only found by onboard sensors. One way of dealing with this disparity is to increase the uncertainty in the position of the robot by a fixed amount associated with obstacle avoidance maneuvers. For example, when planning with a 0.3 m aerial map, the planner could add a 0.9 m obstacle avoidance buffer to the position of the robot. This means that the planner expects the position of the robot to deviate from the commanded path by up to 0.9 meters for obstacle avoidance.

Other causes of inaccuracy in cost estimates are outdated maps, and errors in the process of converting the prior map into a cost map. If these errors are significant, the path found by the planner will be suboptimal and could even be non traversable. The solution to this problem is to replan the path once the discrepancy is detected. The most basic (and least efficient) form of replanning the path is to repeat the planning process from scratch. The most complex (and most efficient) form of replanning is to use an algorithm that is able to repair the search graph and reuse as much as possible of the initial calculations such as D* or D*Lite. Unfortunately these algorithms require backward search and our approach derives significant performance gains from the use of forward search. Furthermore, the use of linear landmarks and its full uncertainty propagation require forward search in order to produce sound results. We will explore intermediate approaches that allow replanning using forward search while allowing greater efficiency than replanning from scratch.

Inaccurate Landmark Map or Landmark Perception

The approach presented in the previous sections assumed that the probability of detecting a given landmark once the robot is within its unique detection region is one. A more realistic approach would be to allow the probability of detecting a landmark to be less than one. These imperfect landmarks can be caused either by inaccuracies in the map or by failures in the detection system. The simplest approach to imperfect landmarks would be to replan the path if a landmark is not found. However, the optimality of our approach relies on the assumption that landmarks will be detected. If landmarks cannot be reliably detected the model of deterministic transitions used to localize the robot is no longer valid.

POMDPs, on the other hand, handle imperfect landmarks gracefully, albeit taking much longer to solve the problem. While an exact solution of the POMDP created when landmarks are imperfect would be intractable, some approximate solutions may produce results in an amount of time that would justify considering them.

One possible approach to deal with imperfect landmarks is PAO*, proposed by Ferguson *et al* [15]. In this approach most of the states in the environment have deterministic costs with the exception of a few *pinch points*. Pinch points are states that are important to the solution but whose costs are unknown (such as doors in an office environment). Instead of solving the complete POMDP problem defined by the deterministic plus non-deterministic states, PAO* has a high level graph of the unknown states and uses heuristics to determine which nodes to expand. When a node is expanded, a 2-D deterministic search is performed between the nodes involved. Using this combination of heuristics, deterministic states and pinch points, PAO* is able to find solutions to worlds with 200x200 deterministic states and 10 pinch points in less than one second on average.

Using PAO* with planning with uncertainty would entail replacing the 2-D deterministic searches with paths planned considering uncertainty. This is a feasible approach as long as the number of landmarks is low, since PAO* scales poorly with the number of pinch points.

Another possible approach is Probabilistic Planning with Clear Preferences (PPCP), by Likhachev and Stentz [43]. Like PAO*, PPCP also solves the underlying POMDP through a series of 2-D deterministic searches. PPCP is an anytime algorithm that is applicable for problems in which it is clear what values of the missing information would result in the best plan. Planning with uncertainty in position is one of such problems, as the preferred outcome of each sensing attempt is to detect the landmark.

In order to create a version of PPCP that considers uncertainty in position, each 2-D search has to be replaced by a path planned considering uncertainty. Because the searches considering uncertainty are slower than 2-D deterministic searches, the algorithm is likely to run slower as well. However, the main problem for the performance of the resulting planner is the fact that PPCP uses backward search to find its solutions. While backward search can be implemented in some of the algorithms presented in this proposal, this would reduce many of the performance gains obtained through state dominance and lazy evaluation.

Discussion

We proposed two important extensions to the planner with uncertainty in position. These extensions allow the planner to handle more complicated and less ideal worlds at the expense of additional computational requirements.

The main limiting factor of both extensions is that the proposed approaches to replanning and planning with imperfect landmarks require backward search. Due to the nature of the problem, if backward search is implemented in the planner with uncertainty we expect a significant performance reduction. This combined with the additional computational requirements of these approaches make it unlikely that the resulting planner will be nearly as fast or efficient as the current implementation. However, it is possible that through the understanding of the techniques and assumptions required for replanning and planning with imperfect landmarks a solution can be devised that preserves the performance of the current implementation and includes some of the benefits of replanning and/or planning with imperfect landmarks.

5 Research Plan and Schedule

5.1 Schedule

October – January: Relaxing Assumptions

- Implementation of PPCP for planning with uncertainty in position to deal with imperfect landmarks.
- Field Tests with imperfect landmarks
- Implementation of replanning
- Field Tests for replanning

February – May: Planning with Linear Landmarks

- Implementation of a planner that uses linear landmarks
- Implementation of a feature detector for linear features
- Field tests for navigation with linear landmarks

June – August: Write Thesis

September: Defend Thesis

5.2 Performance Evaluation

5.2.1 Simulated Experiments

We will perform a quantitative analysis of the algorithms presented by running simulated environments that measure the performance of the algorithm in areas such as planning time and expected cost of the solution. To determine the typical planning time performance of the algorithm we will create simulated worlds with varying characteristics in size, number of landmarks, uncertainty rate and complexity of the terrain. To determine expected cost of the solution we will run Monte Carlo simulations with a more complex uncertainty propagation model and see how the simplified error propagation model compares.

We will compare the algorithms presented with the results obtained when planning without considering uncertainty, and with algorithms that attempt to localize on execution without previously planning a path that considers uncertainty and landmarks.

We will also do a qualitative analysis of representative examples and configurations that show the strengths and weaknesses of the approach.

5.2.2 Field Experiments

In order to validate experimentally the results presented here, we will run tests that show the ability of the approach to navigate using point landmarks, linear landmarks, imperfect cost estimates and imperfect landmarks.

6 Expected Contributions

Our approach is expected to provide the following contributions to the field of robotics

- The first navigation system to reliably and efficiently plan and navigate without GPS in an outdoor environment using point or linear landmarks for localization.
- A near-real-time optimal approach for planning with uncertainty in position using reliable point landmarks.
- The first approach to planning with uncertainty in position for outdoor robots to be experimentally validated in a robotic platform
- A better approach to use high-resolution prior maps when GPS is not available or when the maps are not accurately registered.
- The first approach to planning with uncertainty in position using linear landmarks in continuous-cost domains.
- An any-time approach to planning with uncertainty in position using imperfect point landmarks

Acknowledgments

We would like to thank Max Likhachev, David Ferguson and Nick Roy for the many thorough discussions and exchanges of ideas that prepared the way for the approaches presented here. We would also like thank Marc Zinck, Freddie Dias, Matthew Sarnoff and Eugene Marinelli for developing and maintaining the code that enabled the field tests.

This work was sponsored by the U.S. Army Research Laboratory under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012). The views and conclusions contained in this document do not represent the official policies or endorsements of the U.S. Government.

7 References

[1] E Abbott, “Land-vehicle navigation systems: An examination of the influence of individual navigation aids on system performance,” Ph.D. Dissertation, Stanford University, March 1997

[2] B. Bonet and H. Geffner, "Planning with incomplete information as heuristic search in belief space," in Proceedings of the 6th International Conference on Artificial Intelligence in Planning Systems (AIPS), pp. 52-61, AAAI Press, 2000.

[3] B. Bouilly, T. Siméon, and R. Alami. “A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty”. In Proc. of the IEEE Int. Conf. on Robotics and Automation, volume 2, pages 1327--1332, Nagoya (JP), May 1995.

[4] B. Bouilly. “Planification de Strategies de Deplacement Robuste pour Robot Mobile”. PhD thesis, Insitut National Polytechnique, Tolouse, France, 1997

[5] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 1, 2002, pp. 540—545.

[6] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. “Estimating the absolute position of a mobile robot using position probability grids”. In Proc. of the Thirteenth National Conference on Artificial Intelligence, 1996

[7] W. Burgard, D. Fox, and D. Hennig. “Fast grid-based position tracking for mobile robots”. In Proc. of the 21th German Conference on Artificial Intelligence, Germany. Springer Verlag, 1997

[8] W. Burgard, D. Fox, and S. Thrun. “Active mobile robot localization”. In Proceedings of IJCAI-97. IJCAI, Inc., 1997

[9] A. Cassandra, L. Kaelbling, and M. Littman, “Acting optimally in partially observable stochastic domains”. In Proceedings of the National Conference on Artificial Intelligence (AAAI), 1023—1028, 1994

[10] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo Localization for Mobile Robots,” IEEE International Conference on Robotics and Automation (ICRA99), May, 1999.

[11] C. Dima, N. Vandapel, and M. Hebert, "Classifier Fusion for Outdoor Obstacle Detection," International Conference on Robotics and Automation, IEEE, April, 2004

[12] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. “A solution to the simultaneous localization and mapping (SLAM) problem”. IEEE Transactions on Robotics and Automation, 2001

[13] G. Dudek, K. Romanik and S. Whitesides. “Localizing a robot with minimum travel”. In Proc. 6th ACM-SIAM Symp. on Discrete Algorithms, 437-- 446, 1995

[14] M.E. El Najjar, P. Bonnifait, “A Road-Matching Method for Precise Vehicle Localization Using Belief Theory and Kalman Filtering”. Autonomous Robots 19(2): 173-191, 2005

[15] D. Ferguson, A. Stentz, and S. Thrun, “PAO* for Planning with Hidden State,” Proceedings of the 2004 IEEE International Conference on Robotics and Automation, April, 2004.

[16] D. Fox, “Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation”. Ph.D. Dissertation, University of Bonn, Germany, 1998.

[17] D. Fox, W. Burgard, and S. Thrun. “Active markov localization for mobile robots”. Robotics and Autonomous Systems, 1998

[18] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)., July, 1999.

[19] Th. Fraichard and R. Mermond "Integrating Uncertainty And Landmarks In Path Planning For Car-Like Robots" Proc. IFAC Symp. on Intelligent Autonomous Vehicles March 25-27, 1998.

[20] Th. Fraichard and R. Mermond. “Path Planning with Uncertainty for Car-Like Robots”. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, volume 1, pages 27-32, Leuven (BE), May 1998

[21] Th. Fraichard & Mermond, R. - Path Planning with Kinematic and Uncertainty Constraints. - In: Intelligent Autonomous Systems, pages 30-37. International Scientific Issue, Ufa State Aviation Technical University (RU), 1998.

[22] R.L. French, “Map Matching Origins, Approaches and Applications”. In Proceedings of the Second International Symposium on Land Vehicle Navigation, pages 91-116

[23] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

[24] K. Goldberg, M. Mason, and A. Requicha. “Geometric uncertainty in motion planning”. Summary report and bibliography. IRIS TR. USC Los Angeles, 1992.

[25] J.P. Gonzalez and A. Stentz, “Planning with Uncertainty in Position: an Optimal Planner”, tech. report CMU-RI-TR-04-63, Robotics Institute, Carnegie Mellon University, November, 2004.

[26] [Gonzalez and Stentz, 2005] J.P. Gonzalez and A. Stentz, "Planning with Uncertainty in Position: An Optimal and Efficient Planner," Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '05), August, 2005.

[27] A. Hait and T. Simeon, “Motion planning on rough terrain for an articulated vehicle in presence of uncertainties,” in IEEE International Conference on Robots and Systems, Osaka (Japan), November 1996.

[28] A. Haït, T. Simeon, and M. Taïx, "Robust motion planning for rough terrain navigation".Published in IEEE Int. Conf. on Intelligent Robots and Systems Kyongju, Korea, 1999.

[29] S. Honey, W. Zavoli, K. Milnes, A. Phillips, M. White and G. Loughmiller, “Vehicle navigational system and method”. US Patent 4,796,191, 1989

[30] A. Howard and L. Kitchen. Navigation using natural landmarks. Robotics and Automous Systems, 26:99--115, 1999

[31] R.R. Joshi, "A new approach to map matching for in-vehicle navigation systems: the rotational variation metric," Proceeding of IEEE Intelligent Transportation Systems, Oakland, CA, 2001.

[32] Kaelbling, L., Littman, M., and Cassandra, A. “Planning and acting in partially observable stochastic domains,” Artificial Intelligence, 1998.

[33] R.E. Kalman, "A new approach to linear filtering and prediction problems," Transactions of the ASME--Journal of Basic Engineering, vol. 82, no. Series D, pp. 35--45, 1960.

[34] A. Kelly, “Linearized Error Propagation in Odometry”, The International Journal of Robotics Research. February 2004, vol. 23, no. 2, pp.179-218(40).

[35] S. Koenig, R.G Simmons, “Real-Time search in non-deterministic domains,” In Proceedings of IJCAI-95, pp. 1660-1667, 1995

[36] S. Koenig and R.G. Simmons, “Solving robot navigation problems with initial pose uncertainty using real-time heuristic search”. In Proceedings of the International Conference on Artificial Intelligence Planning Systems, 1998, pp 154—153.

[37] J. Lalonde, N. Vandapel, and M. Hebert, “Automatic Three-Dimensional Point Cloud Processing for Forest Inventory”, tech. report CMU-RI-TR-06-21, Robotics Institute, Carnegie Mellon University, July, 2006

[38] J.C. Latombe, A. Lazanas, and S. Shekhar, "Robot Motion Planning with Uncertainty in Control and Sensing," Artificial Intelligence J., 52(1), 1991, pp. 1-47.

[39] J.C. Latombe, A. Lazanas and S. Shekhar, “Motion Planning with Uncertainty: Practical Computation of Non-maximal Preimages,” in Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, Tsukuba, Japan, 1989.

[40] J.C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers. 1990

[41] S.M. LaValle, Planning Algorithms, Book online, http://msl.cs.uiuc.edu/planning/index.html, 2004

[42] A. Lazanas, and J.C. Latombe, “Landmark-based robot navigation”. In Proc. 10th National Conf. on Artificial Intelligence (AAAI-92), 816--822. Cambridge, MA: AAAI Press/The MIT Press.

[43] M. Likhachev and A. Stentz, “PPCP: Efficient Probabilistic Planning with Clear Preferences in Partially-Known Environments,” Proceedings of the National Conference on Artificial Intelligence (AAAI), 2006

[44] T. Lozano-Perez, M. Mason and R. Taylor, “Automatic Synthesis of Fine-Motion Strategies for Robots,” in proceedings, International Symposium of Robotics Research, Bretton Woods, NH, August 1983.

[45] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish an office-navigating robot. AI Magazine, 16(2):53--60, 1995.

[46] J.M. O'Kane and S.M. LaValle, “Almost-Sensorless Localization,” in Proceedings of the 2005 IEEE International Conference on , vol., no.pp. 3764- 3769, 18-22 April 2005

[47] J. Pineau, G. Gordon & S. Thrun "Point-based value iteration: An anytime algorithm for POMDPs". International Joint Conference on Artificial Intelligence (IJCAI). Acapulco, Mexico. pp. 1025-1032. Aug. 2003.

[48] J. Pineau, “Tractable Planning Under Uncertainty: Exploiting Structure”, doctoral dissertation, tech. report CMU-RI-TR-04-32, Robotics Institute, Carnegie Mellon University, August, 2004.

[49] M. Rao, G. Dudek, and S. Whitesides. Randomized algorithms for minimum distance localization. In Proc. Workshop on Algorithmic Foundations of Robotics, pages 265–280, 2004

[50] N. Roy, and S. Thrun, “Coastal navigation with mobile robots”. In Advances in Neural Processing Systems 12, volume 12, pages 1043—1049, 1999.

[51] N. Roy and G. Gordon. “Exponential Family PCA for Belief Compression in POMDPs”. Advances in Neural Information Processing (15) NIPS, Vancouver, BC. Dec. 2002

[52] N. Roy, Finding Approximate POMDP solutions Through Belief Compression, doctoral dissertation, tech. report CMU-RI-TR-03-25, Robotics Institute, Carnegie Mellon University, September, 2003

[53] N. Roy, Private Conversation. September 1, 2004

[54] C. Scott, “Improved gps positioning for motor vehicles through map matching,” In Proceedings of ION GPS-94, 1994.

[55] D. Silver, B. Sofman, N. Vandapel, J. Bagnell, and A. Stentz, "Experimental Analysis of Overhead Data Processing To Support Long Range Navigation," IEEE International Conference on Intelligent Robots and Systems (IROS), October, 2006.

[56] R. Sim and N. Roy, "Global A-Optimal Robot Exploration in SLAM", in proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 2005.

[57] R. Simmons and S. Koenig. "Probabilistic robot navigation in partially observable environments". In Proc. of the International Joint Conference on Artificial Intelligence, 1995.

[58] R.G. Simmons, R. Goodwin, K.Z. Haigh, S. Koenig, J. O'Sullivan, and M.M. Veloso. "Xavier: Experience with a layered robot architecture". ACM magazine Intelligence, 1997

[59] B. Sofman, J. Bagnell, A. Stentz, and N. Vandapel, "Terrain Classification from Aerial Data to Support Ground Vehicle Navigation", tech. report CMU-RI-TR-05-39, Robotics Institute, Carnegie Mellon University, January, 2006

[60] B. Sofman, E. Lin, J. Bagnell, N. Vandapel, and A. Stentz, "Improving Robot Navigation Through Self-Supervised Online Learning," Proceedings of Robotics: Science and Systems, August, 2006.

[61] C. Stachniss and W. Burgard. "Exploring unknown environments with mobile robots using coverage maps". In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), 2003

[62] H. Takeda and J.C. Latombe, "Sensory uncertainty field for mobile robot navigation," Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on , vol., no.pp.2465-2472 vol.3, 12-14 May 1992

[63] H. Takeda, C. Facchinetti, and J.C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. IEEE Trans. on Pattern Analysis and Machine Intelligence, 16:1002--1015, 1994

[64] J. Tanaka, K. Hirano, T. Itoh, H. Nobuta, and S. Tsunoda, "Navigation system with map-matching method". In SAE Technical Paper Series (SAE No. 900471), 45-50. Warrendale, PA: Society of Automotive Engineers, 1990.

[65] S. Thrun, D. Fox, and W. Burgard. "A probabilistic approach to concurrent mapping and localization for mobile robots". Machine Learning and Autonomous Robots (joint issue), 1998

[66] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural Terrain Classification using 3-D Ladar Data," IEEE International Conference on Robotics and Automation, April, 2004.

[67] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(3):193--205, March 1997