The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Using Linear Landmarks for Path Planning with Uncertainty in Outdoor Environments

Juan P. Gonzalez and Anthony Stentz, *Member, IEEE*

Abstract— **This paper presents two new approaches that enable the use of linear landmarks for planning paths with uncertainty in position in outdoor environments.**

The first approach uses a combination of forward simulation and entropy to reduce the dimensionality of the search space, while still preserving most of the information required to propagate a full covariance matrix.

The second approach adds incremental binning to improve the quality of the solution while still keeping the dimensionality of the search space relatively low.

These approaches provide a better compromise of speed and quality of the solution than most existing approaches, and are able to successfully utilize linear landmarks in large outdoor environments.

I. Introduction

Planning with uncertainty in position deals with the problem of navigating autonomously when good prior maps are available but the position of the robot is not known precisely. Although the uncertainty in the position of the robot is often ignored because of the widespread availability of Global Positioning Systems (GPS), there are many scenarios where GPS is unavailable or its reliability is compromised.

If GPS is not available, the position estimate of the robot depends on dead-reckoning alone, which drifts with time and can accrue very large errors. Most existing approaches to path planning and navigation for outdoor environments are unable to use prior maps if the position of the robot is not precisely known. Often these approaches end up performing the much harder task of navigating without prior information.

An essential part of most approaches to planning with uncertainty in position is the use of landmarks. Because the dead-reckoning error is not bounded, landmarks are required to reduce the uncertainty in the position of the robot and to be able to travel long distances.

However, most existing approaches are only able to use landmarks that reduce the uncertainty in the position of the robot in both x and y directions. Because these landmarks usually look like a point in a map, they are called *point landmarks*. The main limitation of using point landmarks 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 $R=10m$ and an uncertainty rate $\alpha_u=10\%$, the maximum distance that the robot can travel without finding a landmark is $D_{\max} = R / \alpha_u = 100m$. If the landmarks are spaced more than $D_{\max}$ most planners 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. *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 (a poly line). A linear landmark such as a wall, for example, can be detected reliably over all its extent. 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.

The main limitation of linear landmarks is that they only provide accurate information along one direction (perpendicular to the feature), but very little –if any- information along the direction parallel to the feature.

We propose an approach called Planning with Uncertainty in Position using Linear Landmarks (PUPLL) that enables the use of linear landmarks for localization when planning paths with uncertainty in position. This approach uses forward propagation of the full uncertainty covariance matrix, combined with a binning function to reduce the dimensionality of the search space and enable faster planning. While this approach cannot guarantee optimal paths, we show that on most problems it finds solutions that are as good as those found by the best existing approaches, while having much faster planning times.

A. Related Work

Most of the existing approaches to planning with uncertainty in position are limited to indoor environments. See [6] for a thorough review of exiting approaches as of 2007.

Most of the recent approaches are based on sampling-based planning techniques such as Probabilistic Road Maps

Manuscript received February 19, 2009. 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.

J. P. Gonzalez is with General Dynamics Robotic Systems, Pittsburgh PA, 15221 USA (e-mail: jpgonzal@gdrs.com).

A. Stentz is with the Robotics Institute at Carnegie Mellon University, Pittsburgh PA, 15213 USA (e-mail: axs@cmu.edu).

(PRMs) or Rapidly exploring Random Trees (RRTs). PRMs and RRTs are well suited for solving some aspects of planning with uncertainty in position with full covariance propagation. Both handle multi-dimensional spaces well because of their sparse representation of the world. However, sampling-based techniques usually don't handle continuous-cost domains well, although some approaches such as hRRTs, proposed by Urmson [14], have tried to address this shortcoming with some success. Even with extensions such as hRRTs, sample-based planners are usually unable to provide the same optimality and completeness guarantees that grid-based planners do.

Missiuro and Roy's [11] approach addresses the problem of uncertainty in the position of obstacles, by replacing obstacle's vertices with Gaussian distributions. The approach, however, does not handle continuous-cost domains and does not model the uncertainty accrued while moving. Burns and Brock [4] extend Missiuro and Roy's approach to enable the use of an arbitrary obstacle representation and many degrees of freedom. Their implementation is targeted to articulated robots. Alterovitz *et al.* [1][2] introduce the idea of a Stochastic Motion Roadmap (SMR), to steer surgical needles. The trajectory of these needles is uncertain to some extent because the tissue in which they are inserted is not completely known. However, the problem is formulated as one of avoiding binary obstacles while considering the non-holonomic constraints of the needle. Similarly, Pepy and Lambert [12] use RRTs to find safe paths considering uncertainty. Their approach uses a Kalman Filter to estimate the evolution of uncertainty and to model the effect of localization in the planning process.

Melchior *et al* [9][10] have also recently proposed an approach that uses RRTs to propagate a full covariance matrix while planning with uncertainty in position for outdoor environments. This approach is able to handle multiple hypotheses in the outcome of an action as it uses an approach similar to a particle filter to model the evolution of uncertainty during the search process. Because of this model, the planner is not limited to solving problems with low uncertainty rate. Although the approach includes a biasing term to avoid high cost regions using hRRTs [14] its results don't attempt to optimize cost. The goal is more to minimize the end error rather than to minimize the expected cost. The approach is also more tailored for short traverses, and it is unclear how well it would scale to larger environments.

B. Planning with Uncertainty in Position (PUP)

The approach presented here extends the planner with uncertainty in position (PUP) presented in [8], which takes advantage of the low drift rate in the inertial navigation system of many outdoor mobile robots. PUP uses an isometric Gaussian distribution to model position uncertainty and uses deterministic search to efficiently find paths that minimize expected cost while considering uncertainty in position. A linear error propagation model is used, which assumes that the dominant term in the uncertainty propagation is the error in the initial heading.

In this approach a high-resolution map is translated into a cost map, 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. This map is often called a *prior map*.

PUP uses *unique detection regions* to disambiguate point landmarks. *Unique detection regions* are areas in the map where only one landmark is visible, therefore allowing non-unique landmarks to be uniquely identified for a given detection range *R*.

1) State Space Representation

The probability density function (pdf) of the error is modeled as a Gaussian distribution, centered at the most likely location of the robot at step *k*:

$$\begin{aligned} \mathbf{q}_k &= (x_k, y_k) \\ \mathbf{q}_k &: N(\boldsymbol{\mu}_k, {\sigma_k}^2) \\ p(\mathbf{q}_k) &= \frac{1}{\sqrt{2\pi{\sigma_k}^2}} e^{-\frac{1}{2}\frac{(\mathbf{q}_k-\boldsymbol{\mu}_k)^T(\mathbf{q}_k-\boldsymbol{\mu}_k)}{{\sigma_k}^2}} \end{aligned} \tag{1}$$

where $\boldsymbol{\mu}_k = (\mu_{x_k}, \mu_{y_k})$ 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 at step *k*. PUP defines:

$$\varepsilon_k = 2 \cdot \sigma_k \tag{2}$$

such that the boundary of the uncertainty region can be modeled as a disk centered at μ_k with radius ε_k. 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.

Under these assumptions, the augmented state vector

$$\mathbf{r} = (\boldsymbol{\mu}, \varepsilon) \tag{3}$$

defines a 3-D configuration-uncertainty state space, which is also a complete *belief space* [3].

In the cost map, the cost C_o of a cell **q** is defined as the cost to travel from the center of the cell to its nearest edge. PUP extends the idea of this 2-D cost map into the 3-D configuration space by defining the cost to move from the center of the 3-D cell **r** to its nearest edge. This cost can be expressed as:

$$C_{\mathbf{r}}(\mathbf{r}_k) = C_{\mathbf{r}}(\mu_k, \varepsilon_k) = \sum_i C_o(\mathbf{q}_i) p_{\mu_k,\varepsilon_k}(\mathbf{q}_i) \tag{4}$$

where $C_{\mathbf{o}}(\mathbf{q}_i)$ is the deterministic traversal cost as defined by the 2-D cost map at location $\mathbf{q}_i$.

2) Uncertainty Propagation

Outside of Unique Detection Regions

Outside of *unique detection regions* the position estimate of the robot is calculated using dead-reckoning. For traverses up to a few kilometers and with a good dead-reckoning system, the dominant term in the error propagation is the error in the initial heading, which increases linearly with distance traveled. Based on this assumption, PUP uses the following model to propagate uncertainty:

$$\varepsilon_k = \varepsilon_{k-1} + \alpha_u d(\mu_{k-1}, \mu_k) \quad (5)$$

where α_u is the uncertainty accrued per unit of distance traveled, μ_{k-1} is the previous position along the path, ε_{k-1} is the uncertainty at the previous position, and $d(\mu_{k-1}, \mu_k)$ is the distance between the two adjacent path locations μ_{k-1} and μ_k. The uncertainty rate α_u is typically between 0.01 and 0.1 (1% to 10%) of distance traveled.

Inside Unique Detection Regions

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

For practical purposes PUP makes the simplifying assumption that the disk with radius $\varepsilon_k = 2 \cdot \sigma_k$ completely contains all possible locations on (*x,y*) of a given configuration $\mathbf{r}_k = (\mu_k, \varepsilon_k)$. Therefore, if the disk of radius ε_k centered at μ_k is completely contained within a *unique detection* region *i,* we assume that the configuration $\mathbf{r}_k$ is inside the *unique detection region.* As such, we can guarantee that feature *i* will be detected and assume that the uncertainty ε_k will be reduced to a small amount δ.

3) Using Deterministic Search to Plan with Uncertainty in Position

The belief space and 3-D cost map defined above define a graph with positive traversal costs. If the transitions between states are deterministic, we can use deterministic search to find the lowest cost path between any two points in the graph.

We use the following assumptions to ensure that the transitions between states are deterministic. In areas outside *unique detection regions*, planning takes place without sensing landmarks, and can be modeled as deterministic transitions in belief space. If landmarks can be reliably detected, then the areas inside *unique detection regions* can also be modeled as deterministic transitions, as the detection of landmarks is guaranteed. In the transitional areas that are not completely contained within *unique detection regions* the detection of landmarks cannot be either guaranteed or ruled out. However, by assuming that landmarks will only be detected when the uncertainty contour is completely contained within the detection region, we can still model these regions in a deterministic fashion, at the expense of having an overly conservative approach.

We search this graph using 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.

II. Planning With Uncertainty in Position Using Linear Landmarks

The approach presented here extends the planner with uncertainty in position (PUP) in order to use linear landmarks for localization.

A. Uncertainty Propagation

The main difference between point and linear landmarks is that while point landmarks reduce the uncertainty in all directions, linear landmarks usually reduce uncertainty only in one direction at a time. While for point landmarks it was sufficient to model uncertainty as an isometric Gaussian distribution, for linear landmarks it is necessary to model a more general Gaussian distribution that represents separately the uncertainty in *x*, *y* and θ.

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

$$w_v : N(0, \sigma_v{}^2) \text{ and } w_w : N(0, \sigma_\omega{}^2) \quad (6)$$

where w_v is the error in longitudinal speed v due to the longitudinal speed control, and w_ω is the error in the heading rate ω due to the gyro random walk. Using the Extended Kalman filter approximation we can then model the uncertainty propagation for these parameters as follows.

$$\begin{aligned} \mathbf{q}_{k+1} &= f(\mathbf{q}_k, \mathbf{u}_k) \\ \Sigma_{k+1} &= \mathbf{F}_k \cdot \Sigma_k \cdot \mathbf{F}_k{}^T + \mathbf{G}_k \cdot \mathbf{Q}_k \cdot \mathbf{G}_k{}^T \end{aligned} \quad (7)$$

where

$$\begin{aligned} \mathbf{q}_k &= (x_k, y_k, \theta_k) \qquad \mathbf{q}_k : N(\boldsymbol{\mu}_k, \Sigma_k) \\ \mathbf{u}_k &= (v_k, \omega_k) \\ \mathbf{Q}_k &= \frac{1}{\Delta t}\begin{pmatrix} \sigma_v{}^2 & 0 \\ 0 & \sigma_\omega{}^2 \end{pmatrix} \end{aligned} \quad (8)$$

$$\mathbf{F}_{ij} = \frac{\partial f(q_{k,i}, u_{k,j})}{\partial q_{k,i}} \quad \mathbf{G}_{ij} = \frac{\partial f(q_{k,i}, u_{k,j})}{\partial u_{k,j}} \quad (9)$$

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} 1 & 0 & -v_k \sin(\theta_k) \cdot \Delta t \\ 0 & 1 & v_k \cos(\theta_k) \cdot \Delta t \\ 0 & 0 & 1 \end{pmatrix} \\ \mathbf{G} &= \begin{pmatrix} \cos(\theta_k) \cdot \Delta t & 0 \\ \sin(\theta_k) \cdot \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \end{aligned} \quad (10)$$

This model has 9 parameters for the covariance matrix Σ, of which 6 are independent. A complete planner that included Σ in its state space would require modeling the state space as:

$$\mathbf{r} = (\boldsymbol{\mu}, \boldsymbol{\varepsilon}) \quad (11)$$

where $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_\theta)$ and $\boldsymbol{\varepsilon} = (\sigma_{xx}, \sigma_{xy}, \sigma_{yy}, \sigma_{\theta\theta}, \sigma_{x\theta}, \sigma_{y\theta})$. See [6] for a more complete analysis.

1) Using entropy to reduce the dimensionality of the search space

A planner in 9-dimensions is not practical for outdoor environments. The x and y dimensions of the planner alone can be in the order of 1000x1000=10^6 cells, and even as little as 10 cells in each of the additional dimensions would create a planner with $10^6.10^7=10^{13}$ cells.

We propose an alternative approach that performs a forward search using three-dimensional bins. We define a binning function B such that $\mathbf{r}_B(\mathbf{r}) = (\mu_x, \mu_y, B(\mathbf{r}))$, where $B(\mathbf{r})$ is a summary statistic representing uncertainty. The summary statistic implicitly defines equivalency classes for Σ, and should be selected such that it is able to separate significantly different values of Σ.

Starting with the initial state $\mathbf{r}_o$, states are expanded using the uncertainty propagation from (7) and stored in the bin determined by $\mathbf{r}_B(\mathbf{r})$. This approach does not explicitly quantize Σ, as the full covariance matrix is propagated in each state expansion. The quantization of Σ takes place indirectly, when two states within the same bin are compared. As the planner expands states from the OPEN list, each new state $\mathbf{r}_j$ to be expanded is compared with the existing states $\mathbf{r}_i$ at the same (x,y) location according to the following dominance relation:

$$(\mathbf{r}_i \trianglerighteq \mathbf{r}_j) \Leftrightarrow (x_i = x_j) \wedge (y_i = y_j) \wedge (L^*(\mathbf{r}_o, \mathbf{r}_i) \leq L^*(\mathbf{r}_o, \mathbf{r}_j)) \wedge (\Sigma_i \leq \Sigma_j) \quad (12)$$

where $L^*(\mathbf{r}_o, \mathbf{r})$ is the lowest cost path between the start state $\mathbf{r}_o$ and state $\mathbf{r}$, and $(\Sigma_i \leq \Sigma_j) \Leftrightarrow \det(\Sigma_i - \Sigma_j) \leq 0$.

When comparing two states $\mathbf{r}_i$ and $\mathbf{r}_j$ from different bins, if $\mathbf{r}_i$ dominates $\mathbf{r}_j$, $\mathbf{r}_j$ is deleted with no loss of information (since state $\mathbf{r}_i$ is better in cost and uncertainty). The covariance matrix is calculated as a forward transformation and its full precision is preserved. Likewise, if $\mathbf{r}_j$ dominates $\mathbf{r}_i$, $\mathbf{r}_i$ is deleted with no loss of information. If neither one dominates, both states should be preserved, as one has better cost, and the other one has better uncertainty. Since each one is in a different bin, both states can be preserved, and no information is lost.

When comparing two states from the same bin, a similar process takes place. If one of them dominates the other, the dominated one is deleted and no information is lost. However, if neither one dominates, both states cannot be preserved. Since there is only one bin space available, only the state with lower cost is preserved, even if its uncertainty is higher. Only in this case information is lost, and some form of quantization takes place. We call this event a *collision*, and its likelihood depends on the size of the quantization step in the uncertainty dimension and on the choice of binning function.

The extent to which quantization affects the resulting path depends greatly on the topology of the search space. For example, if the search space is homeomorphic with the xy plane, the planner is able to find the optimal solution no matter how large the uncertainty bins are. More frequently, however, the impact of quantization in the solution is seen when we try to achieve a certain uncertainty at the goal. In this case, the planner is only able to achieve such uncertainty within the tolerance of the bin size. For our experiments we use a bin size equal to the uncertainty accrued by traveling one step in the x or y direction. If the cell size is 1 meter and the uncertainty rate is 10%, the uncertainty bins are 0.1m each. While it is possible to find problems in which this resolution is not sufficient, it can be argued that such problems should not be represented on a 1-meter grid.

A natural binning function to use is entropy, which has been successfully used in related approaches such as Roy and Thrun's [13]. Entropy is proportional to the product of the major semi-axis of the covariance matrix Σ. This binning function effectively clusters together covariances with similar dimensions, and differentiates those that have significant differences in their semi-axes. In most environments this works very well, although it is not possible to guarantee that collisions will not take place.

While many other binning functions can be used, we have found that entropy produces very good results in large sets of problems analyzed. It also preserves the topology of the search space, allowing for fast queries to find neighbors.

2) Combining entropy and incremental binning

The main weakness of using entropy is the possibility of *collisions.* This happens, for example, when comparing two ellipses with identical semiaxes but that are rotated with respect to each other. In this case, the ellipses have the same entropy (and belong to the same bin), and the higher cost one is discarded even if it is not dominated.

Recently Censi [5] proposed an approach for planning with uncertainty in position that allows finding minimum time or minimum covariance paths while preserving all parameters from the covariance matrix. In theory, this approach avoids quantizing the covariance matrix by using state dominance to discard unnecessary states, and preserving all other states for a given (x,y) location. In practice, this approach requires a finite tolerance σ_{TOL} when comparing matrices, which should be selected to be as small as possible. The author utilizes a tolerance between 0.001 and 0.005 m. While this small tolerance is apparently insignificant, it has important theoretical and practical implications.

From a theoretical point of view, the need for this tolerance implies that rather than using a continuous representation of Σ, this approach is incrementally binning Σ in arbitrarily small steps and defining equivalency classes of covariances that have semiaxes within σ_{TOL} of each other.

From a practical point of view, the selection of this tolerance greatly affects the performance of the algorithm. By selecting an arbitrarily small σ_{TOL} it is possible to plan with arbitrarily high resolution in the representation of Σ. However, very small values of σ_{TOL} cause the number of states at each (*x*,*y*) location to increase rapidly, therefore significantly increasing the complexity and space requirements of the algorithm.

The main weakness of incremental binning is that the size of the search space is not known a priori, and that it can become arbitrarily large depending on the complexity of the environment and the selection of σ_{TOL}. As the search space becomes larger, the complexity of the algorithm also becomes increasingly large. This is most noticeable in continuous cost worlds, where more alternatives are available and less pruning takes place.

We propose an approach that combines entropy and incremental binning by performing a primary binning based on entropy, and then a secondary incremental binning. We define an explicit equivalence relation between covariances such that:

$$\Sigma_i \sim \Sigma_j \Leftrightarrow (\Sigma_i \leq \Sigma_j') \wedge (\Sigma_i' \geq \Sigma_j) \tag{13}$$

where

$$\Sigma' = \mathbf{P}\begin{bmatrix} (\sigma_1 + \sigma_{TOL})^2 & 0 & 0 \\ 0 & (\sigma_2 + \sigma_{TOL})^2 & 0 \\ 0 & 0 & (\sigma_3 + \sigma_{TOL})^2 \end{bmatrix}\mathbf{P}^{-1}$$

is a version of Σ in which each eigenvalue has been enlarged by σ_{TOL} (the "grow" operator described by Censi in [5]).

Rather than having only one state at each bin, each bin $\mathbf{r}_B$ is allowed to hold up to n_Q states. If there is a *collision* between two states $\mathbf{r}_i$ and $\mathbf{r}_j$ at bin $\mathbf{r}_B$, then their covariances are compared. If they are equivalent, the one with the lowest cost dominates and no information is lost up to a resolution σ_{TOL}. If they are not equivalent, the second state is added to the bin. If another state $\mathbf{r}_k$ also has a *collision* at bin $\mathbf{r}_B$, the covariance of this new state is compared with the covariances of $\mathbf{r}_i$ and $\mathbf{r}_j$. If neither $\Sigma_i \sim \Sigma_k$ or $\Sigma_j \sim \Sigma_k$ then this state is also added to bin. This process is repeated until n_Q states have been added to bin $\mathbf{r}_B$. At this point, any further states with the same bin $\mathbf{r}_B$ that have higher cost and no dominance relationship will be discarded (in a similar way as the entropy-based approach). Only at this point would a collision cause loss of information and could compromise optimality. This approach is resolution-optimal if no bins in the search exceed their capacity.

There are two design parameters that significantly affect the performance of the algorithm: n_Q and σ_{TOL}. In general, smaller values of σ_{TOL} imply higher resolution, longer execution times, and require larger values of n_Q. Finding the optimal values for these parameters is an open question. Experimentally, the following approach has shown very good results: σ_{TOL} is initially set to the uncertainty rate α_u for each bin $\mathbf{r}_B$. Every time a new state is added to the bin, σ_{TOL} is doubled. Using this combined with an n_Q of 8 usually prevents exceeding the capacity of the bins and produces results equivalent to those found by incremental binning at much smaller σ_{TOL} values.

By combining entropy and incremental binning, this approach is able to limit the size of the search space in a sound manner that produces results at least as good as those produced by entropy alone. It also preserves the topology of the search space to a great extent, which is useful for neighborhood queries and other applications.

B. Localization with linear landmarks

In order to plan with uncertainty using linear landmarks we not only need to modify the state expansion and the uncertainty propagation but we also need to define a different approach for localization. Although there are many types of linear landmarks, we will only focus on 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 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. It 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 general, localization with a linear feature can be seen as a projection of the covariance matrix onto the localizing feature, or a marginalization of $p_{\mu_k,\varepsilon_k}(\mathbf{q}_k)$ along the direction *n* locally normal to the linear feature.

During the state expansion, when a state that is being expanded approaches a linear landmark, we calculate the projection of the state distribution $p_{\mu_k,\varepsilon_k}(\mathbf{q}_k)$ onto the 2-D plane that represents the feature for 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). Fig 1 shows an example with the 3-D ellipsoid defined by the 2σ bounds of the covariance matrix 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 $p_{\mu_k,\varepsilon_k}(\mathbf{q}_k)$ over all values of x.

1) Localizing with wall-like features

The localization approach described above assumes that, on execution, the robot will move towards the feature until the feature is found. This collapses the uncertainty in the direction normal to the feature and preserves the Gaussian assumption needed by the planner. In order to model this transition as a deterministic one we need the certainty that the feature used for localization will be detected. While this assumption may not hold for some types of linear features, it does hold for wall-like features: features that will prevent the robot from moving when they are found. Examples of these features are walls, dense tree lines and possibly rivers. Walls and tree lines can be reliably detected with approaches such as the one proposed by Vandapel *et al* [15]. Rivers pose a much more complicated challenge as they can be obscured by vegetation and it may be dangerous to try to find a river by getting very close to it.

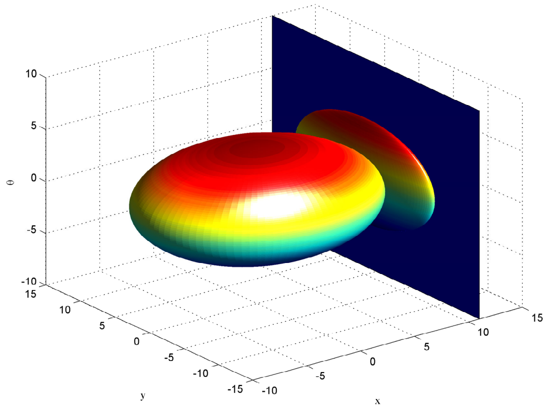

Fig 1. Localization with a linear feature: projection of the ellipsoid representing the 3-D covariance matrix in x,y and θ onto the plane defined by a linear landmark.

Fig 2 shows an example world with wall-like features and the path found by PUPLL. The robot starts with high uncertainty, which would make it costly to go through the channel before the goal. The path, therefore, follows a low cost path to a vertical wall to reduce horizontal uncertainty, and then moves down to a horizontal wall to reduce vertical uncertainty. After this second localization, the uncertainty in the robot's position is very small, but the uncertainty in its heading has not changed since we do not use the walls to improve heading accuracy. The robot then accrues additional uncertainty as it moves, but is able to pass stay within the low cost area in the channel on the way to the goal.

2) Localizing with roads

Localizing with roads poses some challenges not present with wall-like features. The most important challenge is that road detection is only reliable in limited situations. The most reliable type of road detection is that of structured roads (paved roads with lane markers). However, even in this case the detection is only reliable if the vehicle is on the road. Detecting a road when the robot is not on it is a challenging problem not yet solved by current approaches to road detection.

Considering these limitations, the approach proposed here to localize with roads assumes that roads can be used for localization only when the robot is on a road, or when other means of localization can be used to reduce the uncertainty within the width of the road. However, this limitation stem from the detection model, not from the planning approach used. If roads can be reliably detected under more broad circumstances, the limitations described above can be relaxed or removed.

Fig 3 shows a simple example of the planner using roads for localization. Notice how the uncertainty remains low in the direction perpendicular to the direction of travel as long as the robot is on a road. When the robot leaves the road the uncertainty increases rapidly.

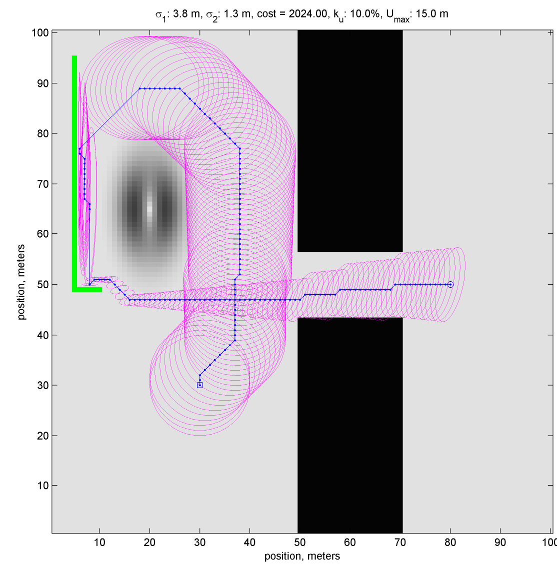

Fig 2. Path planning using wall-like features. Light gray regions are low cost regions, darker regions are higher cost regions and green areas are obstacles.

Even with this conservative approach, localization with roads is a powerful approach to planning with uncertainty in position. Because the main source of uncertainty is the error in heading and roads provide localization in the direction perpendicular to the direction of travel, it is possible to navigate very long distances without the need for any other localization approaches.

Fig 4 shows a more realistic example of the planner using roads for localization over a large area. In the example shown, the planner is able to find a path that optimizes cost and that keeps the uncertainty in position at less than 10 meters at all times over a 4.7 km path.

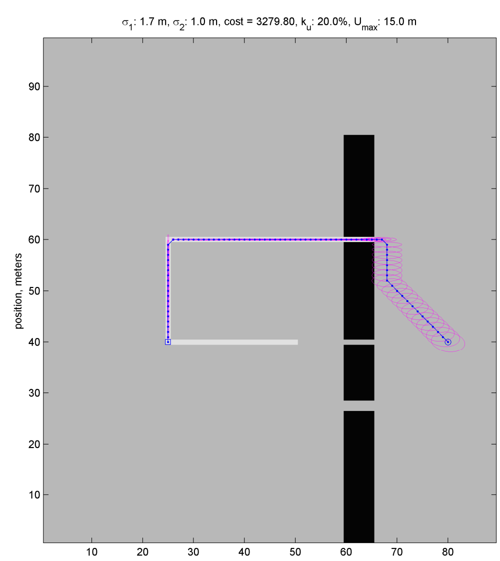

Fig 3. Path planning using roads for localization. Light gray areas are low cost roads that can be detected by the robot. Darker areas are higher cost regions.

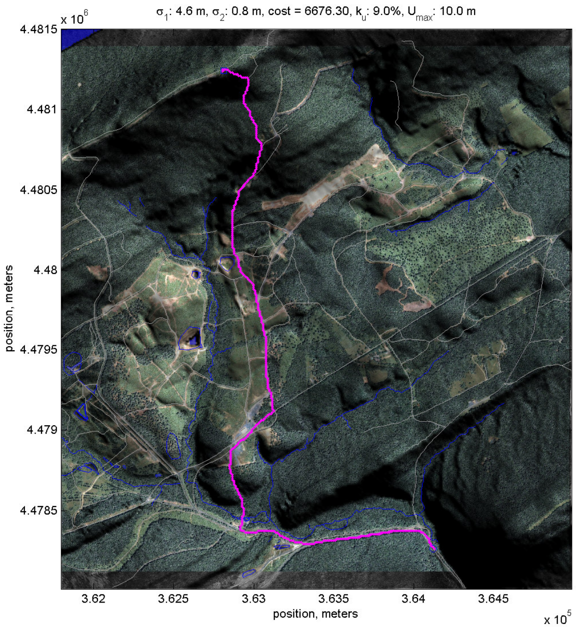

Fig 4. Localization with roads over a large area. Aerial image with shaded relief of Indiantown Gap, PA. Gray lines are roads. Blue lines are rivers. The purple line shows the path found by the planner, which minimizes expected cost and maintains the uncertainty at less than 10 meters after traveling for 4.7 km. The world size is 3.5km by 3.2 km, with 10 m cell spacing.

C. Performance

Although the computational complexity of the approaches presented here is similar to that of the 1-D PUP planner presented in [8], in practice their performance is not nearly as good. This is partly because of the increased overhead in the expected value calculations, and partly due to reduced ability to cache intermediate results.

Fig 5 shows the processing time of our approaches as well as Censi's incremental binning in simulated fractal worlds like the one in Fig 6. Our two approaches (entropy and entropy plus incremental binning) have similar performance, and are able to find a path in less than 10 seconds in worlds up to 300x300, and in a few minutes in worlds up to 500x500. Censi's approach takes between 10 and 100 times longer. The processing time for the 1-D PUP planner is also shown as a reference.

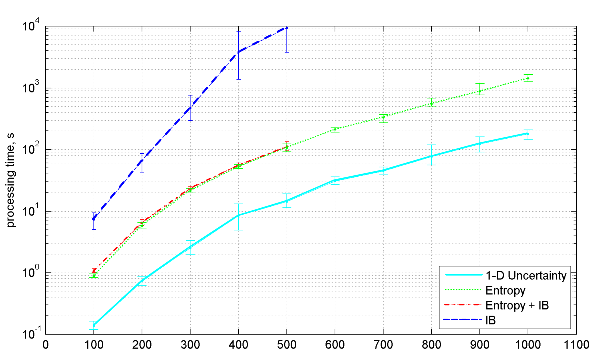

Fig 5. Processing time comparison between PUP, PUPLL-entropy, PUPLL-entropy+IB and Censi's approach (IB) for fractal worlds of sizes varying from 100x100 to 1000x1000

In the previous example, the entropy-based approach found paths as good as the paths found with Censi's or the entropy plus incremental binning approaches. While this is not uncommon, there are environments in which using entropy alone results in paths with higher costs. Fig 7 shows one of such environments, a complex world with multiple walls in different orientations, as well as high cost areas (darker areas). Notice how the planner localizes with walls of different orientations, while staying away from high-cost regions. Notice also that the localizations only take place when needed in order to reduce cost of to get through a narrow passage, instead of trying to localize continuously.

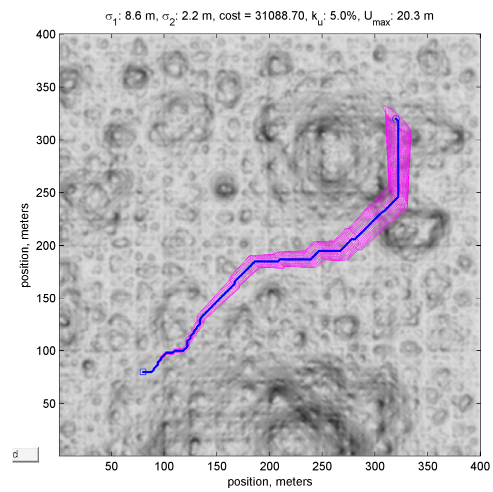

Fig 6. Sample fractal world used for performance simulations.

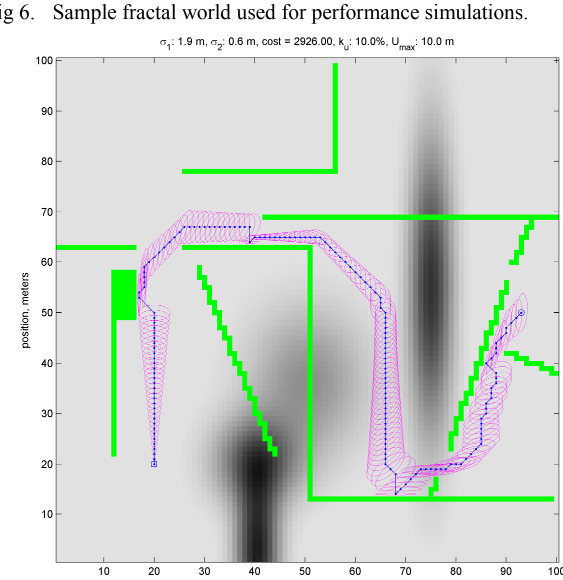

Fig 7. Complex sample world with multiple walls (green obstacles)

Fig 8 shows the path cost of Censi's approach for this world with different values of σ_{TOL} as well as our two approaches. Notice how the entropy-based approach has a small error of about 1% with respect to the lowest cost path. In contrast, when using entropy plus incremental binning, the planner is able to find the lowest cost path. Fig 9 shows the processing time for this example. Notice how entropy is about 8 times faster than entropy plus incremental binning and significantly faster than Censi's approach for most values of σ_{TOL}. Also notice that there are no solutions for Censi's approach when σ_{TOL} is smaller than 0.02. This is because our implementation of this approach is unable to handle more than 1000 bins, and this example requires significantly more than 1000 bin levels for σ_{TOL} smaller than 0.02.

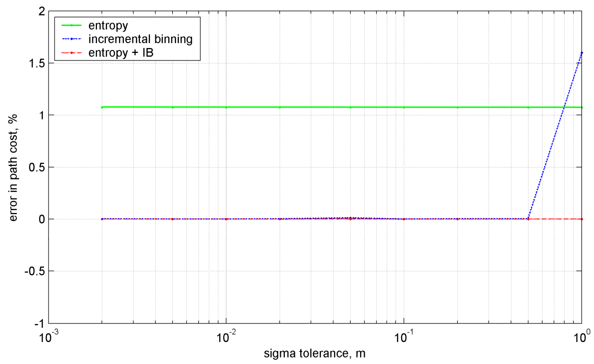

Fig 8. Path cost error of PUPLL vs Censi's approach for varying σ_{TOL} values,

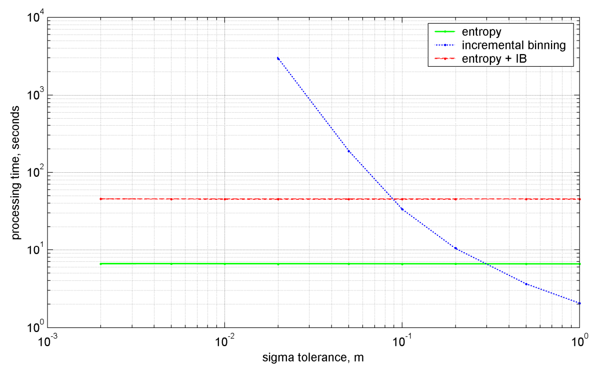

Fig 9. Processing time of PUPLL vs Censi's approach for varying σ_{TOL} values.

III. Conclusions

The approaches presented here build on the original PUP approach and add the ability to localize with linear landmarks, which 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 may not represent the full belief space for the problem makes the planner potentially suboptimal and incomplete. In practice, however, the paths found by the planner usually match the results of Censi's approach (which claims to be optimal within the limitations of the resolution and the representation).

The advantages of using a single parameter to represent uncertainty are multiple. The space and computational requirements of the algorithm are several orders of magnitude lower than what would be required to plan in the complete parameter space. Also, unlike Censi's approach, the space requirements of the algorithm are clearly bounded.

We also propose an approach that combines the best features of the entropy representation and the incremental binning approach. This approach is able to find optimal paths in more complex environments than entropy alone. While this approach is somewhat slower than the entropy approach, it is still faster than incremental binning and also provides clear bounds on the space complexity of the algorithm.

References

[1] R. Alterovitz, A. Lim, K. Goldberg, G. Chirikjian, and A. Okamura, "Steering flexible needles under Markov motion uncertainty". Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, , 1570-1575, 2-6 Aug. 2005.

[2] R. Alterovitz, T. Simeon, and K. Goldberg, "The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty". Proceedings of Robotics: Science and Systems, 2007.

[3] 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

[4] B. Burns and O. Brock, "Sampling-Based Motion Planning With Sensing Uncertainty". Robotics and Automation, 2007 IEEE International Conference on, 3313-3318, 10-14 April 2007.

[5] A. Censi, "Robot Motion Planning with Uncertainty". Universita degli Studi di Roma La Sapienza, 2007.

[6] J.P. Gonzalez, Planning with Uncertainty in Position Using High-Resolution Maps, doctoral dissertation, tech. report CMU-RI-TR-08-02, Robotics Institute, Carnegie Mellon University, May, 2008.

[7] 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.

[8] J.P. Gonzalez and A. Stentz, "Planning with Uncertainty in Position Using High-Resolution Maps," In Proceedings of the IEEE Int. Conference on Robotics & Automation (ICRA) 2007, April, 2007.

[9] N. Melchior, J. Kwak, and R. Simmons, "Particle RRT for Path Planning in Very Rough Terrain," Proceedings of the NASA Science Technology Conference 2007 (NSTC-07), May, 2007.

[10] N. Melchior and R. Simmons, "Particle RRT for Path Planning with Uncertainty," 2007 IEEE International Conference on Robotics and Automation, April, 2007, pp. 1617-1624.

[11] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2006, 1261-1267, May 15-19, 2006.

[12] R. Pepy and A. Lambert, "Safe Path Planning in an Uncertain-Configuration Space using RRT," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2006, 5376-5381, Oct. 2006.

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

[14] C. Urmson and R. Simmons, "Approaches for Heuristically Biasing RRT Growth," IEEE/RSJ IROS 2003, October, 2003.

[15] 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.