Extending the Path-Planning Horizon

Bart Nabbe and Martial Hebert

The International Journal of Robotics Research 2007; 26; 997
DOI: 10.1177/0278364907084100

Extending the Path-Planning Horizon

Bart Nabbe
Tandent Vision Science, Inc.
San Francisco CA 94111

Martial Hebert*
The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213

Abstract

The mobility sensors on a typical mobile robot vehicle have limited range. Therefore a navigation system has no knowledge about the world beyond this sensing horizon. As a result, path planners that rely only on this knowledge to compute paths are unable to anticipate obstacles sufficiently early and have no choice but to resort to an inefficient local obstacle avoidance behavior.

To alleviate this problem, we present an opportunistic navigation and view planning strategy that incorporates look-ahead sensing of possible obstacle configurations. This planning strategy is based on a "what-if" analysis of hypothetical future configurations of the environment. Candidate sensing positions are evaluated based on their ability to observe anticipated obstacles. These sensing positions identified by this forward-simulation framework are used by the planner as intermediate waypoints. The validity of the strategy is supported by results from simulations as well as field experiments with a real robotic platform. These results show that significant reduction in path length can be achieved by using this framework.

KEY WORDS—autonomous vehicles, path planning, perception, forward simulation, inference

1. Introduction

At the core of many autonomous robotics systems is a mobility system that takes data from sensors as input, reconstructs the 3D geometry of the terrain around the vehicle, assesses the drivability of the terrain, detects obstacle regions, and modifies its currently planned path to avoid newly discovered non-drivable areas. The cycle is repeated many times (typically at 10 Hz or better) until the vehicle reaches its goal destination. Typically, such a system is implemented by maintaining a representation of the world in the form of a discrete 2.5D grid which is used for planning. Irrespective of the implementation details of such mobile robot systems, their performance is always severely limited by the fact that the robot can only plan as far as it can sense. Since typical sensors on a mobile robotic ground vehicle have a fairly short range (< 100 m) the planner performance is severely limited, this effect is known as the so-called myopic planning effect (Figure 1).

This limitation occurs because the planner is limited by the maximum range of the mobility sensors. Since typical mobility sensors, such as Laser Radar (LADAR) or passive stereo vision, typically will only acquire data up to a few tens of meters to 100 meters, the planner has no knowledge of what might be encountered beyond the sensed perimeter. As a result, the planner is unable to anticipate obstacles sufficiently early and has no choice but to plan paths close to obstacle boundaries.

For autonomous navigation over long distances, this issue affects the performance of the system by greatly increasing the length of the path traveled by the vehicle. Consequently, the power consumed is increased and, more importantly, the risk of exposure to threats is also increased. In addition, the relative short range of the mobility sensors forces the vehicle to drive closer to terrain obstructions than is safe or necessary.

One solution is to use sensors with longer range to acquire information about the terrain further ahead. We use the term "mid-range" sensors to denote sensors that can acquire range measurements up to a few hundreds meters[1]. However, to use these sensors one needs to take into account constraints about the sensing geometry and acquisition method. The constraints may include increased computation time, which prevents continuous acquisition of data; narrow field of view; constraints on data acquisition procedure (for example, structure from motion (SFM) techniques require that video data be acquired over a sufficiently long traverse in order to establish a long enough

The International Journal of Robotics Research
Vol. 26, No. 10, October 2007, pp. 997–1024
DOI: 10.1177/0278364907084100

Figures 2–4, 6, 8–23, 25, 26, 28 appear in color online: http://ijr.sagepub.com

* Corresponding Author
1. The term "mid-range sensing" is used to distinguish from the type of sensors used for (RSTA) Reconnaissance Surveillance and Target Acquisition that are commonly referred to as long-range sensors. These RSTA sensors be used for target acquisition up to several kilometers.

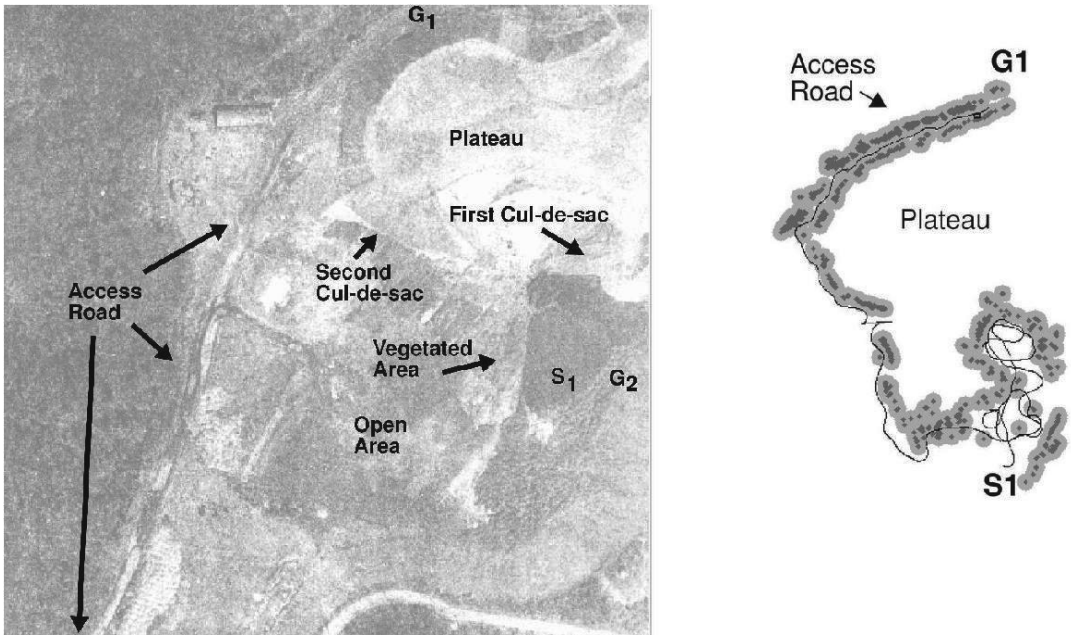


Fig. 1. Typical example of poor performance due to lack of sensor planning and mid-range sensing (Left: Overhead view of terrain; Right: Executed path with detected obstacles shown as shaded regions). The path from S1 to G1 intersects a large hill which is discovered only when the vehicle enters a large cul-de-sac, causing the executed path to be substantially more expensive than the path that would have been followed, had the obstruction been discovered earlier. (From Stentz, 1990)

baseline. All of these constraints imply that it is not practical to sense the environment continuously over a large field of view. Instead, to exploit these sensing resources, we need to do two things: 1) We need to be able to decide *when to look*, so that the longer range data is acquired only at specific times, given the constraints of the acquisition system; and 2) we need to decide *where to look*, so that the data is acquired in the direction that is most informative for planning the next path. Our objective in this paper is to design techniques that combine the current state of knowledge of the robot's environment, the current position of the goal location, and the sensing constraints, to generate a policy to decide when and where to acquire sensor measurements in a way that maximizes the utility of the measurements.

With the problem set in this framework, a solution can be outlined to solve the navigation and view planning problem. Its main component is a *forward simulation* approach which evaluates the benefit of going through a intermediate way-point at which a mid-range sensor measurement is to be taken, instead of going to the goal immediately. Conceptually, many different possible waypoints can be evaluated and a benefit (or utility) measure can be computed for each of them. The utility of a sensor location measures the degree to which taking a measurement at that location decreases the expected risk of planning a path to the goal in the future. The sensing location that yields the largest decrease in expected path risk can

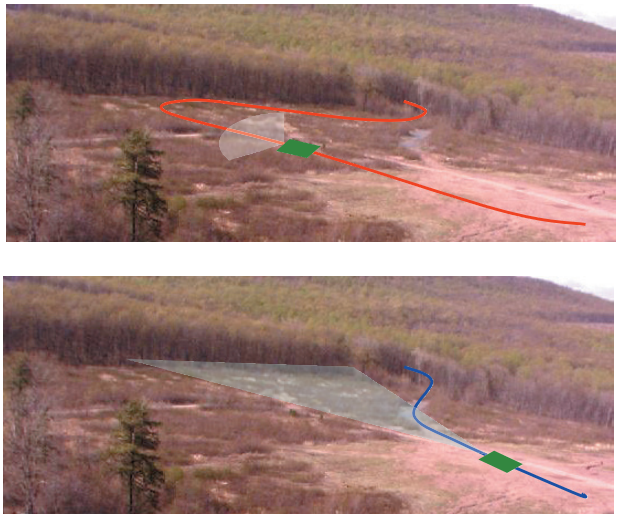


Fig. 2. Top: poor performance due to lack of sensor planning and mid-range sensing, the robot has to rely on mobility sensor data only. Bottom: the pinch point is detected earlier, which allows the planner to adjust the path accordingly.

be selected. This *"what-if"* analysis considers the additional path length incurred as a result of detecting an obstacle late

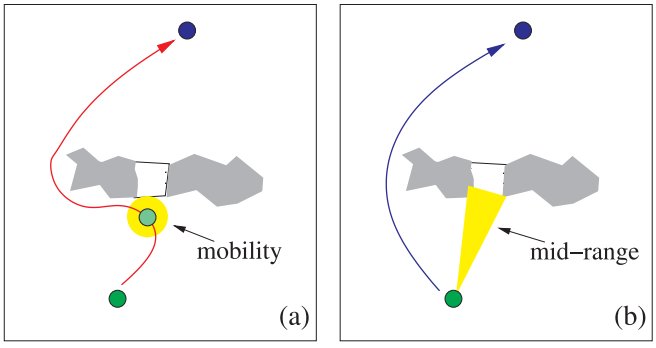


Fig. 3. The difference between the two path lengths in diagram (a) and (b) yields a positive utility from taking an observation. In (a) no mid-range measurement was taken. In (b), a more optimal path was planned due to observing missing data.

rather than early by taking an additional mid-ranging sensor measurement (Figure 3).

The problem with this approach is that it would require the enumeration of all the possible configurations of the world that are consistent with the current data. Since this is clearly intractable, we develop an approximation in which only the *most likely* configuration of the world is used to compute sensing utility. The most likely configuration of obstacles in the world is inferred by combining the currently observed configuration and a prior model. These anticipated obstacles also allow for grouping sparse mid-range sensor data into more meaningful obstacle regions (Figure 4).

2. Related Work

Path planning for mobile robots operating in unstructured environments has received a lot of attention (Daily et al., 1988a,b; Bailey et al., 2000; Goto and Stentz, 1987; Hebert et al., 1997; Laubach and Burdick, 1999; Volpe et al., 1996; Maurette, 2003; Lacroix et al., 2002; Pacisx et al., 2005; Harmon, 1987; Hong et al., 2002a; Rasmussen, 2002; Rosenblum, 2005; Bellutta et al., 2000; Shoemaker and Borenstein, 1998). Many of these systems follow the classical robotic sense plan act cycle. The sensor acquisition system processes the raw data into an obstacle grid that represents the traversability of the environment. A heuristic search algorithm such as A^* (Nilsson, 1969, 1971) would uses this map to plan a path toward the goal. This path is then handed off to the controller (Langer et al., 1994) to be executed. Other, more reactive methods, have been developed for local navigation (Koren and Borenstein, 1991; Ye and Borenstein, 2004). These ideas can be taken further by incorporating detailed vehicle models and forward simulation of paths as in Kelly (1994) a predictive classical control framework. The controller takes the vehicle dynamics and the current vehicle attitude into account for predicting what is safe to navigate in the immediate future. Lacaze et al. (1998) extend this idea to include a further planning lookahead step, in which a graph of all possible vehicle trajectories is evaluated based on vehicle dynamics.

Unstructured outdoor environments are much larger than typical indoor setups and require therefore methods that efficiently deal with the increased problem space. Methods for efficient planning in these large environments have explored several different techniques. One approach is to represent the grid at multiple resolutions and to use a coarse-to-fine approach to efficiently compute a path (Pai and Reissell, 1998; Devy et al., 1995). Another approach to make planning more efficient is to reduce the amount of computation. Stentz (1990, 1994) reduces the computation time by reusing previous planning results. Further improvements that remove the problems associated by planning on a discrete grid yielded Field-D* (Ferguson and Stentz, 2005a,b). Field-D* interpolates between the discrete grid coordinates without compromising the completeness of the standard algorithm. Another approach is to reduce the search space, For example, Spero and Jarvis (2002) uses rapidly-exploring random trees to do so.

Most related to our work is the problem of "View Planning" in which we need to decide where to acquire the next sensor reading given certain constraints. Part of the work in this area focuses on 2D worlds with polygonal structures (Chen and Huang, 1994; Kleinberg, 1994). There is also extensive work on view planning in the area of map building. For example, the work of Moorehead (2001), González-Baños and Latombe (1998) and Zelinsky et al. (1993) with respect to sensing for exploration, focuses on complete exploration of the environment. The robot is driven by a utility metric that represents some notion of information gain for exploring certain areas. It does not deal with view planning in the context of deploying a sensing device, but merely deals with visiting locations and ensuring that the entire environment has been observed. An objective function based solely on information gain is not appropriate for active sensing for planning since it does not account for additional costs for sensor deployment. Kruse et al. (1996) describe a more useful function for active sensor deployment. A "planning–sensing–updating" cycle is described, which uses the current and incomplete world model with the motion constraints to compute the next-best view such that an objective function in configuration space is maximized. More insight can be gained from the "active vision" community. For example Kutulakos (Kutulakos and Dyer, 1992; Kutulakos et al., 1994) a formalization of vision-guided exploration, including a method for exploration of arbitrary surfaces in 3D. Other exploration approaches (Banta et al., 1995) propose methods that can plan for a next view very quickly, but without guarantee on optimality, or in more constrained scenarios (Maver and Bajcsy, 1993; Pito, 1996).

Different from the view planning methods are the sensor based path planning methods. They are different in that views are not planned for covering a certain region, but vehicle mo-

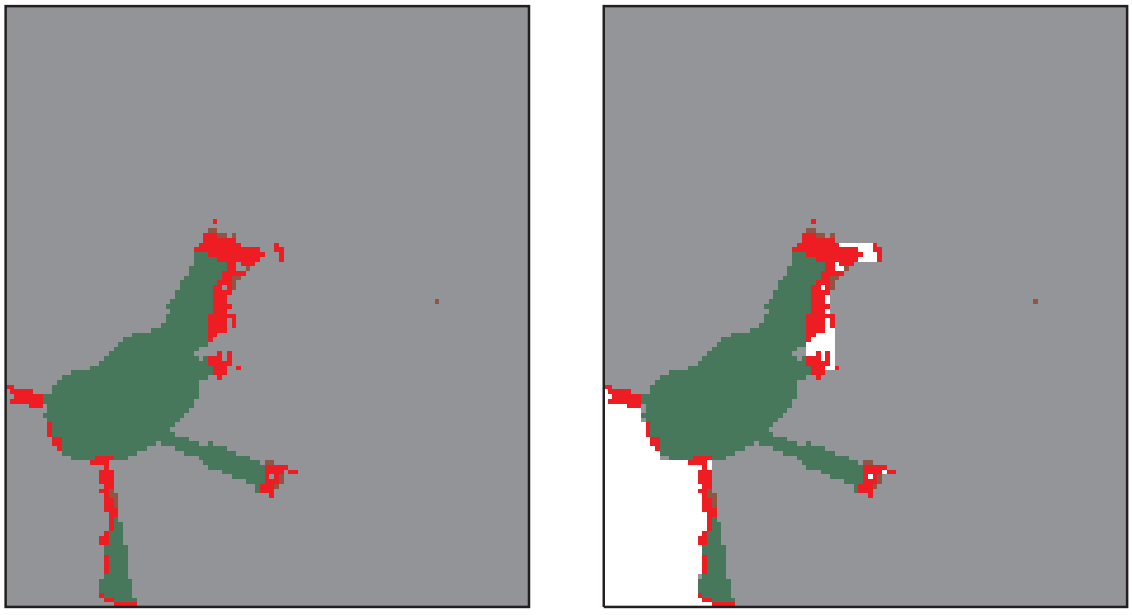


Fig. 4. The inferred world as shown on the right has the inferred obstacles marked in white, the inferred empty space, observed traversable and observed obstacles are marked in increasingly darker shades. The sensed world is shown on the left for reference. The inferred part captures the most likely configuration of the world based on the current observations.

tion is planned with respect to sensor constraints and/or coverage of the area to traverse. In other words, the motion or path of the robot is planned simultaneously with the sensing action. For example the algorithm described in Choset and Burdick (1995) and Choset (1996) continuously senses around the robot's perimeter and constructs a topological map while it navigates toward the goal. While the algorithm in Choset and Burdick (1995) and Choset (1996) continuously senses and plans a path, the sensing itself does not involve any particular view planning. If we would reason about the environment that we have observed so far, we can actively take measurements so that we can compute better paths. In González-Baños and Latombe (2000), González-Baños and Latombe (2002), and Grabowski et al. (2003) such an active sensor based planning method is presented. The algorithm is based on detecting regions that are safe to navigate. These regions are extended at every iteration according to newly sensed data. Candidate view points are generated at the boundaries of these safe regions, the viewpoint that exposes the largest fraction of the unknown boundary is selected for the next viewing position. A drawback of this algorithm is that it assumes that the range data does not contain any errors. Noisy data can cause mis-registration of range scans which may cause the result of the algorithm to become invalid. The sensing is also solely based on coverage and cannot be tailored for navigation purposes. The approach we have taken to overcome these issues is to use a confidence label on each data item and evaluate paths only with respect to the goal destination.

An alternative idea is explored in the "Bug" family of algorithms (Lumelsky and Stepanov, 1987; Rao et al., 1993). These algorithms assume a simple contact sensor and simple rules to navigate through unknown environments. Kamon et al. (1996) extend this technique to incorporate the usage of range sensors by using the reduced visibility graph, better known as the tangent graph. The local tangent graph, which is the visible subset of the tangent graph to the robot, can be constructed by using the measurements of a laser range finder. Instead of driving up to an obstacle, as in the bug algorithm the TangentBug algorithm, senses the perimeter and follows the local tangent graph. Laubach (Laubach, 1999; Laubach and Burdick, 1999) extends the TangentBug algorithm even further into the WedgeBug (RoverBug) algorithm which explicitly deals with a sensor model. She adds some virtual states to the typical bug states (Motion toward goal and Boundary following); in these states the WedgeBug algorithm senses more from its environment to determine the local tangent graph which it uses to generate new path segments from. This algorithm is very elegant and provably correct, however it still suffers from the same problems as the safe region algorithm, in that there is no notion of uncertainty in sensor data and it needs continuously detectable obstacle boundaries.

Most similar to our work is the work by Gancet and Lacroix (2003). They present in their paper the Perception-guided path planning (PG2P) approach which is a hybrid sensing and path planning method. Some of the same philosophy that we have embraced have been incorporated in the PG2P algorithm as

well, although there are some notable differences between the two approaches. Most importantly, the PG2P algorithm focuses on their definition of Confidence of Perception or COP. Their COP depends only on prior knowledge about the sensor model. For the type of range sensors they consider, the COP can be stated as a function of the perception distance. The COP is high for a location in close proximity and low for a sensed cell that is further away. In addition, the COP can only be used if the cell to be viewed has an (obstacle/traversable) label and corresponding probability distribution already. In contrast with the PGP algorithm, in which the influence to the path from a given (obstacle/traversable) label is considered, our approach reasons about the effect on the path that unseen parts of the terrain have. Using the COP function together with the confidence of the label, a traversability cost is defined (Gancet and Lacroix, 2003). The result is that the terrain's traversability is reinforced by the confidence in this measurement. Similar to what we propose is the reasoning for taking measurements based on a metric. Taking a measurement and therefore reducing the confidence level will have an influence in the new traversability cost. They also define a utility for sensing that takes into account how much the traverse cost changes as these simulated measurements are taken. The robot will then take a sensor measurement when this utility is positive. Alternatively they select nodes on the boundary of the already observed area (so they can be reached safely) as locations for observations. However, this heuristic will not alleviate the problem of the myopic sensing horizon, since when we get to this boundary sensing location, we might discover too late that we encounter an obstacle.

Another important component of the problem is to model the uncertainty in sensing and planning (in our case, we are primarily concerned with the map uncertainty). Ways to model this uncertainty is in a player game-theoretic framework have been proposed (Esposito and Kumar, 2000; LaValle, 1995; LaValle and Sharma, 1997). A different approach that is based on an AND–OR graph search method was presented by Ferguson and Stentz (2004) and Ferguson et al. (2004). This method relies on imperfect prior map data to identify decision points for which the traversability is not known. These decision points, or pinch-points, are used as decision nodes to build the AND–OR graph that is used to plan the expected best path.

3. Algorithm Description

Our overall approach is summarized in Figure 5. The difference from a traditional Sense–Plan–Act cycle (Figure 5, left-hand side) is the computation of intermediate waypoints. At these waypoints extra terrain observations are acquired with a mid-range sensor and a new waypoint along the path to the goal is computed (Figure 5, right-hand side). These waypoints are sensing locations that are computed to maximize the utility of the sensor measurements. The approach has two main ingredients. The first one is a way to evaluate the *utility of taking a sensor measurement* from a position in the world, given a current, partial model of the terrain accumulated from prior sensor readings. The utility is computed by simulating the paths that the vehicle might follow in the future (forward simulation). To be able to compute the utility, we need to *predict* likely future configurations of the world, given our current knowledge. This is the second ingredient: The ability to predict, or *hallucinate future configurations of the world based on the current model*.

To determine the next observation location (waypoint) and observation direction, alternative future scenarios are evaluated by running a forward simulation. The current state is extrapolated into the future and the expected path length is used to determine if the tentative observation location contributed to a path-length reduction.

3.1. Definitions and Notations

We assume that the terrain is represented by a discrete grid of cells $\mathcal{S} = \{i = 1, \ldots, M\}$. We denote by D the observed data from the terrain, where $D = \{d_i\}_{i\epsilon\mathcal{S}}$ and d_i is the data at cell i. In general, d_i could be the set of values describing all of the raw data collected at the corresponding cell. Since such a general representation would be difficult to handle, we use a simpler representation in which we summarize the entire data at i into a single number that relates directly to the local shape, and therefore the local drivability, of the terrain. Specifically, we define d_i as the terrain gradient estimated at i from sensor data. While the local gradient was chosen primarily because it was easy to implement for the experiments shown later, more general drivability costs can be used for d_i (Sofman et al., 2006) without significant changes to the basic formulation of the approach. In general, only a small fraction of the environment has been sensed. Therefore, d_i has a valid value only at those cells for which sensor data has been acquired. There is no requirement that terrain slope be available everywhere. In fact, in most experiments, we start with an empty map in which no cell contains valid data.

We encode the distribution of obstacles in the map by associating a label l_i to each cell i, with $l_i\epsilon\{-1, 1\}$. In our case, l_i indicates the presence or absence of an obstacle at location i (Figure 6). We denote the entire set of labels by $L = \{l_i\}_{i\epsilon\mathcal{S}}$.

The mid-range sensor model is parameterized as follows (Figure 7). From a given orientation and sensing location, the sensor acquires data within a cone characterizing the field of view and the maximum range. We denote by θ the orientation of the sensor. For implementation reasons, it is sometimes convenient to (over-)parameterize the sensor orientation by specifying the location (x, y) of a point of interest toward which the sensor is aimed. We assume that measurements are generated for the terrain cells covered by the field of view cone, given a sensor orientation and position. This is is, of course, an

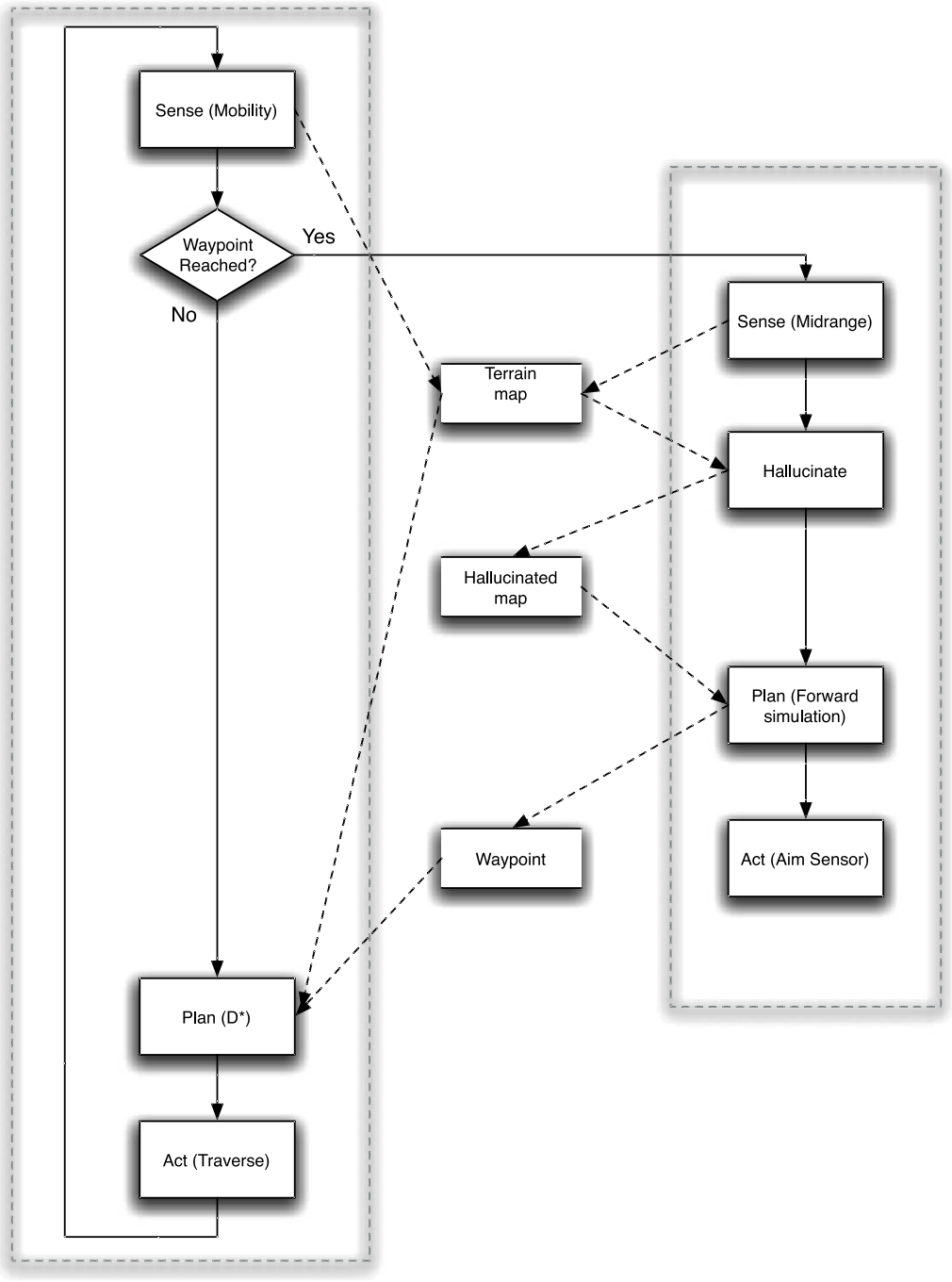


Fig. 5. Mid-range planning sensing algorithm overview. The leftmost diagram is the low level navigation loop. The diagram on the right shows the flow of the algorithm that does the forward simulation and sensor planning. Solid lines denote control flow of the algorithm, while dashed lines denote data transfer.

optimistic assumption but it is consistent with the sensing utility defined later in this section. Finally, we assume a nominal cost of α associated with a single mid-range sensing operation. This constant cost aggregates the computational cost, as well as any other overhead involved in taking a measurement.

We assume that the mobility sensor acquires data in a circular region around the vehicle. We assume that, for a given vehicle location, the terrain slope is estimated at all the terrain cells within the field of view. Obstacle labels are computed from the estimated slope at each cell, taking into account also

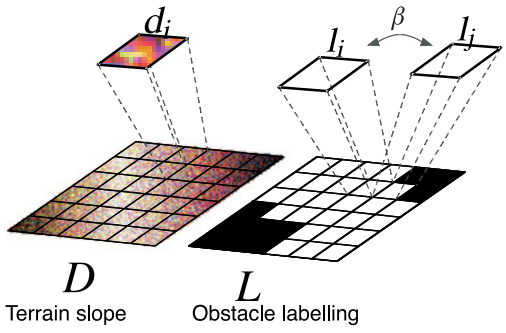


Fig. 6. Depiction of the terrain model parameters. d_i denote the terrain gradient at cell i and l_i denotes the corresponding label.

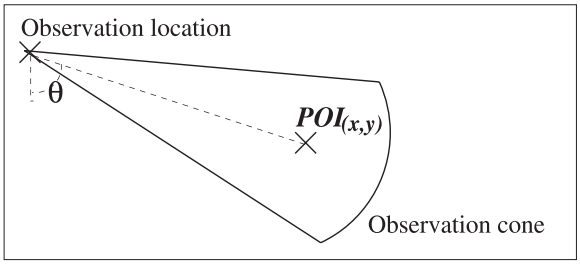


Fig. 7. Mid-range sensor geometry.

the distance to the sensor, so that cells closer to the sensor detect obstacles with higher confidence. Here again, the exact type of features used for obstacle detection is not essential, and more sophisticated terrain characterization algorithms can be substituted. The assumption of a circular field of view is not applicable for many mobile robots, but it not essential to the approach. It is used only for efficiency reasons to simplify the forward simulation portion of the algorithm, but the approach would apply as well to a forward-looking sensor as described in the experiments section.

We use D* as the baseline path planner in this approach. At each cycle, D* generates a path to the goal based on the current configuration of the terrain map. In addition, D* is also used for generating forward simulated paths to evaluate the sensing utility values. We used D* because it fully supports dynamic updates of the terrain map, so that paths can be quickly replanned based on new sensor data. However, any other planner with that capability could be used as well. The only requirement is that it computes efficiently the optimal path given a particular terrain configuration.

We denote by $C_{ns}(L, \mathbf{x}, \mathbf{v}_G)$ the cost of the optimal path, assuming the distribution L of obstacle labels, from a current location $\mathbf{x}$ to a goal location $\mathbf{v}_G$, *without* taking any additional mid-range measurements. The path cost is defined as the length of the path. Similarly, we denote by $C_s(L, \mathbf{x}, \mathbf{u}, \theta, \mathbf{v}_G)$

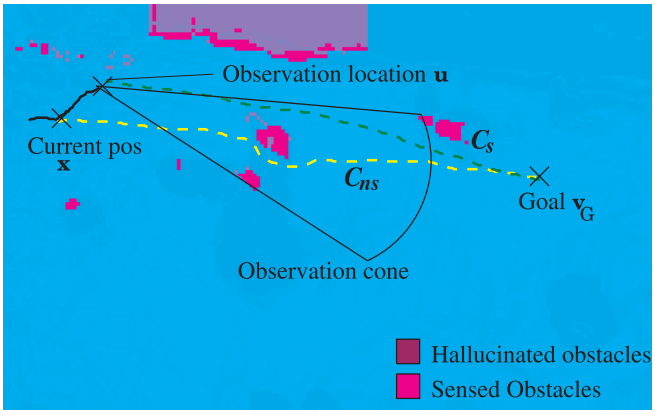


Fig. 8. Illustration of the path computed with (C_s) and without sensing (C_{ns}) in an example world configuration.

the cost of the optimal path from $\mathbf{x}$ to $\mathbf{v}_G$ but, this time, assuming that we take an intermediate mid-range measurement from position $\mathbf{u}$ and orientation θ (Figure 8).

3.2. Evaluating Sensing Utility

The first component to our solution is to evaluate the utility of taking a mid-range sensor measurement from a particular waypoint location. We define the utility of a sensing waypoint as the expected gain (possibly negative) in path length that is achieved by moving to that waypoint from the current position and taking a mid-range measurement. Let us assume that the robot is at position $\mathbf{x}$ and that it has accumulated sensor readings over part of the environment, so that it has a current partial (possibly empty) knowledge of the terrain. Let us also assume for a moment that we know the *actual* configuration of obstacles in the map, L, in addition to the current map from sensor readings. If the robot stops at an intermediate waypoint $\mathbf{u}$ to take a mid-range sensor reading at orientation θ, it would get additional information, presumably leading to a more efficient path. Since we assume that we know the true underlying distribution of obstacles, we can compare the two paths costs, and we define the utility of the taking a measurement at $(\mathbf{u}, \theta)$ as being the relative gain, possibly negative, computed as

$$\frac{(C_{ns}(L, \mathbf{x}, \mathbf{v}_G) - C_s(L, \mathbf{x}, \mathbf{u}, \theta, \mathbf{v}_G) - \alpha)}{C_{ns}(L, \mathbf{x}, \mathbf{v}_G)}. \quad (1)$$

C_{ns} and C_s are computed by forward simulation of the robot through the map in configuration L, using the mobility sensors (Figure 8). In principle, we should choose the sensing location $(\mathbf{u}, \theta)$ which maximizes this gain since it is the one that would lead to a more efficient path. Of course, we do not know the actual terrain configuration L (if we did, we would not need any sensing!) so the ratio above cannot

be computed directly in practice. Instead, we need to replace it with an *average* computed over all possible configurations of the worlds that are compatible with the current, partially known obstacle map. More precisely, assuming that there are N possible distributions of obstacles in the map, we denote by $L_j, j = 1, \ldots, N$ the jth labeling. Given the data observed so far, not all configurations are equally likely, so we associate a probability $P(L_j \mid D)$ with each distribution of labels. $P(L_j \mid D)$ measures the likelihood that the true configuration of the world is L_j, given the data observed so far D. Assuming that this probability distribution can be computed, we can estimate the expected gain in path cost by averaging over all possible configurations L_j:

$$\left(\sum_j \frac{P(L_j|D)\big(C_{ns}(L_j,\mathbf{x},\mathbf{v}_G) - C_s(L_j,\mathbf{x},\mathbf{u},\theta,\mathbf{v}_G) - \alpha\big)}{C_{ns}(L_j,\mathbf{x},\mathbf{v}_G)} \right). \quad (2)$$

Finally, to compute the utility of taking a measurement at an intermediate waypoint $\mathbf{u}$, we eliminate the sensor orientation by picking the one that yields the highest expected gain:

$$\psi_{\mathbf{u}}(\mathbf{x}, \mathbf{v}_G) = \max_\theta \left(\sum_j \frac{P(L_j|D)\big(C_{ns}(L_j,\mathbf{x},\mathbf{v}_G) - C_s(L_j,\mathbf{x},\mathbf{u},\theta,\mathbf{v}_G) - \alpha\big)}{C_{ns}(L_j,\mathbf{x},\mathbf{v}_G)} \right). \quad (3)$$

A positive utility means that the expected path cost to the goal is lower if we take an additional sensor measurement at $\mathbf{u}$ than if we would go to the goal directly and possibly discover a blocked route later.

In principle, $\psi_{\mathbf{u}}$ can be computed at every $\mathbf{u}$ by simulating the two paths of cost C_{ns} and C_s and the resulting utility map (Figure 9) can be used by the planner to select the most favorable sensing position. This "forward simulation" (Kelly and Stentz, 1998; Esposito and Kumar, 2000; LaValle, 1995) is at the heart of our approach. A location in the map has positive utility if the expected traverse from the current location to the goal through that location is cheaper than the expected cost for going to the goal directly (Figure 10). This approach of incorporating utility as a cost is similar to the approach that Rosenblatt (Rosenblatt, 1997, 2000) takes in his arbiter. However, his utility map only looks at a few possible actions over a limited horizon and does not deal with additional utility for sensing purposes.

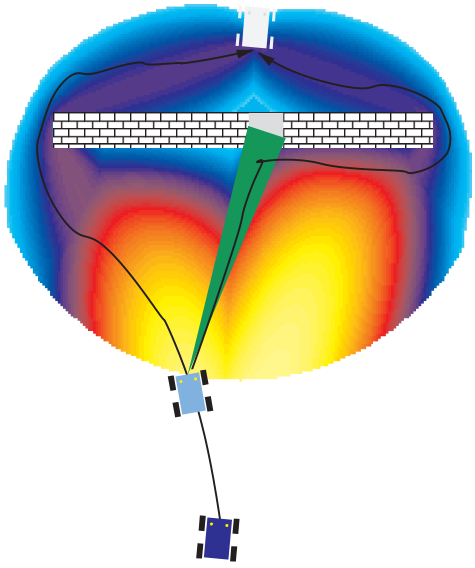


Fig. 9. The utility map displayed for a very simple partially known world. The robot depicted at the bottom of the figure denotes the current robot position. A high measure of interest is represented with a bright value. The brightest point (highest utility) is used as the next sensing location.

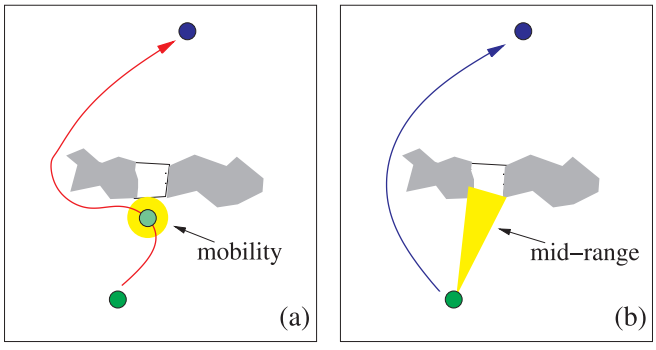


Fig. 10. (a) The obstacle is detected late, resulting in an expensive detour. In (b) a mid-range sensor measurement supplies the missing information to plan a more optimal path. The difference between the two path lengths yields a positive utility for the observation location.

3.3. Hallucinating Worlds

Both C_s and C_{ns} can be evaluated by running the planner on different "virtual" configurations of the world map corresponding to different configurations L_j. In reality, however, this would require the enumeration of *all* possible configurations of the world, which is clearly a combinatorially large set. Furthermore, the sum above must be evaluated, in principle, not only for all cell locations, but also for all possible sensor orientations. Therefore, the computation of the optimal utility function defined above is not tractable and an approximation must be used to reduce the computation while retaining near-optimal evaluation of the utility.

More precisely, we need a mechanism to predict which parts of the environment are not traversable so that we can evaluate the sensing utility by summing over the most likely

configurations of the world, while discarding the unlikely configurations. In practice, only a tiny fraction of the combinatorially large number of configurations of the world is likely to occur, thus permitting the use of drastic approximations in the evaluation of the utility. This problem can be expressed as an inference problem: given what we have observed so far and a prior model, we can infer, or *hallucinate*, the underlying world map and hence the traversability. With this hallucinated world we can perform forward simulation and thus compute the paths that are needed to compute the utility for taking additional mid-range sensing measurements.

In order to be able to hallucinate the world as it will appear in the future, we could use a previously learned distribution of worlds and use an inference mechanism to draw most likely world configurations from it. Since we are only interested in knowing if a particular cell is traversable, a binary representation of the world will be sufficient. A hallucinated traversability label for a particular map cell needs to adhere to the underlying spatial distribution of obstacle vs. non-obstacle. This problem of grouping features together has been studied extensively in many fields, for example computer vision (Besag, 1986; Won and Derin, 1992; Li, 2001).

In order to compute the utility $\psi_{\mathbf{u}}$ we use a common approximation which assumes that $P(L \mid D)$ can be approximated by a delta function (Kumar and Hebert, 2006). This amounts to replacing the average configuration of the world with the most probable configuration or maximum a posteriori (MAP) estimate:

$$P(L \mid D) \approx \delta(L, \hat{L}), \tag{4}$$

with

$$\hat{L} = \text{argmax}_L P(L \mid D). \tag{5}$$

Intuitively, $\hat{L}$ is the most likely world configuration inferred from the observed data, and the utility function is then approximated by

$$\psi_{\mathbf{u}}(\mathbf{x}, \mathbf{v}_G) = \max_\theta \left(\frac{\left(C_{ns}(\hat{L},\mathbf{x},\mathbf{v}_G) - C_s(\hat{L},\mathbf{x},\mathbf{u},\theta,\mathbf{v}_G) - \alpha\right)}{C_{ns}(\hat{L},\mathbf{x},\mathbf{v}_G)} \right). \tag{6}$$

The next problem we need to address is how to compute $\hat{L}$ from the distribution $P(L \mid D)$. We start with the observation that the labels on cells expressing traversability are not independent. If a cell is classified as non-traversable, then there is a high probability that the neighboring cells are also non-traversable except at the discontinuities. This is also true for the traversable cells. This kind of contextual dependency in the labels on 2D lattices has been well studied and is often referred to as spatial smoothness.

A standard approach for representing this spatial smoothness is to use Markov random fields (MRFs), which can incorporate local contextual constraints in labeling problems in a principled manner (Li, 2001). These MRFs provide an efficient method to group features based on their labeling while adhering to an underlying spatial model. This inference technique has been successfully applied in the field of computer vision (Besag, 1986; Li, 2001; Won and Derin, 1992) to infer the original image from noisy data. Because of the similarity of the problems addressed in the computer vision community with our problem, inferring an obstacle labeling for a partially known terrain led us to formulating our inference problem as an MRF inference.

MRFs are generally used in a probabilistic generative framework that models the joint probability of the observed data and the corresponding labels. In that framework, the posterior over the labels given the data is expressed using Bayes rule as

$$P(L \mid D) \propto p(L, D) = p(D \mid L)P(L). \tag{7}$$

The first term in the product, $p(D \mid L)$, is the likelihood of the data. In general, it is assumed that the data at each cell is conditionally independent given the labels at that cell, i.e. $p(D \mid L) = \prod_{i \epsilon S} p(d_i \mid l_i)$ (Besag, 1986; Li, 2001). We model the likelihood for each class, traversable/non-traversable, as a Gaussian distribution. From fitting experiments with natural terrain data a Gaussian model appears to capture the distribution adequately. Different models may be necessary if other costs are used for d_i instead of the terrain slope:

$$P(d_i \mid l_i) = \frac{1}{\sqrt{2\pi\sigma_{l_i}^2}} \exp\left\{ \frac{1}{2\sigma_{l_i}^2} (d_i - \mu_{l_i})^2 \right\} \tag{8}$$

The parameters of the two Gaussians representing the likelihood for these classes can be easily estimated from labeled training data. This estimation takes place by drawing examples of the typical type of environment in which the robot is to navigate. Thereafter the robot will use these estimated parameters for online classification/hallucination of terrain. For the current model, this estimation boils down to fitting a single Gaussian to each of the two classes.

After modeling the likelihood of the data, we need to model the prior over label configurations $P(L)$, which encodes the notion of spatial smoothness of the terrain labels. Since we have only two classes (traversable/non-traversable), this can be expressed in a MRF formulation as a binary classification problem, for which commonly the Ising model is used to represent local spatial dependencies (Kumar and Hebert, 2006; McCoy and Wu, 1973). This model approximates $\log P(L)$ by a sum of terms of the form $\beta l_i l_j$, where i and j are neighboring cells in the map, l_i and l_j are their labels, and $\beta > 0$ is a constant. If two neighboring cells have the same labels, e.g., they are both obstacle cells, the corresponding term will increase the overall probability. Conversely, neighboring cells with different labels will decrease the overall probability. Intuitively, this model favors terrain configurations with large homoge-

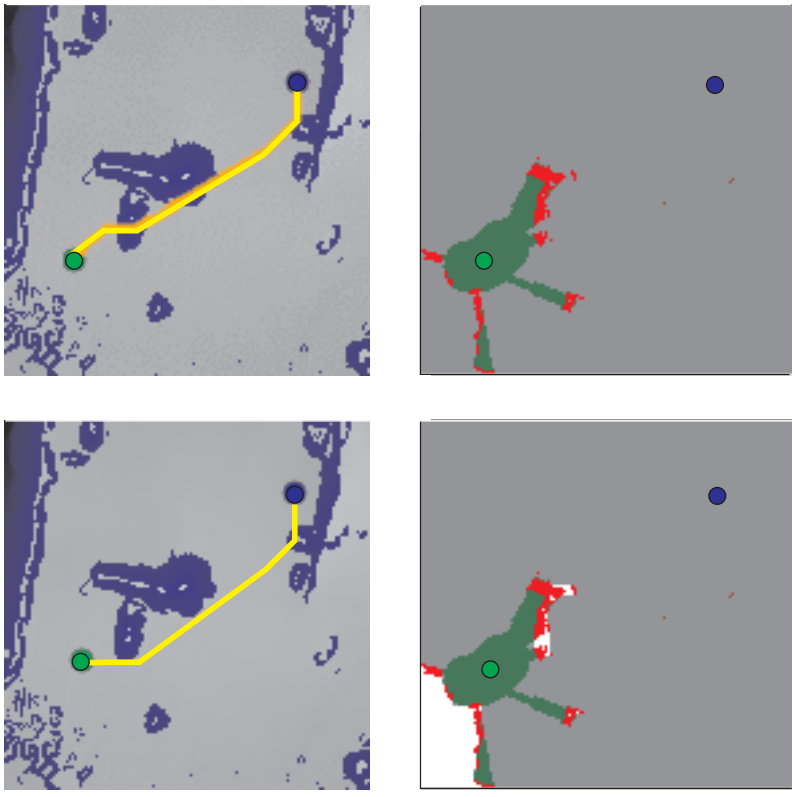


Fig. 11. Typical example that shows the influence of the obstacle hallucination. The robot (lower left circle) travels to the goal (upper right circle). The path is plotted in between. The hallucinated obstacles are in white, hallucinated empty space in gray, observed traversable and observed obstacles are in respectively darker shades. In the top panel, the planner uses only the information from the sensors and it plans a path through the obstacle. In the bottom panel, the hallucination algorithm merges obstacles correctly together and produces a better path.

neous regions over configurations with randomly scattered obstacle cells. This is consistent with the real-world situations in which obstacles are solid objects or extended terrain surfaces, not isolated points on the terrain. The parameter $\beta > 0$ controls how homogeneous the labeling should be. If β is close to 0, all the configurations are equally likely. For very large values of β, only those worlds with near constant labels (everything is an obstacle, or everything is traversable) do survive with significant, non-zero probability. This model is isotropic in that there is no preferred shape or direction for the obstacle regions. Non-isotropic models could be used also, but they require additional training and more parameters.

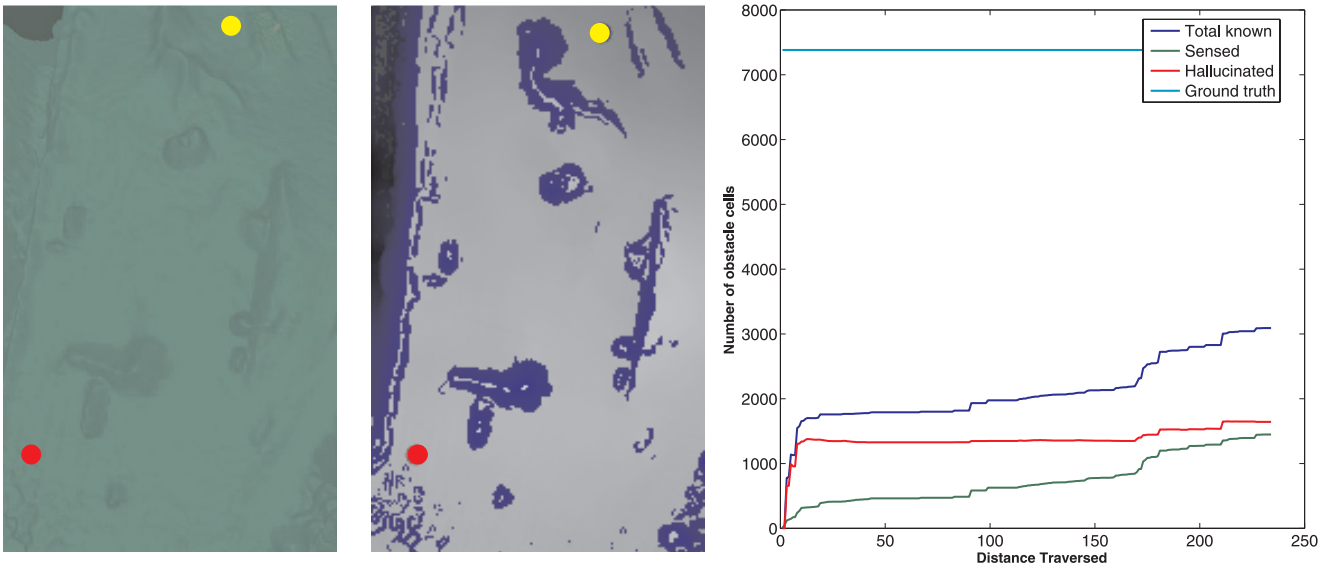
Combining the likelihood model $P(D \mid L)$ with the prior over labels $P(L)$, we can write the overall posterior over labels as follows:

$$P(L \mid D) = \frac{1}{Z} \exp\left(\sum_{i \epsilon \mathcal{S}} \log p(d_i \mid l_i) + \sum_{i \epsilon \mathcal{S}} \sum_{j \epsilon \mathcal{N}_i} \beta l_i l_j \right), \quad (9)$$

in which Z is the normalizing constant, the partition function, and $\mathcal{N}_i$ is the set of cells in the neighborhood of cell i. For this MRF model, a standard result is that computing the Maximum A Posteriori (MAP) configuration of the labels, $\hat{L}$, given the data can be solved exactly using graph min-cuts/max-flow algorithms if $\beta > 0$ (Greig et al., 1989; Besag, 1986). With such a min-cuts algorithm (Kumar and Hebert, 2006), we now have a way of computing $\hat{L} = \text{argmax}_L P(L \mid D)$.

The min-cuts/max-flow algorithm that solves the MRF equation (9) produces a labeling $\hat{L}$ of traversable/non-traversable cells that is most consistent with the current data and with the smoothness model captured by the Ising model. This binary labeling of traversable and non-traversable cells is updated for a given instance of the currently observed terrain data and represents the most anticipated configuration of the world (Figure 11). This configuration $\hat{L}$ is then used to compute the utility $\psi_{\mathbf{u}}(\mathbf{x}, \mathbf{v}_G)$ in Equation (6). The best waypoint $\mathbf{u}$ can then be selected by choosing the location with the maximum utility.

It is interesting to examine how the influence of the inference behaves over a typical traverse of the robot. To illustrate



(a) (b) (c)

Fig. 12. Example demonstrating the influence of the obstacle hallucination over thecourse of a few waypoints. (a) is an overhead view of the terrain, with start and goal location marked. The ground truth obstacle labeling is displayed in (b). The cumulative number of obstacle cells is plotted in (c). The graphs confirm the intuition that while more of the environment gets explored, less needs to be inferred. In the limit, the number of additional cells inferred at each step will go to zero and the total number of known obstacle cells will equal the number for the ground truth labeling.

the operation of the obstacle inference algorithm, Figure 12 shows an example experiment. In Figure 12(a) and (b), the example terrain is shown with its ground truth obstacle labeling. The planner simulated a robot traversal between the circular markers.

The number of map cells that are detected as obstacles at every position along the path is shown in Figure 12(c), along with the number of hallucinated cells. The ground truth, that is, the total number of actual obstacle cells computed directly from the underlying elevation map of Figure 12(a) is also included. This number provides a baseline for reference: It is the total number of obstacle cells that would be included in the map if the entire world had been explored and sensed.

When we look at the number of cells hallucinated in Figure 12(c) over the course of a traverse, we see that the number of inferred cells first increases rapidly and then tapers off when we have sensed more of the environment. This is because initially there is no data available, therefore the hallucination algorithm has no evidence for predicting any possible obstacles. It will therefore not infer any obstacles. As some sparse terrain data has been sensed, there is enough evidence for some cells to be inferred and classified as obstacles. However, as even more of the environment becomes known and inferred obstacles get confirmed, the number of inferred obstacles added at each step increases more slowly or even vanishes. This happens around distance 20 in the example of Figure 12. This is true until new data becomes available from an area completely unobserved previously, this will become a new seed spawning off more inferred observations so that the cycle repeats. This happens around distance 160 in the example of Figure 12. In the limit, when all cells have been explored there will be no inferred obstacles, the experiment does not show this final configuration.

It is important to note that we use here a simple model for the interaction between cells. This model may not be sufficient for representing more complex relations. For example, if we expect the structures that we encounter in a particular environment to be (on average) elongated in a particular direction, we would need to use a non-isotropic model. In general, interaction models learned from the training data should be used. Nevertheless, the basic MRF model used here is sufficient to validate our basic approach based on inferring likely terrain configuration by combining the currently known terrain data and a prior model. The development of richer prior models is left for future work.

3.4. Overall Algorithm Operation

In the approach presented so far, we need to search over all possible values of θ, which is potentially a very expensive operation. Intuitively, only a small subset of θ values are worth

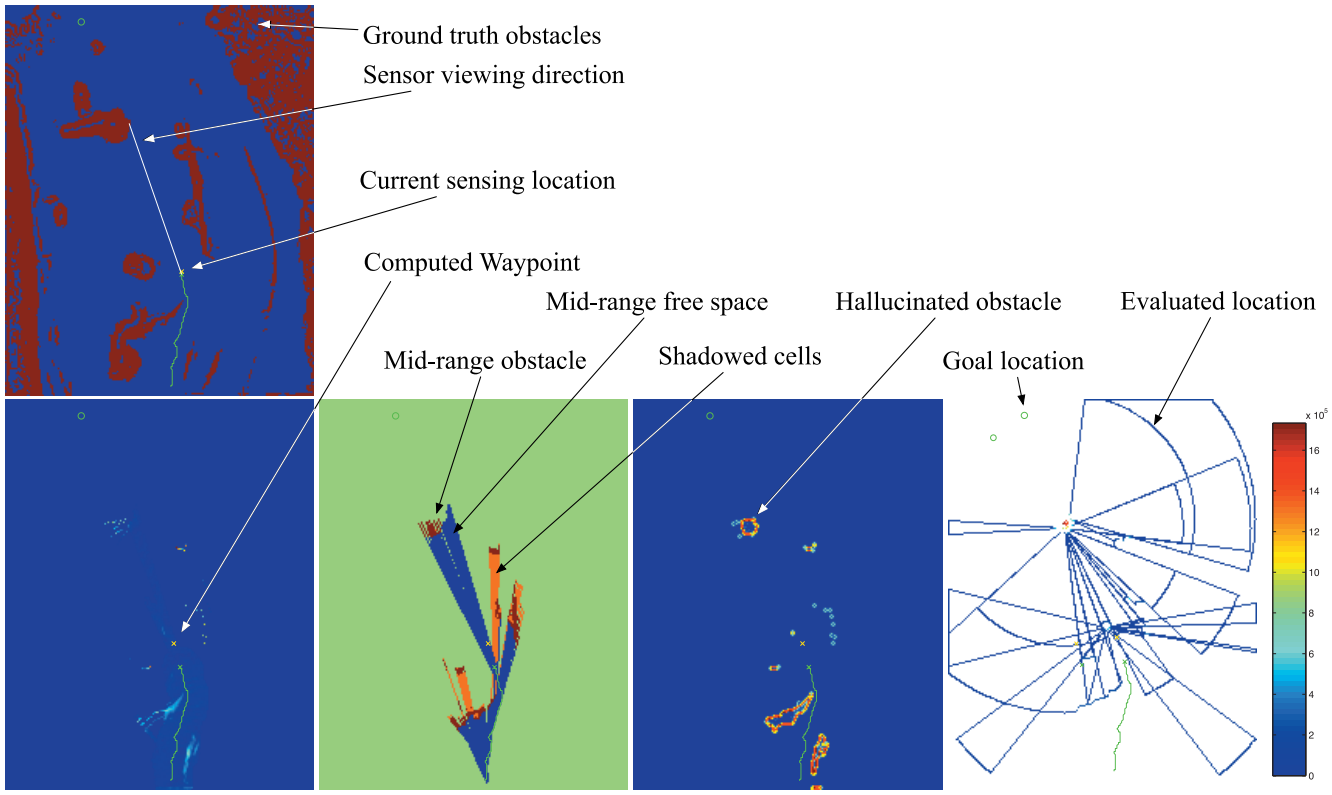


Fig. 13. The diagrams that are used in the example presented in Figure 14 use the notation that is detailed here. Counter-clockwise from the top left is depicted: terrain map (ground truth), observed terrain map, mid-range pencils of rays, binary hallucinated obstacle map and the utility map.

considering. For example, it is not useful to take a measurement that exposes those parts of the world that are very unlikely to be traversed, given the current path and world model.

We can use the MAP estimate $\hat{L}$ to reduce the search space by restricting ourselves to evaluating only those sensor orientations that point to cells in the MAP estimate that differ from the current known labeling. These areas obtained by differencing the $\hat{L}$ from the current map provide a small number of candidate orientations, thus preventing exhaustive search through all possible values of θ. This is a reasonable strategy because cells for which the labeling remains unchanged after inference have no influence on the path and therefore do not need to be observed again.

The overall operation of the complete algorithm shown in Figure 5 is illustrated in Figure 13 and Figure 14. First, some initial range measurements are acquired to bootstrap the forward simulation. Then the inference procedure produces the anticipated obstacles, which are used to evaluate the alternative paths in the forward simulation and select the best intermediate observation location. It could happen that no useful mid-range is found for a long time. This would happen, for example, if the initial map is empty and no new obstacles are added for a long time. To avoid this situation, we force the system to take a mid-range measurement after a fixed timeout interval, even if the forward simulation does not expect that there is anything to sense.

4. Evaluation

Several experiments were carried out to validate the proposed algorithm. These include results from field tests with an autonomous vehicle, as well as results from a much larger set of simulated experiments.

4.1. Experiment with Wide-Baseline Stereo

In this example, we used a Pioneer DX robot to navigate across a parking lot to reach a goal behind a building. The Pioneer is equipped with a single camera (the second image was captured at a fixed three meter baseline to collect wide-base line stereo images). A sparse set of 3D points was computed from the images with a wide baseline stereo algorithm (Tuytelaars and Gool, 2000; Nabbe and Hebert, 2002). The point set was then used as the single source of information to compute the next

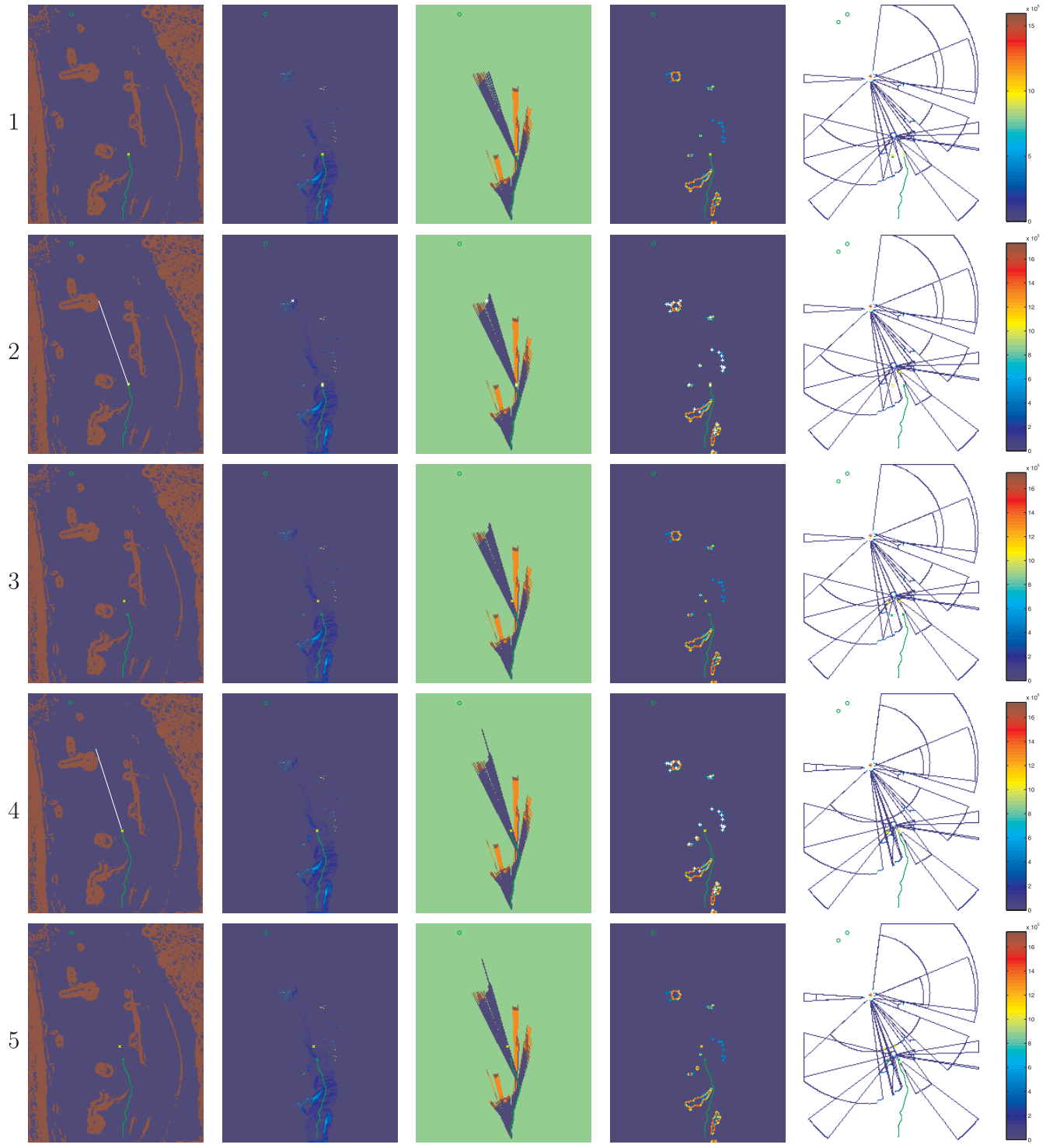


Fig. 14. In row (1), the robot is about to arrive at a waypoint. This waypoint is reached and depicted in row (2). The robot takes a mid-range sensor reading in the shown direction. It then plans for a new waypoint, which is shown in row (3). This new waypoint is reached in row (4) and the cycle repeats.

Table 1. Smith Hall parking lot run, experiment parameters.

Map size x (m)	Map size y (m)	Resolution m/cell	Distance Start-Goal	Mobility Range (m)	Mid-range Range (m)	FOV mid-range degrees	Timeout Cycles	β [0...1]
100	100	0.5	45	1	100	45	20	0.7

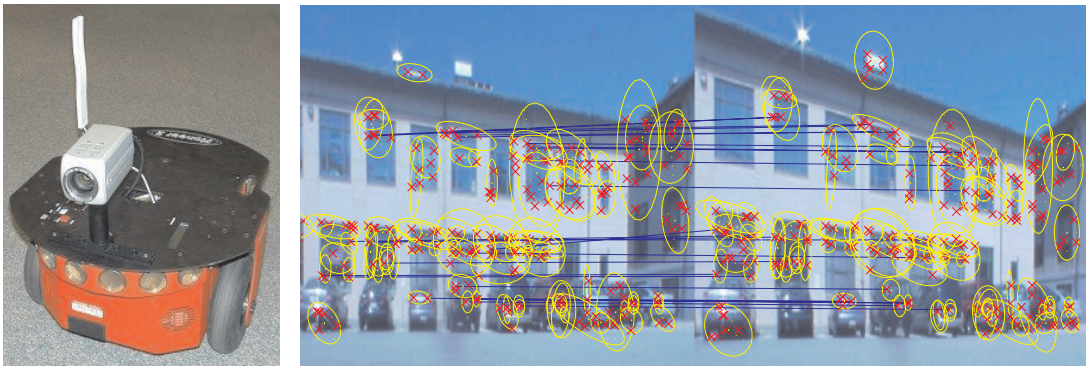


Fig. 15. The Pioneer mobile robot with camera mounted is shown on the left. Shown on the right is the example range data that was computed by a wide-baseline stereo system, using a region based approach (Nabbe and Hebert, 2002).

Fig. 16. The experiment was conducted in the parking lot, with the start in front of the building and the goal (out of view) behind it. The dark trace shows the path as computed by a mobility only approach, the lighter trace shows approximately the path from the mid-range sensing approach.

best vantage point. With this particular configuration, range measurements of up to 100m can be acquired. Figure 15 shows a typical image pair that was used to compute this data (Nabbe and Hebert, 2002). The parameters used in this experiment are summarized in Table 1.

The path that the robot followed in this example is indicated in Figure 16. The mid-range sensing/planning approach did detect that the straight path to the goal was blocked and found immediately a passage around the obstacle, as indicated by the lighter trace in Figure 17.

4.2. Experiment with Laser Range Finder

Further experiments were conducted with a robotic Deere eGator (Figure 18). This allowed us to test our approach in much more natural environments. The eGator was modified by the National Robotics Engineering Consortium to incorporate sensing, computation and actuation for autonomous navigation. It includes a SICK sensor with a range of 60 m.

Short-range mobility data is simulated by cutting off the SICK data at 4 m. Mid-range sensing is simulated by using

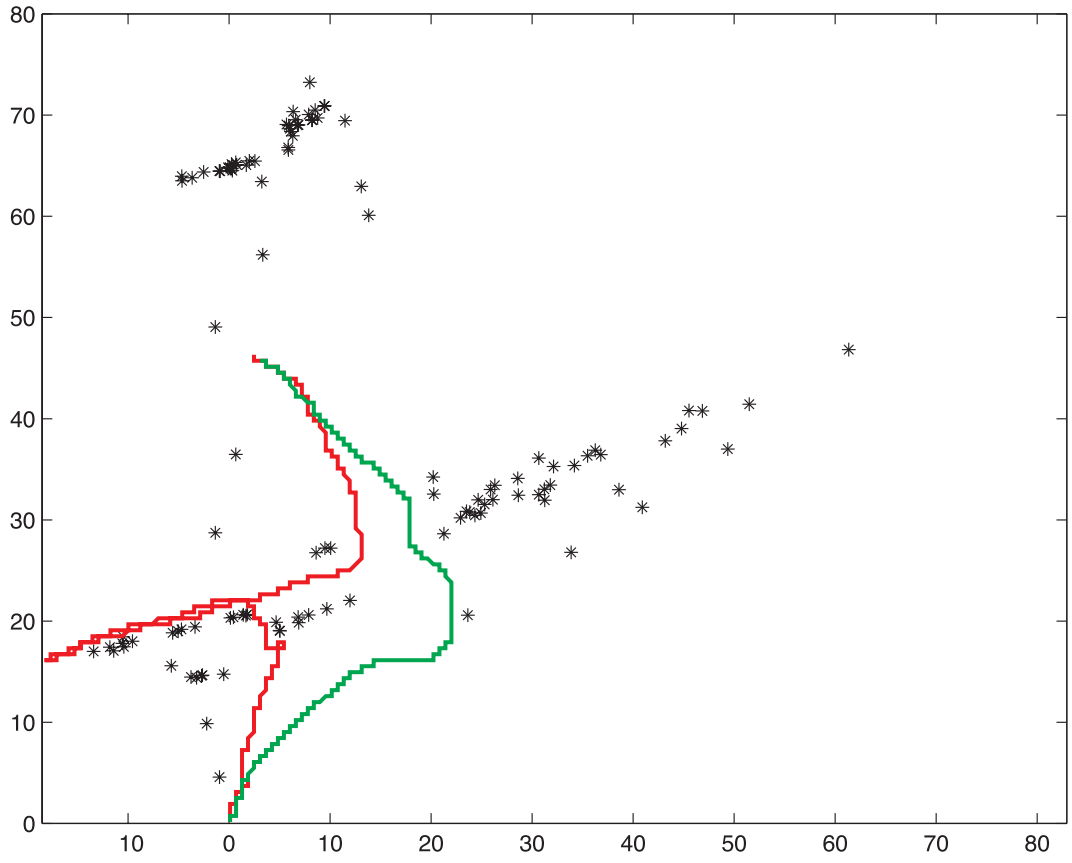


Fig. 17. A birds-eye view of the example test scenario in a structured environment, in which the mobility sensing navigation skirts around the building (darker shade) and the Mid-range sensing/planning algorithm (lighter shade) finds a passage and heads straight toward it.

Fig. 18. The robotic Deere eGator, equipped with a nodding laser range finder which gives us range measurements up to 60 m in a 100° horizontal and 45° vertical FOV.

the full range of the SICK and by limiting the field of view to 6°. Below are two example experiments that were conducted with the eGator. The first experiment was staged on the "Cut" which is a grassy area with a few trees and some artificial obstacles (Figure 19). The second example was set in Schenley park, which is slightly larger and contains a higher density of obstacles.

4.3. Detailed Analysis

The field experiments are encouraging, but it is difficult to evaluate performance over a statistically meaningful large set of conditions (environment configuration, sensor characteristics, start and goal point locations). This is particularly difficult given the wide range of variation in possible terrain configurations observed in the real world, which is very difficult to generate in physical experiments. In order to analyze more closely the performance of the algorithm, we describe an experimental protocol that allows us to run controlled experiments over a large number of different paths and terrain patches. We describe first how these experiments are carried out, emphasizing the experimental procedure.

Fig. 19. Example experiment "On the Cut": left: the robot is at its starting position, the path marked is planned by the mid-range sensing and planning framework. It runs through the planned vantage points (marked with circles) where a mid-range sensor measurement is taken. The path marked in the darker shade is taken if only mobility sensing was used. Right: the robot has reached its goal position which is behind the initial obstruction.

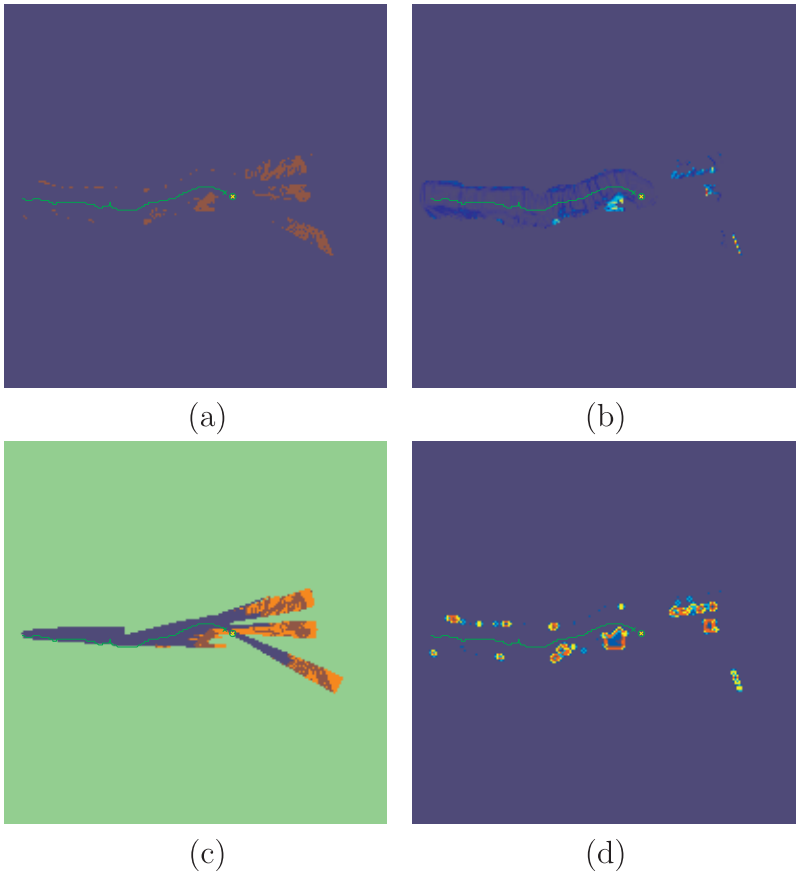


Fig. 20. Example experiment "On the Cut": The internal maps of the mid-range sensing and planning algorithm are shown after the robot completed its run: (a) the binary obstacle map as sensed during the run are shown; (b) the observed gradient from the mobility sensing system; (c) the mid-range data. The hallucinated obstacles are shown in panel (d).

Fig. 21. Example experiment "In Schenley park": Left: the robot is at its starting position, the path marked is planned by the mid-range sensing and planning framework. It runs through the planned vantage points (marked by circles) where a mid-range sensor measurement is taken. Right: the robot has reached its goal position which is behind the initial obstruction.

Table 2. "In Schenley park", experiment parameters.

Map size x (m)	Map size y (m)	Resolution m/cell	Distance Start-Goal	Mobility Range (m)	Mid-range Range (m)	FOV mid-range degrees	Timeout Cycles	β [0...1]
150	150	0.5	140	4	40	6	20	0.7

For running our controlled experiments, we have implemented a simulator that uses 2.5D elevation maps. We have used maps from both the U.S. Geological Survey and also high resolution maps from the CMU Helicopter Lab and the CMU 3D Computer Vision Group (Vandapel et al., 2003, 2004). These maps are used in this simulation as ground truth to evaluate the quality of the paths. Also the vehicle uses a simulated mobility sensor to generate the sensor data which populates the map as the vehicle progresses toward its goal. In all the experiments, we start with an empty map. The robot itself is modeled as a single point, and can freely move without any restrictions in the plane. This does not pose a serious limitation, since the view planning navigation algorithm we have presented is a high level planner, and we could ultimately use low level vehicle controller that does incorporate vehicle dynamics (Kelly, 1994).

Typically, an experiment is executed in the following way: First we acquire terrain data which is then scaled such that it reflects a real world configuration. A script generates the largest connected traversable component for which all coordinates are stored. Typical examples of randomly drawn start/goal pairs, and the corresponding paths are shown in Figure 25. Because the selection is random, some samples are "trivial" (the goal can be reached from the start through a straight line path), as shown in the left panel of Figure 25. In such cases, we do not expect to achieve any gain in path efficiency by using mid-range sensing. However, we do include these cases in the experiments because they do occur in practice and because it is important to verify that the sensor planning algorithm does not *degrade* the performance of the system. In other words, we use such cases to verify that the algorithm "does no harm" when there is no gain to be achieved. Finally, paths that span less than twice the range of the mobility sensing system are discarded since the goal in such cases is substantially closer to the start location than the maximum range of the mid-range sensor.

Additional experiments were conducted using the environment displayed in Figure 23. This terrain data from an open coal mine was collected by the CMU Helicopter Lab and is of very high spatial resolution. We sampled it down to a resolution of 0.5m/cell (Table 3).

We have executed 500 trial runs with randomly chosen start and goal positions (two of them are shown in Figure 25). We analyze the data from these runs in three ways. First, we compare the lengths of the paths generated by using the mid-range sensor planning method with the paths executed by using mobility sensing only. This is shown in Figure 24, in which the runs are sorted in the order of increasing gain for the sensor-based planning method. This first type of analysis is intended to evaluate the amount gained from using sensor planning. It is important to note that the gain can vary dramatically depending on the start and goal points. Intuitively, little gain can be expected if the area between the start and goal points is completely unobstructed, in which case *any* planning strategy would perform well.

The graph shows clearly that for an unobstructed path, the algorithm leads to a slightly longer path. This is because the vehicle might veer off the "path" in order to get better cov-

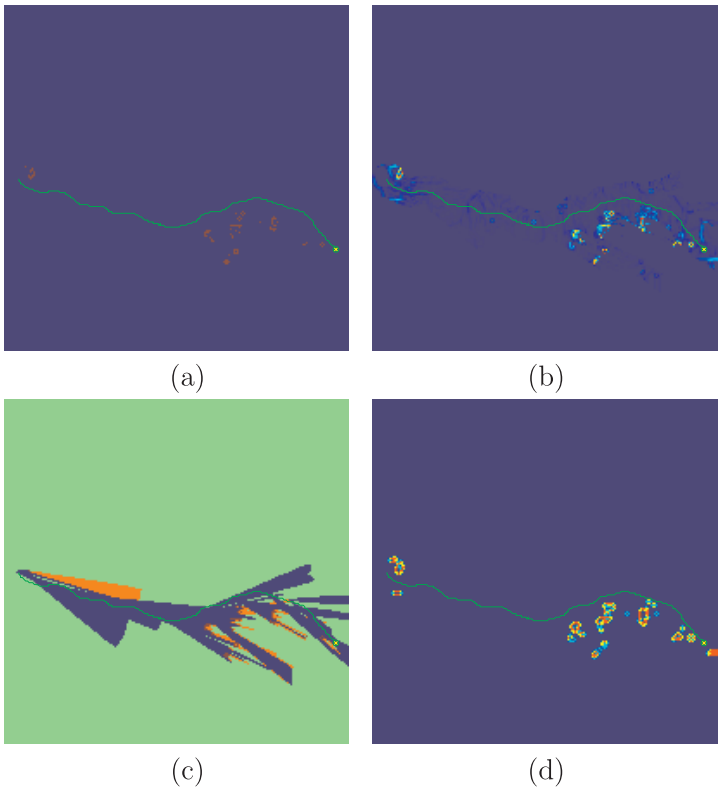


Fig. 22. Example experiment "In Schenley park": The internal maps of the mid-range sensing and planning algorithm are shown after the robot completed its run. In (a), the binary obstacle map as sensed during the run are shown. (b) shows the observed gradient from the mobility sensing system. The mid-range data is displayed in display (c). The hallucinated obstacles are shown in panel (d).

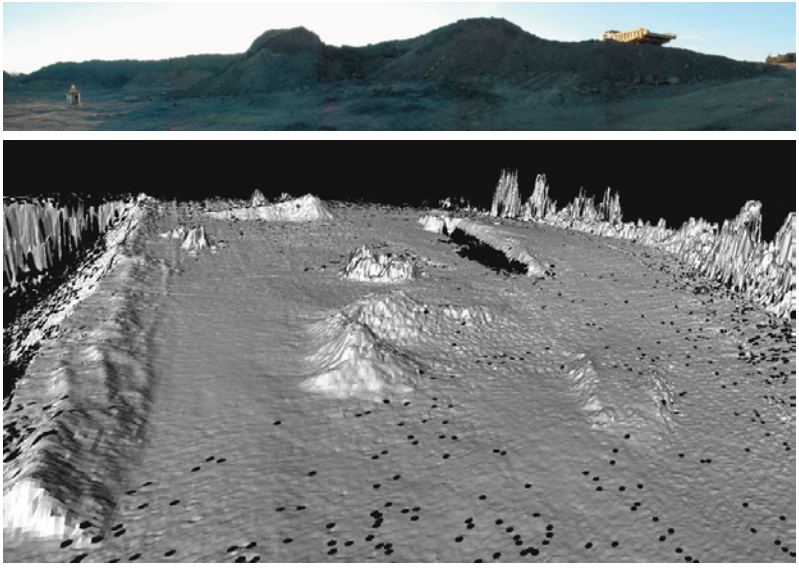


Fig. 23. Top: a panorama view of the open coal mine test site from which an elevation map was collected. Bottom: a rendering of this elevation map.

Table 3. Open coal mine data-set, experiment parameters.

Map size x (m)	Map size y (m)	Resolution m/cell	Distance Start-Goal	Mobility Range (m)	Mid-range Range (m)	FOV mid-range degrees	Timeout Cycles	β [0...1]
235	258	1	121	12	120	5	18	0.7

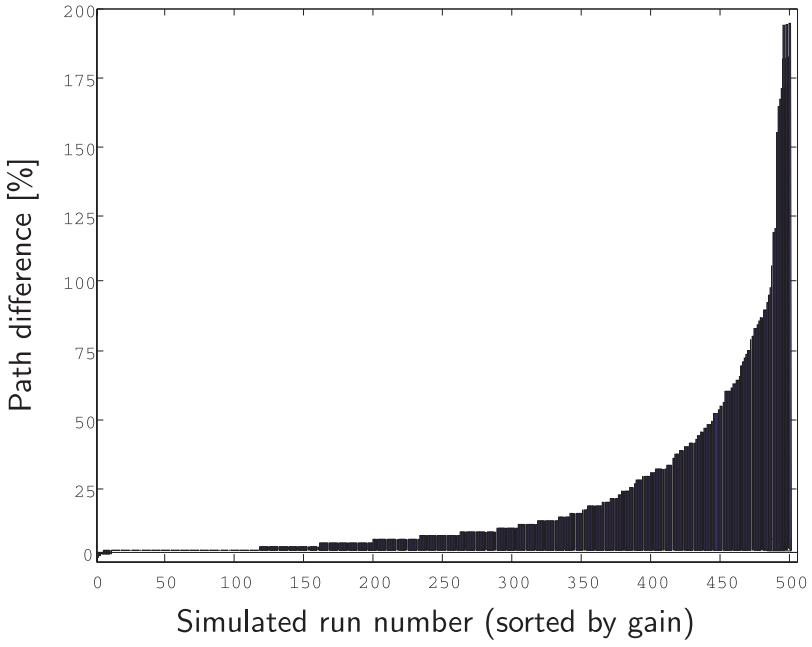


Fig. 24. The mid-range planning sensing algorithm compared to the standard navigation approach. A positive percentage shows how much shorter the path from the mid-range sense/planning is over the mobility only approach. The average gain in path length is +18%. However, more interesting is that +76% of the runs exhibit positive gain (up to almost +200% gain) and that of those that have negative gain, the maximum loss is −6%. This is due to the overhead introduced by the explore behavior that tries to find better paths. For 22% of the runs there was no gain or loss.

erage (Figure 25). Over all runs, on average the gain in path length is +18%. Also, 76% of the runs exhibit positive gain (up to almost +200% gain). Of those that have negative gain, the maximum loss is only −6%, furthermore, as the left illustration in Figure 25 shows, these cases are those for which the paths in the environment are unobstructed.

Second, it is important to compare the paths obtained with our sensor planning heuristics with the paths generated by using a mid-range sensor that senses all the time in every direction, since our claim is that the algorithm generates a "good" selection of when/where to sense during motion of the robot. If our planner were to generate paths that have substantially higher cost than those generated by using the sense all the time/everywhere strategy, it would indicate that our heuristics can be improved. This part of the analysis is summarized in Figure 27, in which the lengths of the paths generated from our planning approach and from the sense all the time/everywhere approach are plotted (with (+)) as a scatter plot. The plot indicates that the lengths are similar: i.e., the values are scattered near the diagonal (the correlation coefficient is $\rho = 0.99$). This result verifies empirically our hypothesis that continuous sensing of the environment is not necessary, provided that suitable heuristics are used for computing when and where to sense.

Finally, a third type of analysis uses the paths generated by an omniscient planner, which has complete knowledge of the entire map prior to execution, as the baseline for comparison. Such an omniscient planner generates the shortest paths that can be generated by any planner given the environment and the selections of start and goal locations. As such, it is a useful baseline to quantify the degradation of performance due to the limited sensor horizon. This analysis is shown in Figure 27 as well, in which the paths generated by using our mid-sensor planning strategy and the paths executed by the omniscient planner (marked with ○) are shown again as a scatter plot. The graph shows that the paths generated with mid-range sensor planning are almost as close to the optimal from the omni-

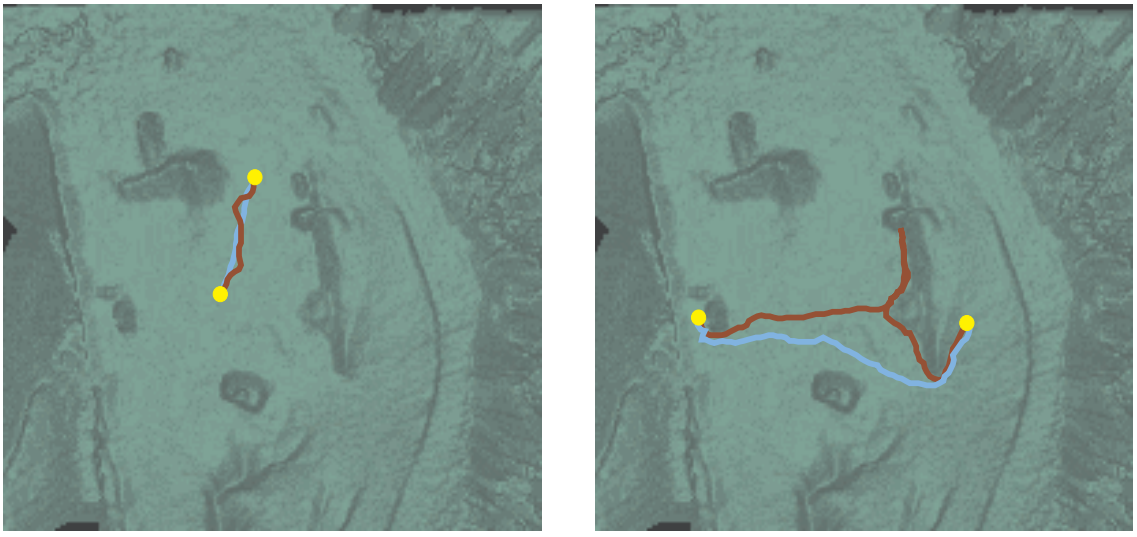


Fig. 25. Typically, bad performance is due to the fact that little gain can be expected if the area between the start and goal points is completely unobstructed. This is due to the fact that the vehicle might veer off the "path" in order to get better coverage (diagram on the left). Much gain can be found if a major obstruction is anticipated early (diagram on the right).

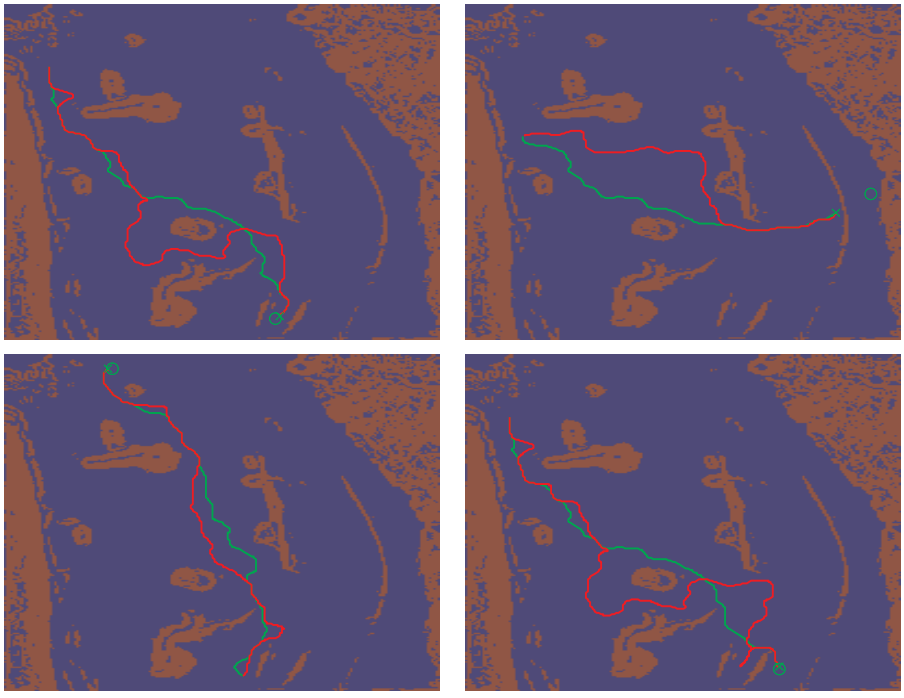


Fig. 26. A random sample of runs drawn from the set of 500. In a lighter shade are shown the paths resulting from our mid-range sensing and navigation framework. The paths shown in the darker shade are from a planner that has to rely on mobility only sensor data.

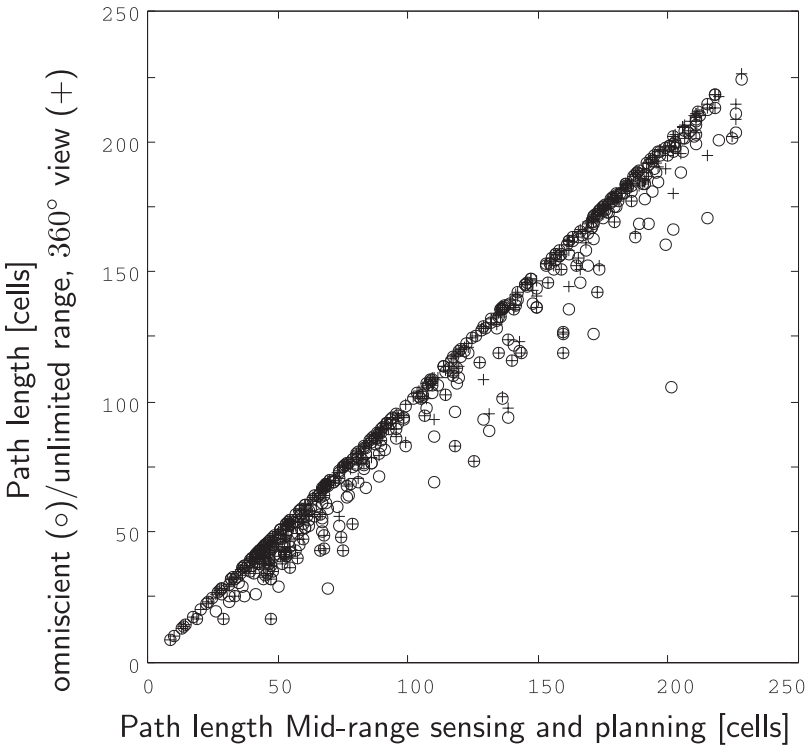


Fig. 27. The Mid-range planning sensing algorithm compared in a scatter plot to a hypothetical continuous sensing method with an unlimited range, 360° view range finder (no viewpoint planning (+)). and also compared to the true shortest path (○). The vertical axis is the path length from the omniscient planner and also the planner using continuous sensing and planning. The horizontal axis is the path length from our Mid-range sensing/planning method. Results show that the performance of our mid-range sensing approach are almost as close to the optimal as they were to the continuous sensing planner.

scient planner as they were to the continuous sensing planner. Specifically, the correlation coefficient is $\rho = 0.98$ between the paths planned by the omniscient planner and our mid-range sensor planning approach.

The same experimental procedure was used on terrain maps from the USGS database. For reasons of space, we report only the results for one data set from the vicinity of Flagstaff, Arizona (rendered in Figure 28). The parameters used in these experiments are summarized in Table 4.

As before, we first compare the lengths of the paths generated by using the mid-range sensor planning method with the paths executed by using mobility sensing only. This is shown in Figure 29. On average the gain in path length is +19%. Also, 66% of the runs exhibit positive gain (up to almost +177%). And of those that have negative gain, the maximum loss is –7%.

Second, we compare the paths obtained with our sensor planning heuristics with the plan generated by using a mid-range sensor that senses all the time in every direction. The result of this analysis is summarized in Figure 30, in which the lengths of the paths generated from our planning approach and from the sense all the time/everywhere approach are plotted (with (×)) as a scatter plot. The correlation coefficient for this experiment is ($\rho = 0.98$).

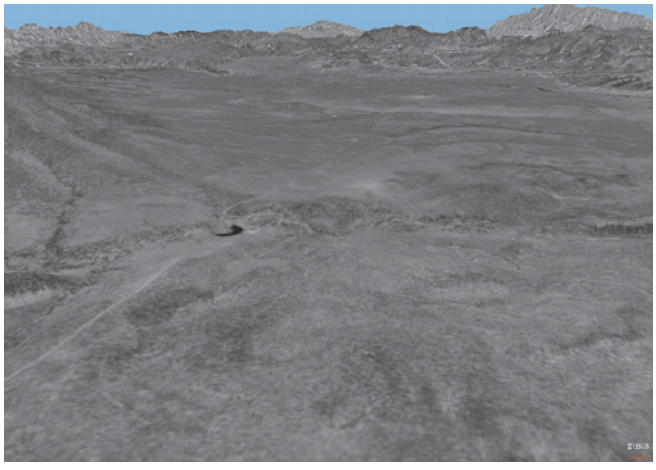


Fig. 28. A rendering of the USGS elevation map from a terrain patch in the vicinity of Flagstaff, AZ.

Table 4. Experimental parameters for the Arizona dataset.

Map size x (m)	Map size y (m)	Resolution m/cell	Distance Start-Goal	Mobility Range (m)	Mid-range Range (m)	FOV mid-range degrees	Timeout Cycles	β [0...1]
3000	2000	10	991	40	1000	8	12	0.7

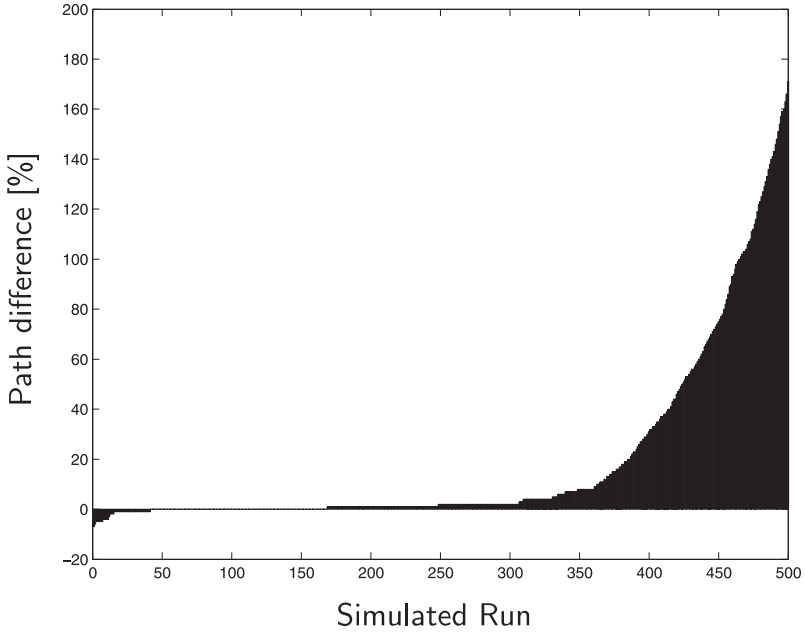


Fig. 29. The mid-range planning sensing algorithm compared to the standard navigation approach. A positive percentage shows how much shorter the path from the mid-range sense/planning is over the mobility only approach. The average gain in path length is +19%. In addition, +66% of the runs exhibit positive gain (up to almost +177%) and that of those that have negative gain, the maximum loss is −7%. For 25% of the runs there was no gain or loss.

Third, we use the paths generated by an omniscient planner that has complete knowledge of the entire map prior to execution as the baseline for comparison. This analysis is shown in Figure 31, in which the paths generated by using our mid-sensor planning strategy and the paths executed by an omniscient planner (marked with ○) are plotted again as a scatter plot. For this experiment, the correlation coefficient is $\rho = 0.97$.

4.4. Discussion

We have shown examples in which our algorithm uses mid-range laser range finder data as well as data from a wide-baseline stereo system to successfully plan view points and navigate a real robot in an outdoor environment. In addition we have performed an extensive evaluation of the framework by means of experiments with a simulator that uses real terrain data.

Our analysis shows that the mid-range sensing and planning algorithm performs on average better than a strategy based on mobility sensing only. Moreover, in many scenarios the algorithm was shown to outperform a strategy based on a mobility only sensing approach by a factor of two. The mid-range sensing strategy is outperformed only in cases in which the paths are trivial (i.e., there is no obstruction between start and goal points). In those cases, the additional penalty incurred for using the mid-range sensing strategy is at most on the order of 6% of path length across all experiments. Overall, our algorithm often performs almost as well as if using an ideal, continuous unlimited range, unlimited field of view sensing approach. Typically we find a correlation coefficient of 0.98

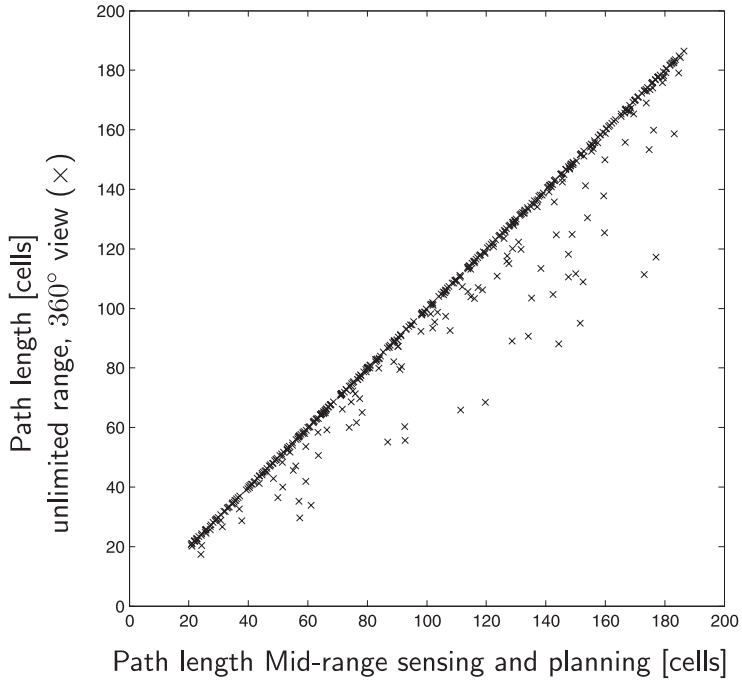


Fig. 30. The mid-range planning sensing algorithm compared in a scatter plot to a hypothetical continuous sensing method with an unlimited range, 360° view range finder (no viewpoint planning (×)). The vertical axis is the path length for the planner using continuous sensing and planning. The horizontal axis is the path length from our mid-range sensing/planning method.

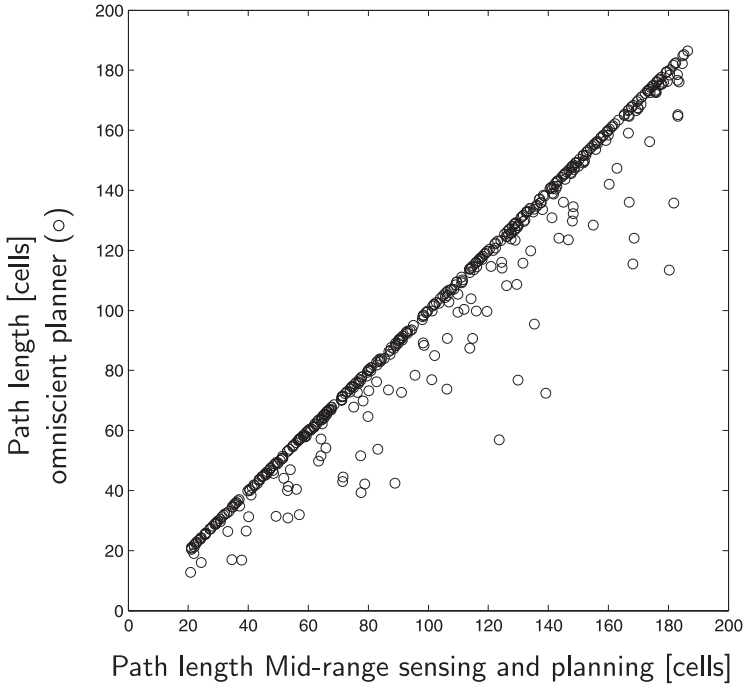


Fig. 31. The mid-range planning sensing algorithm compared in a scatter plot to the true shortest path (○). The vertical axis is the path length from the omniscient planner. The horizontal axis is the path length from our mid-range sensing/planning method.

between the path costs obtained by ideal continuous sensing and those obtained by planning mid-range sensing according our algorithm.

The experiments have also highlighted some of the limitations of the algorithm. The scenarios in which it makes sense to apply our combined view planning and navigation approach include outdoor scenarios in which the difference between the mobility sensor range and the long-range sensor is at least an order of magnitude: e.g., a few tens of meters (50 m) for the mobility sensing perimeter range and up to a few hundred meters for long-range sensors. Of course, it is always possible to construct artificial worlds for which the algorithm will perform poorly, or will even make the least optimal decision. For example, a checkerboard-like distribution of obstacles would clearly violate the smoothness assumption used in the inference. Also, it would be very difficult to make sensible sensing predictions in a complicated maze with small openings. This is not surprising: the algorithm relies entirely on hallucinating obstacles based on partial information. If the world is designed specifically such that the actual obstacles are in the opposite configuration from the most natural configuration based on sensor data, then it is bound to defeat the prediction algorithm. However, our experiments with patches random drawn from *real* terrain data, such as the USGS data, suggests that such situations do not occur in natural environments. This validates the smoothness assumption that we use as our prior model in the MRF formulation.

5. Conclusion

Navigation for ground vehicles is hindered by the limitations of sensor capabilities. The limited sensing range that is available often leads to inefficient paths because important terrain features or obstacles are not observed in a timely manner, so the navigation algorithm has no other choice than to skirt past these obstructions. The use of mid-range sensing can alleviate this problem, since these types of sensors will provide data up to a couple of hundreds of meters. However, applicability of these mid-range sensing devices is limited because of both geometric constraints and the type of data that is returned. Mid-range data is typically sparse and mid-range sensors need to be properly aimed at the region of interest such that the sensor's field of view is not limited by nearby obstructions. We attempted to address both of these issues. We proposed a novel mid-range sensing and navigation strategy that relies on a new forward simulation technique that uses hallucinated worlds to determine "when and where" to look. This forward simulation is based on a novel application of a well studied inference mechanism in a probabilistic framework to reason about the most likely future world scenario. Based on this prediction, a cost benefit analysis is performed to determine which locations on our way to a goal destination could be used to acquire more mid-range data to aid navigation.

We demonstrate that the concept of forward simulation can be brought successfully into a working algorithm. This forward simulation framework evaluates alternative scenarios for navigation or sensing based on the most likely scenario. The algorithm is shown to be able to improve navigation of a robot in both simulated and real scenarios. The sensing and navigation strategy can opportunistically reduce the path cost by 50% in comparison with a mobility only navigation strategy. The algorithm is also shown to degrade gracefully. More specifically, if the algorithm fails to identify a beneficial waypoint, it will fall back to navigating toward the goal as if our viewpoint planning algorithm was not present.

Much work remains to be done. Most importantly, we would like to extend our framework such that better prior models could be used. Although we have demonstrated examples on real autonomous vehicles, there are still improvements required to make the algorithm a true online algorithm. We used a simplified model in which the traversability data at each cell is represented only by the local terrain slope. More elaborate representation use multiple costs computed from the raw data. Also, we have used the simplest definition of path cost, *i.e.*, its length. It would be interesting to use more elaborate models that incorporate continuous costs instead of a binary obstacle/non-obstacle representation.

Finally, the current implementation of the forward simulation is based on a MRF inference mechanism. Although quite powerful in expressing local spatial groupings, the MRF is not sufficient to capture structures in real environments. Alternative inference mechanisms or possibly even a hierarchical multi-resolution MRFs can be explored in the future such that more intricate structures could be inferred. Work in the pattern recognition community is directly related to this question, and new techniques from this field could possibly be tailored to suit our needs.

Acknowledgements

This work was conducted through collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-209912.

References

Alami, R. and Simeon, T. (1994). Planning robust motion strategies for a mobile robot. *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1312–1318.

Bailey, T., Nebot, E., Rosenblatt, J., and Durrant-Whyte, H. (2000). Data association for mobile robot navigation: a graph theoretic approach. *IEEE International Conference Robotics and Automation*, Vol. 3, pp. 2512–2517.

Banta, J., Zhien, Y., Wang, X., Zhang, G., Smith, M., and Abidi, M. (1995). A best-nextview algorithm for three-dimensional scene reconstruction using range images. *SPIE*, Vol. 2588, pp. 418–429.

Bellutta, P., Manduchi, R., Matthies, L., Owens, K., and Rankin, A. (2000). Terrain perception for demo III. *Intelligent Vehicles Conference*, pp. 326–331.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society*, B-48: 259–302.

Brooks, R. and Flynn, A. (1986). A robust layered control system for a mobile robot. *Transactions on Robotics and Automation*, Vol. 2(1): 14–23.

Chen, Z. and Huang, C. (1994). Terrain exploration of a sensor-based robot moving among unknown obstacles of polygonal shape. *Robotica*, 12: 33–44.

Choset, H. (1996). *Sensor-Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. Ph.D. thesis, California Institute of Technology.

Choset, H. and Burdick, J. (1995). Sensor based planning, part I: The generalized voronoi graph. *Proceedings of the 1995 IEEE International Conference on Robotics and Automation (ICRA '95)*, Vol. 2, pp. 1649–1655.

Connolly, C. (1985). The determination of the next best view. In *IEEE International Conference on Robotics and Automation*, pp. 432–435.

Daily, M., Harris, J., Keirsey, D., Olin, K., Payton, D., Reiser, K., Rosenblatt, J., Tseng, D., and Wong, V. (1988a). Autonomous cross-country navigation with the ALV. *Proceedings of the International Conference on Robotics and Automation*, pp. 718–726.

Daily, M., Harris, J., and Reiser, K. (1988b). An operational perception system for cross-country navigation. *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 794–802.

Devy, M., Chatila, R., Fillatreau, P., Lacroix, S., and Nashashibi, F. (1995). On autonomous navigation in a natural environment. *Robotics and Autonomous Systems*, **16(1)**: 5–16.

Esposito, J. M. and Kumar, V. (2000). Closed loop motion plans for mobile robots. *IEEE International Conference on Robotics and Automation*, pp. 2777–2782.

Ferguson, D. and Stentz, A. (2005a). Field D*: An Interpolation-based Path Planner and Replanner. *Proceedings of the International Symposium on Robotics Research (ISRR)*, pp. 1926–1931.

Ferguson, D. and Stentz, A. (2005b). The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-Uniform Cost Environments. *Technical Report* CMU-RI-TR-05-19, Carnegie Mellon School of Computer Science.

Ferguson, D. and Stentz, A. T. (2004). Planning with imperfect information. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*.

Ferguson, D., Stentz, A. T. and Thrun, S. (2004). PAO* for planning with hidden state. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 2840–2847.

Gancet, J. and Lacroix, S. (2003). Pg2p: A perception-guided path planning approach for long range autonomous navigation in unknown natural environments. *IEEE International Conference on Intelligent Robotics and Systems*, pp. 2992–2997.

González-Baños, H. and Latombe, J. (1998). Planning robot motions for range-image acquisition and automatic 3d model construction. *Integrated Planning for Autonomous Agent Architectures*, AAAI Fall Symposium Series.

González-Baños, H. H. and Latombe, J. C. (2000). Robot navigation for automatic model construction using safe regions. *International Symposium on Experimental Robotics*, pp. 405–415.

González-Baños, H. H. and Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. *International Journal of Robotics Research*, **12(10–11)**: 829–848.

Goto, Y. and Stentz, A. (1987). The CMU system for mobile robot navigation. *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 99–105.

Grabowski, R., Khosla, P., and Choset, H. (2003). Autonomous exploration via regions of interest. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Greig, D. M., Porteous, B. T., and Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, **51(2)**: 271–279.

Haddad, H., Khatib, M., Lacroix, S., and Chatila, R. (1998). Reactive navigation in outdoor environments using potential fields. In IEEE, editor, *International Conference on Robotics and Automation*, Vol. 2, pp. 1232–1237.

Hait, A. and Simeon, T. (1996). Motion planning on rough terrain for an articulated vehicle inpresence of uncertainties. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 1126–1133.

Hait, A., Simeon, T., and Taix, M. (1999). Robust motion planning for rough terrain navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 11–16.

Harmon, S. Y. (1987). The ground surveillance robot (GSR): An autonomous vehicle designed to transit unknown terrain. *Robotics and Automation*, RA-3(3): 266–279.

Hebert, M., Thorpe, C., and Stentz, A. T. (1997). *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*, Dordrecht, Kluwer.

Hong, T.-H., Chang, T., Rasmussen, C., and Shneier, M. (2002a). Feature detection and tracking for mobile robots using a combination of ladar and color images. *IEEE In-*

ternational Conference of Robotics and Automation, Washington, DC, IEEE, pp. 4340–4345.

Hong, T.-H., Rasmussen, C., Chang, T., and Shneier, M. (2002b). Fusing ladar and color image information for mobile robot feature detection and tracking. *7th International Conference on Intelligent Autonomous Systems*.

Kakusho, K., Kitahashi, T., Kondo, K., and Latombe, J. (1985). Continuous purposive sensing and motion for 2d map building. In *IEEE International Conference on Systems, Man & Cybernetics*, Vol. 2, pp. 1472–1477.

Kamon, I., Rivlin, E., and Rimon, E. (1996). A new range-sensor based globally convergent navigation algorithm for mobile robots. *IEEE International Conference on Robotics and Automation*, pp. 429–435.

Kelly, A. and Stentz, A. T. (1998). Rough terrain autonomous mobility – Part 2: An active vision, predictive control. *Autonomous Robots*, **5**: 163–198.

Kelly, A. J. (1994). Ranger – an intelligent predictive controller for unmanned ground vehicles. *Technical Report*, The Robotics Institute, Carnegie Mellon University.

Kleinberg, J. (1994). On-line search in a simple polygon. *5th ACM-SIAM Symposium on Discrete Algorithms*, pp. 8–15.

Koenig, S. and Likhachev, M. (2002). Improved fast replanning for robot navigation in unknown terrain. *International Conference on Robotics and Automation*, Vol. 1, pp. 968–975.

Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *International Conference on Robotics and Automation*, Vol. 2, pp. 1398–1404.

Kruse, E., Gutsche, R., and Wahl, F. (1996). Efficient, iterative, sensor based 3D map building using rating functions in configuration space. *International Conference on Robotics and Automation*, Vol. 2, pp. 1067–1072.

Kumar, S. and Hebert, M. (2006). Discriminative random fields. *International Journal of Computer Vision*, **68(2)**: 179–202.

Kutulakos, K., Dyer, C., and Lumelsky, V. (1994). Provable strategies for vision-guided exploration in three dimensions. In *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1365–1372.

Kutulakos, K. N. and Dyer, C. R. (1992). Recovering shape by purposive viewpoint adjustment. *Proceedings IEEE Conference Computer Vision and Pattern Recognition, CVPR*, Los Alamitos, CA, IEEE Computer Society.

Lacaze, A., Moscovitz, DeClaris, Y., and N. Murphy, K. (1998). Path planning for autonomous vehicles driving over rough terrain. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 50–55.

Lacroix, S. and Chatila, R. (1997). Motion and perception strategies for outdoor mobile robot navigation in unknown environments. *The 4th International Symposium on Experimental Robotics IV*, pp. 538–547, London, Springer.

Lacroix, S., Chatila, R., Fleury, S., Herrb, M., and Simeon, T. (1994). Autonomous navigation in outdoor environment: adaptive approach and experiment. *International Conference on Robotics and Automation*, Vol. 1, pp. 426–432.

Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R. (2000). Autonomous rover navigation on unknown terrains. demonstrations in the space museum "cité de l'espace" at toulouse. In *7th International Symposium on Experimental Robotics*.

Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R. (2002). *Autonomous Rover Navigation on Unknown Terrains: Functions and Integration*. *The International Journal of Robotics Research*, **21(10–11)**: 917–942.

Langer, D., Rosenblatt, J., and Hebert, M. (1994). A reactive system for off-road navigation. *International Conference on Robotics and Automation*.

Laubach, S. and Burdick, J. (1999). An autonomous sensor-based path-planner for planetary microrovers. *International Conference of Robotics and Automation*, Vol. 1, pp. 347–354.

Laubach, S. L. (1999). *Theory and Experiments in Autonomous Sensor-Based Motion Planning with Applications for Flight Planetary Microrovers*. Ph.D. thesis, California Institute of Technology.

LaValle, S. (1995). *A Game-Theoretic Framework for Robot Motion Planning*. PhD thesis, University of Illinois, Urbana, IL.

LaValle, S. and Sharma, R. (1997). On motion planning in changing, partially-predictable environments. *International Journal of Robotics Research*, **16**: 775–805.

Li, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*, Tokyo Springer.

Lumelsky, V. J. and Stepanov, A. A. (1987). Path-planning strategies for a point mobile automaton moving amidst obstacles of arbitrary shape. *Algorithmica*, **2**: 403–430.

Maksarov, D. and Durrant-Whyte, H. (1995). Mobile vehicle navigation in unknown environments: a multiple hypothesis approach. In , editor, *Control Theory and Applications*, Vol. 142,Piscataway, NJ, IEEE, pp. 385–400.

Maurette, M. (2003). Mars rover autonomous navigation. *Autonomous Robots*, **14(2–3)**: 199–208.

Maver, J. and Bajcsy, R. (1993). Occlusions as a guide for planning the next view. *IEEE Pattern Analysis Machine Intelligence*, **15(5)**: 417–433.

McCoy, B. M. and Wu, T. T. (1973). *The Two-Dimensional Ising Model*, Cambridge, MA, Harvard University Press.

Moorehead, S. (2001). *Autonomous Surface Exploration for Mobile Robots*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Nabbe, B. and Hebert, M. (2002). Toward practical cooperative stereo for robotic colonies. *IEEE International Con-*

ference on Robotics and Automation, Vol. 4, pp. 3328–3335.

Nabbe, B. and Hebert, M. (2003). *Proceeding of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 920–927.

Nashashibi, F., Fillatreau, P., Dacre-Wright, B., and Simeon, T. (1994). 3D autonomous navigation in a natural environment. *IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 433–439.

Nilsson, N. J. (1969). A mobile automaton: An application of artificial intelligence techniques. *Proceedings of the 1st International Joint Conference on Artificial Intelligence*.

Nilsson, N. J. (1971). *Problem Solving Methods of Artificial Intelligence*, New York, McGraw-Hill, pp. 509–516.

Nister, D. (2003). An efficient solution to the five-point relative pose problem. *Conference on Computer Vision and Pattern Recognition*, **26(6)**: 756–770.

Ó'Dúnlaing, C. and Yap, C. K. (1985). A retraction method for planning the motion of a disc. *Journal of Algorithms*, **6**: 431–475.

Pacisx, E. B., Everett, H., Farrington, N., Kogut, G., Sights, B., Kramer, T., Thompson, M., Bruemmer, D., and Few, D. (2005). Transitioning unmanned ground vehicle research technologies. *Unmanned Ground Vehicle Technology VII*, Vol. 5804, Orlando, FL, SPIE.

Pai, D. and Reissell, L.-M. (1998). Multiresolution rough terrain motion planning. *Transactions on Robotics and Automation*, **14(1)**: 19–33.

Pito, R. (1996). A sensor based solution to the next best view problem. *13th International Conference on Pattern Recognition*, Vol. 1, pp. 941–945.

Rao, N., Kareti, S., Shi, W., and Iyengar, S. (1993). Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. *Oak Ridge National Laboratory Document ORNL/TM-12410*.

Rasmussen, C. (2002). Combining laser range, color and texture cues for autonomous road following. *IEEE International Conference of Robotics and Automation*, pp. 4320–4325, Washington, DC.

Rimon, E. and Canny, J. (1994). Construction of c-space road maps from local sensory data: What should the sensors look for? *International Conference on Robotics and Automation*, Vol. 1, pp. 117–123.

Rimon, E. and Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. *Robotics and Automation*, Vol. **8(5)**, pp. 501–518.

Rosenblatt, J. (1997). *DAMN: A Distributed Architecture for Mobile Navigation*. Ph.D. thesis, Carnegie Mellon University Robotics Institute, Pittsburgh, PA.

Rosenblatt, J. (2000). Optimal selection of uncertain actions by maximizing expected utility. *Autonomous Robots*, **9(1)**: 17–25.

Rosenblum, M. (2005). Immune systems are not just for making you feel better: They are for controlling autonomous robots. *Unmanned Ground Vehicle Technology VII*, Vol. 5804.

Rosenblum, M. and Gothard, B. (2000). A high fidelity multi-sensor scene understanding system for autonomous navigation. *Intelligent Vehicles Symposium*, pp. 637–643.

Rosenblum, M. and Rajagopalan, V. (2005). The road "more" traveled: A foundation for autonomous roadway operations. *Unmanned Ground Vehicle Technology VII*, Vol. 5804.

Shapiro, L. and Brady, J. (1992). Feature-based correspondence: An eigenvector approach. *Image and Vision Computing*, **10(5)**: 283–288.

Sharir, M. (1989). Algorithmic motion planning in robotics.?? *Computer*, **22(3)**: 9–20.

Sharma, R., Mount, D., and Aloimonos, Y. (1993). Probabilistic analysis of some navigation strategies in a dynamic environment. *T-SMC*, **23**: 1465–1474.

Shoemaker, C. M. and Borenstein, J. (1998). The Demo III UGV program: A testbed for autonomous navigation research. *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference* (Gaithersburg, MD), pp. 644–651.

Sofman, B., Lin, E., Bagnell, J., Cole, J., Vandapel, N., and Stentz, A. T. (2006). Improving robot navigation through self-supervised online learning. *Journal of Field Robotics*, **23(12)**: 1059–1075.

Spero, D. and Jarvis, R. (2002). Path planning for a mobile robot in a rough terrain environment. *Third International Workshop on Robot Motion and Control*, pp. 417–422.

Stentz, A. (1990). *The NAVLAB System for Mobile Robot Navigation*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. *International Conference on Robotics and Automation*, pp. 3310–3317.

Tuytelaars, T. and Gool, L. V. (2000). Wide baseline stereo matching based on local affinely invariant regions. *Proceedings British Machine Vision Conference*.

Vandapel, N., Amidi, O., and Miller, J. R. (2004). Toward laser pulse waveform analysis for scene interpretation. *IEEE International Conference on Robotics and Automation*, pp. 950–954.

Vandapel, N., Donamukkala, R. R., and Hebert, M. (2003). Experimental results in using aerial ladar data for mobile robot navigation. *International Conference on Field and Service Robotics*.

Volpe, R., Balaram, J., Ohm, T., and Ivlev, R. (1996). The Rocky7 Mars rover prototype. *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Vol. 3: 1558–1564.

Whaite, P. and Ferrie, F. (1990). From uncertainty to visual exploration. *3rd International Conference on Computer Vision*, pp. 690–697.

Wixson, L. (1994). Viewpoint selection for visual search. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 800–805.

Won, C. S. and Derin, H. (1992). Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP*, **54**: 308–328.

Ye, C. and Borenstein, J. (2004). A method for mobile robot navigation on rough terrain. *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3863–3869.

Zelinsky, A., Jarvis, R., Byrne, J., and Yuta, S. (1993). Planning paths of complete coverage of an unstructured environment by a mobile robot. *International Conference on Advanced Robotics*, pp. 533–538.