Path Planning with Hallucinated Worlds

Bart Nabbe, Sanjiv Kumar and Martial Hebert
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{bana,skumar,hebert}@cs.cmu.edu

***Abstract*— We describe an approach that integrate mid-range sensing into a dynamic path planning algorithm[1]. The algorithm is based on measuring the reduction in path cost that would be caused by taking a sensor reading from candidate locations. The planner uses this measure in order to decide where to take the next sensor reading. Ideally, one would like to evaluate a path based on a map that is as close as possible to the true underlying world. In practice, however, the map is only sparsely populated by data derived from sensor readings. A key component of the approach described in this paper is a mechanism to infer (or ”hallucinate”) more complete maps from sparse sensor readings. We show how this hallucination mechanism is integrated with the planner to produce better estimates of the gain in path cost occurred when taking sensor readings. We show results on a real robot as well as a statistical analysis on a large set of randomly generated path planning problems on elevation maps from real terrain.**

I. Introduction

As mobile autonomous systems have become more and more capable, the tasks that they can perform have also scaled up. In the early stages of mobile robots, robots were able to navigate around in artificially constructed worlds and mazes. Currently, most mobile robotic applications are still in the domain of constrained environments, such as for example, indoor applications. However, mobile autonomous systems have now entered into the realm of unstructured environments, with the Demo III [23], Perceptor [29] and DARPA Grand Challenge programs as well as the Mars rover expeditions being the best known.

The core technology that allows these autonomous systems to succeed is a good representation of the environment in which they operate. However, if these environments are not known beforehand and if there is no structure to exploit, traditional navigation and sensing methods will fail. Specifically, they will suffer from the so called myopic planning effect [18]. This effect is due to the fact that the planner is limited by the maximum range of the mobility sensors. Since typical mobility sensors, such as LADAR (mobility laser range) or passive stereo vision, will only acquire data up to a few tens of meters, the planner has no knowledge about what to encounter 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.

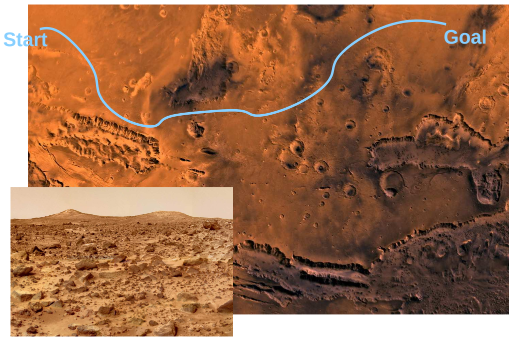

Fig. 1. An illustration of a long range navigation task in unknown Martian terrain. If major obstacles can be identified in an early stage, the autonomous system can plan a path accordingly. (From http://photojournal.jpl.nasa.gov)

For autonomous navigation over long distances, this limitation degrades 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 relative short range of the mobility sensors forces the vehicle to drive closer to terrain obstructions than is safe or necessary.

Sensing solutions that could provide the planner with information about the environment at much longer range are becoming available. These so called “mid-range sensors” are capable of partially reconstructing the three-dimensional environment of the vehicle at longer range, e.g., up to 500m. These sensors are based either on reconstructing structure by integrating long sequences of monocular images acquired as the vehicle moves [20], or on reconstructing the structure by stereo matching across widely separated views either from multiple vehicles or from a single vehicle at two different positions [1]. In addition we refer to sensing systems for obstacle avoidance as “mobility sensing”, these include sonars, short baseline stereo and laser range finders which provide data up to a maximum of a few tens of meters [5].

With mid-range data available to the path planner, it can anticipate obstacles ahead of time and can therefore plan a more optimal path to the goal. However, there are several major issues that need to be addressed first for this to work.

First, it is not practically feasible to construct such a large map of the terrain in all directions around the vehicle since it would require a prohibitive amount of

[1] This work was supported by the Army Research Labs Robotics Collaborative Technology Alliance Contract DAAD 19-01-2-0012

computation and special hardware to achieve sensor coverage. Therefore, it is important that we only focus sensing and computation to those areas that are of importance to the path planner. Second, all the mid-range sensing techniques require more costly data collection and processing procedures than the real-time sensing strategies used for mobility. Therefore it is not possible to continuously scan the environment and process the data. An active strategy is needed to decide when to incur extra sensing cost to aid collecting terrain data critical for navigation.

We have proposed an approach to address these issues in [19]. However, even with optimal sensor planning, the planning decisions are still based on partial data from the sensors. The effectiveness of the planner is greatly enhanced if we could *infer* (or *hallucinate*) a more complete representation of the underlying world from the data. Figure 2 illustrates this idea. In Figure 2(a), the partial map built by the robot's sensors suggests that a gap exists between the two obstacle regions at the center of the map, leading to the path from the current robot position to the goal shown in Figure 2(a). However, *if* the gap were actually blocked by as-of-yet undiscovered obstacles, the robot would incur a much higher-cost path by skirting the obstacle regions by using the shorter-range mobility sensors (this path is shown in red in Figure 2(b)). If we had discovered the occlusion earlier by using mid-range sensing, the robot would have used a more efficient path (shown in blue in Figure 2(b)). Therefore, the right sensing strategy would have been to take one mid-range sensor reading in the direction of the potential occlusion.

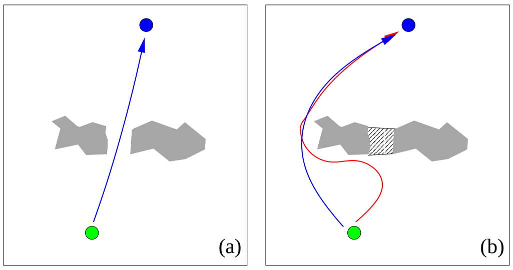

Fig. 2. The partial map in (a) suggests a gap between the two obstacle regions. A large deviation from the path will be incurred if the gap is actually blocked (b).

To implement such a strategy, the question that needs to be answered is: How do we detect automatically such ”interesting” parts of the environment at which to take sensor readings? In [19], we used simple heuristics to predict the location of such areas. In this paper, we propose a principled approach to inferring likely obstacle configurations from sparse data and from a probabilistic model of typical configurations of natural environments. We summarize the framework from [19] (with a different formulation) in Section III. We show how the problem of predicting map configurations can be implemented by using a probabilistic formulation and how it can be integrated in the sensor planning approach in Section IV. We conduct experiments on real and synthetic data in Section VI. These experiments parallel closely the experiments of [19], with the crucial difference that the predictions used in the sensor planning are generated by using the inference techniques of Section IV.

II. Prior work

The work that we previously presented [19] identified two main bodies of work, sensor planning and coverage planning, that are related to the framework described here. In addition, there is also related work in the computer vision community that is of importance to the creation of larger volumes of data from sparse sets. We will briefly discuss the relevance of this work.

When we consider the literature that describes sensor planning as an active vision problem, the work concentrates mostly around hand-eye coordination [27] or view planning for dense reconstruction [15][11]. All these techniques exploit the fact that the environment is constrained and that there is a specific object of interest.

The second category looks at the planning for sensing problem by formulating it as a coverage or an exploration problem. Examples of work in this area are [7] which covers the well known art gallery problem, Moorehead [16] who focuses on coverage. Some of the concepts about information gain as presented in his work have a similar flavor to the work presented here. González-Báños [8] focuses also on coverage. It is an integrated sensing planning approach, however his method does not allow for incremental updates on already covered areas if more accurate data becomes available.

In [12] and [13] Laubach presents a navigation strategy based on the “Bug” family of path planners. It does incorporate sensing and path planning as an integrated solution. It does however have some drawbacks that make it less usable for navigating in unstructured environments. It has no notion of uncertainty in sensor data and it needs continuous detectable obstacle boundaries. It also only senses at discrete decision points. In addition, the traversability cost is limited to a binary representation, traversable or non-traversable, which can not express costs of traversing for example, vegetated terrain. In addition, its requirement for continuous obstacles prevents the use of sparse mid-range data.

None of these approaches use any prior learned model to infer denser world models from sparse representations. The work presented in this article was inspired by the data hallucination framework as presented in [2] and is well known in the vision community. Applying data hallucination for path planning is a novel application of this concept. The work presented in [6] and [17] uses a Bayesian framework to estimate map structure to aid in localization. Which is very much in the same flavor as the work we describe here, however, their method relies on a structured environment and is not usable for robot navigation.

III. Sensor Planning

We propose a solution to incorporate the usage of mid-range data with forward simulation and data hallucination.

At the core of this solution is a grid based D* planner [25], [26]. The D* planner takes as input a collection of cost maps that describes the current knowledge about the world. In our case, we use D* to compute the best path from the current vehicle location to the goal point as usual, but, in addition to the usual obstacle costs, we also use costs that reflect the utility of mid-range sensing from different locations in the map and to denote traversable areas according to our hallucinated world model. With these extra terms added to the planners cost function, the planner will attempt to find a path through traversable terrain, that still avoids obstacles but that also maximizes the utility of sensing.

This approach of incorporating utility into a cost function is similar to the approach Rosenblatt [21][22] takes in his arbiter. However, his utility map does only look at a few possible actions over a limited horizon and does not deal with additional utility for sensing purposes, nor does it have any notion of what to anticipate in the future.

With this approach we address two key issues in planning:

- Active planning for mid-range sensing, addresses which locations the robot could visit in order to collect mid-range data that would be beneficial to the path planner.
- Anticipating traversability, this will be addressed by hallucinating worlds based on sparse evidence from sensor data.

In our approach, the planner uses a cost computed at every location of a grid covering the extent of the map. We decompose the cost into three parts: The current state of the world ρ, the added utility for sensing purposes ψ and a heuristic cost that incorporates mid-range and hallucinated data φ.

ρ is the actual cost of traversing the cell based on data from the short range obstacle detection system. This is the cost that is used in the simplest mobility system in which mobility sensors, such as LADAR or stereo, insert obstacles in the world map. In practice we use a normalized and smoothed version of the observed terrain gradient:

$$\rho = f\left(\frac{|\nabla\left(G_\sigma \otimes \text{ObservedTerrain}\right)|}{\text{ObstacleThreshold}}\right) \quad (1)$$

with

$$f(x) = \begin{cases} x & \text{if } x < 1 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

While this is a simple cost model for traversability, this function can be arbitrarily complex and might encode many different other costs, our framework does not restrict the choice of this model [22],[24].

ψ is the (inverse) benefit of visiting a particular cell based on its added utility for sensing purposes. Lower values indicate a vantage point for the mid-range sensor that is most beneficial given the current world knowledge and the currently planned path. This utility captures the trade off between going to the goal directly or going to the goal via an intermediate observation location.

φ expresses our knowledge about the traversability of a grid cell as discovered by the mid-range sensing system and expanded by the hallucination algorithm. This cost is similar to ρ in that it encodes the local obstacle-ness of the terrain. It is useful to separate the two costs because, we have less confidence in the hallucinated world.

The combination of these costs, C, captures the notion of utility for sensing purposes and will also make cells that are anticipated to be traversable based on our both actual and hallucinated data more favorable. In the planner, the cost are combined as

$$C = \begin{cases} 1 & \text{if } \rho = 1 \\ a\rho + b\psi + c\varphi & \text{otherwise} \end{cases} \quad (3)$$

with

$$a = \epsilon, b = (1-\epsilon)\tau, c = (1-\epsilon)(1-\tau) \quad (4)$$

where the parameter ϵ controls the amount of deviation from the behavior of a traditional mobility-based path planner. If $\epsilon = 0$, the vehicle executes the default path computed from the traversability maps alone, it will not do any sensor planning, nor will it exploit the mid-range and hallucinated data. τ captures the level of exploration the robot might do while on-route. If $\tau = 0$, the robot will stay close to its current most promising path. With increasing τ, the robot will stray off its path to visit observation locations with a good vantage point on its way.

The computation of the combined utility value needs not be restricted to this simple linear combination, as long as this function is admissible. That is, the cost has to increase monotonically with ρ, φ, ψ.

IV. Hallucinating data for planning

Following our approach from [19], the sensing cost ψ is based on evaluating the gain obtained by taking a sensor reading for a given robot location $\mathbf{x}\epsilon\mathbb{R}^2$, observation location $\mathbf{u}$ and goal location $\mathbf{v}_G\epsilon\mathbb{R}^2$ (Figure 3). The utility for visiting $\mathbf{u}$ while navigating to the goal is defined as:

$$\psi_{\mathbf{u}}(L, \mathbf{x}, \mathbf{v}_G) = 1 - \frac{C_{ns}(L, \mathbf{x}, \mathbf{v}_G) - C_s(L, \mathbf{x}, \mathbf{u}, \mathbf{v}_G)}{C_{ns}(L, \mathbf{x}, \mathbf{v}_G)} \quad (5)$$

In which L is the obstacle Labeling of the world map cells, $C_{ns}(L, \mathbf{x}, \mathbf{v}_G)$ the path cost for navigating from $\mathbf{x}$ to $\mathbf{v}_G$ without mid-range sensing and $C_s(L, \mathbf{x}, \mathbf{u}, \mathbf{v}_G)$ represents the path cost for navigating from $\mathbf{x}$ to $\mathbf{v}_G$ via an intermediate location $\mathbf{u}$ for collecting mid-range sensor data at $\mathbf{u}$. (Figure 3). Since the goal location ($\mathbf{v}_G$) is assumed to be fixed we will avoid overloading the notation by omitting $\mathbf{v}_G$. Therefore, the total Benefit, $\psi_{\mathbf{u}}$, at a position $\mathbf{u}$ averaged over all possible labelings of the world, for visiting a particular observation location while on-route to goal, $\psi_u(L, \mathbf{x}, \mathbf{v}_G)$ can be expressed as:

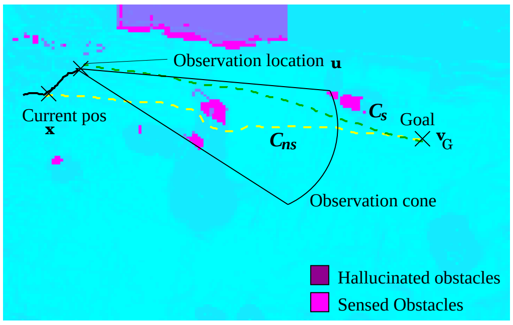

Fig. 3. Illustration of the path computed with (C_s) and without sensing (C_{ns}) in a most likely hallucinated world configuration.

$$\begin{aligned}\psi_u(L,\mathbf{x},\mathbf{u},\mathbf{v}_G) &= E_{P(L|D)}\left[\psi_{\mathbf{u}}(L,\mathbf{x},\mathbf{u},\mathbf{v}_G)\right] \\ &= \sum_{L\epsilon\mathcal{L}} P\left(L \mid D\right)\psi_{\mathbf{u}}(L,\mathbf{x},\mathbf{u},\mathbf{v}_G)\end{aligned} \quad (6)$$

D, the data, is the gradient of the terrain at every cell of the map. $\mathcal{L}$ is the set of all possible labelings of L. Note that each value of L represents a world model (configuration of obstacles) that is consistent with the data but is not actually observed.

A. *MRF Model*

Each L is a world configuration hallucinated from the data. If $\mathcal{N}$ is the number of cells in the map, $|\mathcal{L}| = 2^{\mathcal{N}}$ becomes computationally intractable in practice even with small values of $\mathcal{N}$. In order to make it possible to compute $\psi_{\mathbf{u}}$ we will use a common approximation that assumes that $P(L \mid D)$ can be approximated by a delta function [10]. This amounts to replacing the average configuration of the world with the most probable configuration.

$$P(L \mid D) \approx \delta(L, \hat{L}) \quad (7)$$

with

$$\hat{L} = \operatorname*{argmax}_{L} P(L \mid D) \quad (8)$$

Intuitively, $\hat{L}$ is the most likely world configuration inferred from the observed data.

The next problem we need to address is how to compute $P(L \mid D)$. 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 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 dependencies in the labels on 2D lattices has been well studied in computer vision tasks and is often referred to as spatial smoothness.

A popular approach for representing spatial smoothness is to use Markov Random Fields (MRFs), which can incorporate local contextual constraints in labeling problems in a principled manner [14]. MRFs are generally used in a probabilistic generative framework that models the joint probability of the observed data and the corresponding labels. In other words, let D be the observed data from the terrain, where $D = \{d_i\}_{i\epsilon\mathcal{S}}$ and $d_i\epsilon\mathbb{R}$, d_i is the data from the i th site, and $\mathcal{S}$ is the set of sites. Let the corresponding labels at the terrain sites be given by $L = \{l_i\}_{i\epsilon\mathcal{S}}, l_i$ and $l_i\epsilon\{-1,1\}$. In our case, l_i indicates the presence or absence of an obstacle at location i. In the MRF framework, the posterior over the labels given the data is expressed using the Bayes rule as,

$$P(L \mid D) \propto p(L,D) = p(D \mid L)P(L) \quad (9)$$

B. *Data Term*

The first term in the product, $p(D \mid L)$, in Equation 9 is known as the likelihood of the data. In general, it is assumed that the data at each site is conditionally independent given the labels at that site. i.e. $p(D \mid L) = \prod_{i\epsilon S} p(d_i \mid l_i)$ [3][14]. We model the likelihood for each class, traversable/non-traversable, as a Gaussian.

$$P(d_i \mid l_i) = \frac{1}{\sqrt{2\pi\sigma_{l_i}^2}} \exp\left\{\frac{1}{2\sigma_{l_i}^2}(d_i - \mu_{l_i})^2\right\} \quad (10)$$

The parameters of the two Gaussians representing the likelihood for these classes can be easily learnt from labeled training data. This learning takes place for a typical type of environment where the robot is to navigate in, for example on Mars, an exemplar is used for learning where after the robot will use these parameters for online classification/hallucination of terrain. For the current model, this learning boils down to fitting a single Gaussian two each of the two classes.

C. *Field Model*

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 except at discontinuities. Since we have only two classes (traversable/nontraversable), this can be expressed in a MRF formulation of a binary classification problem. The data likelihood $p(D \mid L)$ is assumed to be conditionally independent given the labels and the label interaction field $P(L)$ is assumed to follow a homogeneous and isotropic Ising model with only a pairwise interaction term: $\log(P(L)) = \beta l_i l_j$ where β is the interaction parameter of the MRF. The Ising model favors neighboring sites with the same labels and penalizes the dissimilar labels by cost β [10].

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:

$$P(L \mid D) = \frac{1}{Z_m} \exp\left(\sum_{i\epsilon\mathcal{S}} \log p(d_i \mid l_i) + \sum_{i\epsilon\mathcal{S}} \sum_{j\epsilon\mathcal{N}_i} \beta l_i l_j\right) \quad (11)$$

In which Z_m is a normalizing constant, often referred to as the partition function. For this MRF (Equation 11), 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 $\beta > 0$ [9][3]. With such a min-cuts algorithm [10], we have now a way of computing $\hat{L} = \mathrm{argmax}_L\, P(L \mid D)$.

D. Inference

The min-cuts/max-flow algorithm, that solves the MRF equation (11) produces a labeling $\hat{L}$ of traversable/nontraversable cells that is most consistent with the world model and the current data. This binary labeling of traversable and nontraversable cells is updated at every cycle with the current data and represents the most anticipated configuration of the world (Figure 4).

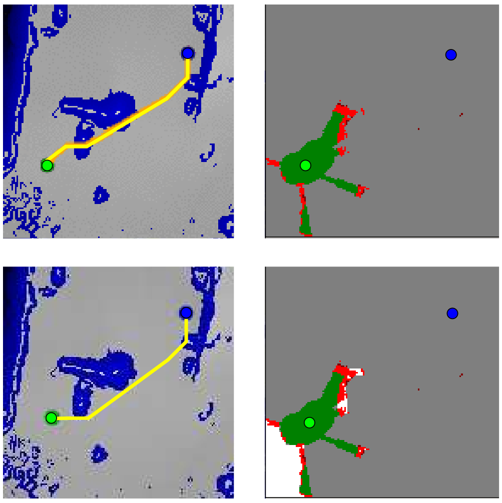

Fig. 4. Typical example that shows the influence of the obstacle hallucination. The robot (in green) travels to the goal (blue). The path is shown in yellow. The hallucinated obstacles are in white, hallucinated empty space in gray, observed traversable in green and observed obstacles are red. In the top panel, the planner uses only the information from the sensors and it plans a path trough the obstacle. In the bottom panel, the hallucination algorithm merged obstacles correctly together and produced a better path.

To illustrate the operation of the obstacle inference algorithm, we have included an example experiment (Figure 5 and Figure 6). In Figure 5, the example terrain is shown with its ground truth obstacle labeling. The planner simulated a robot traversal from the red to the yellow markers.

The number of map cells that are detected as obstacles at every position along the path is shown in Figure 6, 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 5(left) 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.

Initially, there is no data available, therefore the hallucination algorithm has no evidence for 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, when even more of the environment becomes known and inferred obstacles get confirmed, the number of inferred obstacles decreases. In the limit, when all cells have been explored there will be no inferred obstacles.

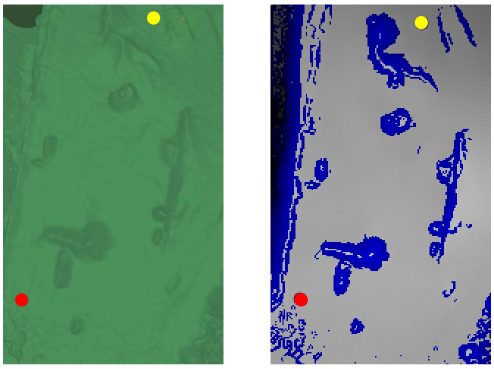

Fig. 5. The terrain that was used in the example to show the influence of the obstacle hallucination over the course of a typical traverse. On the left is a perspective view of the terrain, with start and goal location marked. The ground truth obstacle labeling is displayed on the right. Obstacle cells are marked in blue.

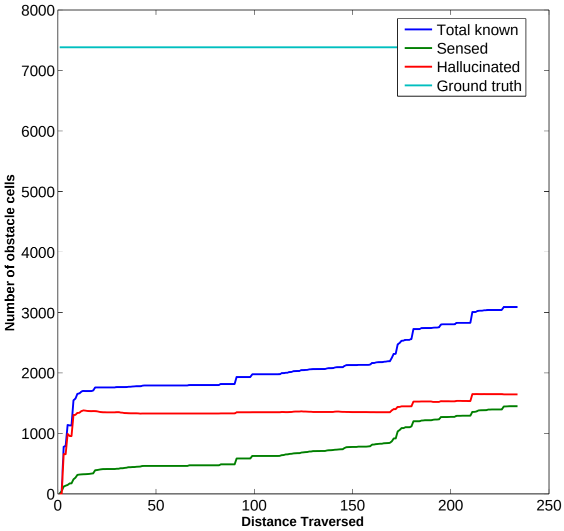

Fig. 6. The number of obstacle cells is plotted over the course of an example trajectory. The graphs confirm the intuition that while more of the environment gets explored, less needs to be inferred. In the limit, the number of inferred cells will go to zero and the number of total known obstacle cells will equal the number for the ground truth labeling.

V. Implementation Details

Smoothing

Since we want to combine the hallucinated traversability map into a smooth cost landscape for our planner we use a smoothed version of this hallucinated map to guide our robot over anticipated traversable terrain:

$$\varphi_{\mathbf{x}} = G_\sigma \otimes L \tag{12}$$

In which L is the hallucinated map and G_σ a Gaussian kernel with mean σ.

Restricting Sensor Orientation

The formulation of the benefit cost as defined in Equation 5 assumes an omni-directional sensor. Since, in practice, a mid-range sensor will have a limited field of view, our model needs to be adapted to compensate for this limitation. We can easily do so by taking the maximum of the sensing cost ψ over all possible sensor orientations for a sensor with a limited field of view:

$$\psi_{\mathbf{u}}(L,\mathbf{x},\mathbf{v}_G) = 1 - \operatorname*{argmax}_{\theta} \left(\frac{C_{ns}(L,\mathbf{x},\mathbf{v}_G) - C_s(L,\mathbf{x},\mathbf{u},\theta,\mathbf{v}_G)}{C_{ns}(L,\mathbf{x},\mathbf{v}_G)} \right) \tag{13}$$

As we have discussed in previous work [19], this is computationally expensive. However, we also showed that we can restrict ourselves to only looking at interesting areas. We would therefore fix θ in such that it would be looking at a "Point Of Interest" (POI) (Figure 7).

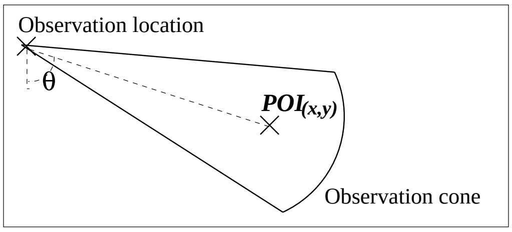

Fig. 7. The relation between θ and (x_{poi}, y_{poi})

This point of interest is chosen to be at a location that is likely to be traversed, has not been seen before, or is in an area cluttered with obstacles. This can be expressed with the following definition:

$$\mathrm{POI}_(x,y) = \operatorname*{argmax}_{(x,y)} \left(\sum_{(x',y')\in A} \left[r\left(1 - \vartheta_{(x',y')}\right) \left(1 - \left|2\left(\kappa_{(x',y')} - \frac{1}{2}\right)\right|\right)\right]\right) \tag{14}$$

In which r is the (inverse) distance from the current path, ϑ is equal to 1 if the cell has been observed and κ expresses traversability. This traversability is the union of both the traversability from both the mobility data and the hallucination framework: $\kappa = \max(\rho, \varphi)$.

When to Look

The framework as presented so far, has a way of evaluating the benefit of sensing. This utility is used in two ways: First it is used to decide where to sense. Since the utility is folded into the navigation cost, the benefit ψ will guide the robot towards locations that are advantageous for taking mid-range sensor measurements. Second, it is used to decide when to sense to to avoid taking measurements continuously. If the benefit at the current location $\psi_{\mathbf{x}}$ is above a threshold the robot will take a sensor measurement.

VI. Discussion

The experiments we have conducted on real and synthetic data parallel closely the experiments of [19], with the crucial difference that the predictions used in the sensor planning are generated by using the inference techniques presented in the previous sections.

We conducted an example experiment in an outdoor environment in which we compare our approach to the standard mobility sensing and planning approach.

For this purpose, we had our Pioneer DX robot navigate across the parking lot to find a goal behind a building. The path that the robot followed is indicated by a dark trace in Figure 8. Data was previously collected along this path with a wide baseline stereo algorithm [19]. The points were used then as the single source of information in the batch algorithm to provide the mid-range sensing/planning algorithm with data. The batch algorithm used a camera model that was identical to the camera on the robot. 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 9. The path as found by the mid-range sensing approach had a total length of 63 meters and is 56% shorter than the path generated by mobility sensing only.

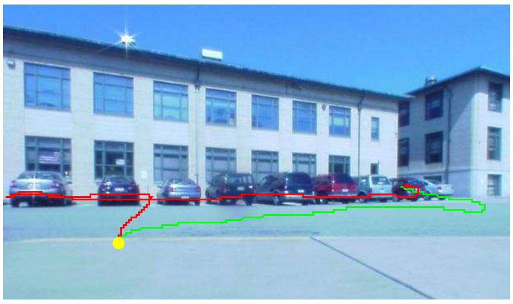

Fig. 8. The experiment was conducted on the parking lot, with the start in front of the building and the goal (out of view) behind it. The dark trace sketches the path as computed by a mobility only approach, the lighter traces shows approximately the path from the mid-range sensing approach.

We also ran experiments in simulation. For the terrain we used real terrain data from the USGS database. Different start and goal positions were chosen randomly within the constraint that there exists a path between start and goal. The parameters for this example can be found in table I. Results from 100 trial runs were collected.

Map size x (m)	Map size y (m)	Resolution m/cell	Average distance Start-Goal (m)
3000	2000	10	929

ϵ	τ	Mobility Range (m)	Mid-range Range (m)	FOV mid-range deg
0.4	0.6	40	1000	8

TABLE I

The parameters for the 100 trial run experiment.

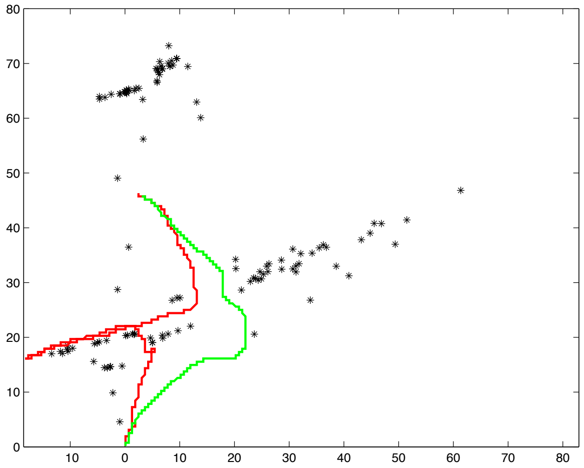

Fig. 9. a birds-eye view of the real test scenario, in which the mobility sensing navigation skirts around the building (dark) and the Mid-range sensing/planing algorithm (light) finds a passage and heads straight towards it.

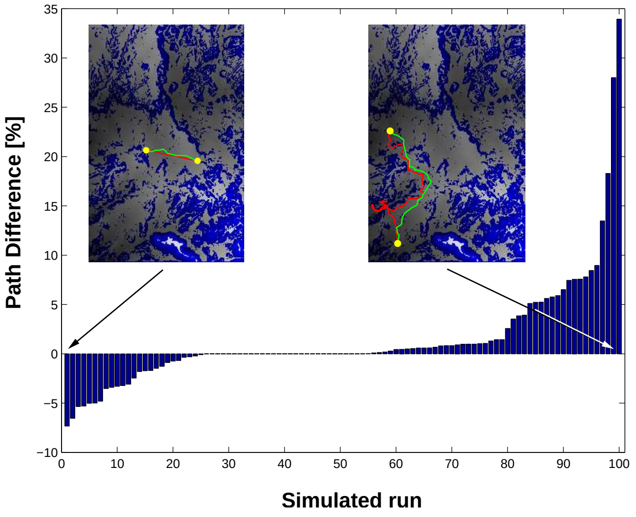

Fig. 10. The Mid-range planning sensing algorithm compared to the standard navigation approach. We used the same presentation of the data as in our previous work since the same quantities were measured. Note however that the results are based on the new data hallucination approach. A positive percentage 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 2%. However, more interesting is that 47% of the runs exhibit positive gain (up to 34%) and that of those that have negative gain, the maximum loss is -7%. This is due to the overhead introduced by the explore behavior that tries to find better paths. For 27% of the runs there was no gain or loss.

Figure 10 compares the lengths of the paths generated by using the mid-range sensor planning method with the paths executed by using mobility sensing only. The runs are sorted in order of increasing gain for the sensor-based planning method. This analysis allows us to assess 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 due to the fact that the vehicle might veer off the "path" in order to get better coverage. On average, the reduction in path length is 2%. However, more interesting is that our results show that 47% of the runs exhibit positive gain (up to 34%). And that of those that have negative gain, the maximum loss is only -7%, furthermore, as the left illustration in the figure shows, these cases are those for which the paths in the environment are unobstructed.

It is also interesting to compare our method against an approach in which the system senses all the time/everywhere, since this will be the upper bound of how well our algorithm could perform. If our planner were to generate paths that are substantially higher cost than those generated by using the sense all the time/everywhere strategy, it would indicates that our heuristics can be improved. The results of this experiment can be seen in Figure 11, in which the lengths of the paths generated from our planning approach and from the sense all the time/everywhere approach are plotted as a scatter plot. The plot verifies empirically that our hypothesis is correct since the values are scattered near the diagonal (the correlation coefficient is $\rho = 0.99$) which indicates that the path lengths are similar.

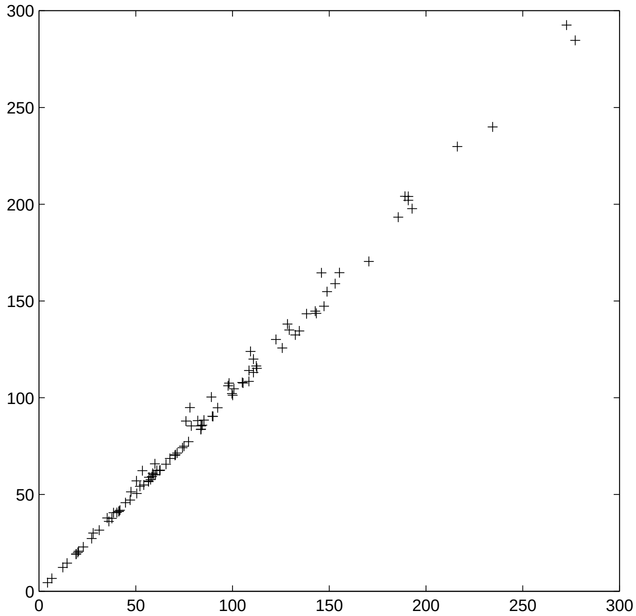

Fig. 11. The Mid-range planning sensing algorithm compared in a scatter plot to a hypothetical continuous sensing method with an unlimited range and 360° view range finder (no viewpoint planning). The horizontal axis is the path length from continuous sensing and planning, the vertical axis the Mid-range sensing/planning path length.

VII. Conclusion

We have presented our initial results on a combined sensing/planning approach that can alleviate the well know sensing horizon problem. Our method exploits the usage of mid-range sensing data by hallucinating most likely traversability maps. It also takes into account sensor planning to reduce the cost incurred with collecting mid-range data. We have shown that our method incurs only a small extra cost in open environments, but in more cluttered environments a great benefit can be achieved.

Current work involves primarily the quantitative characterization of the increase in performance in plan execution in typical missions. The objective is to use the sensing strategies developed under the Robotics CTA in order to demonstrate the combined planning and sensing system. Longer-term research involves more extensive tests in both simulation and on real outdoor autonomous systems. Negotiation between multiple vehicles for optimal sensing strategies is of particular interest. In that respect, the cost/benefit model described above fits well in the negotiation strategies based on economic models as described in [4] and [28].

References

[1] Motilal Agrawal and Kurt Konolige. Wide-baseline image matching for motion estimation. Submitted to IEEE International Conference on Image Processing, 2004.

[2] Simon Baker and Takeo Kanade. Hallucinating faces. In *Fourth International Conference on Automatic Face and Gesture Recognition*, March 2000.

[3] J. Besag. On the statistical analysis of dirty pictures. *Royal Statistical Soc.*, page 259:302, 1986.

[4] M. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *Proceedings of the 6 th International Conference on Intelligent Autonomous Systems, Venice, Italy*, July 2000.

[5] H.R Everett. *Sensors for Mobile Robots: Theory and Application*. A K Peters, Ltd, Wellesley, MA, 1995.

[6] D. Fox, J. Ko, K. Konolige, and B. Stewart. A hierarchical bayesian approach to the revisiting problem in mobile robot map building. In *International Symposium of Robotics Research*, 2003.

[7] H. González-Báños and J.C. Latombe. Planning robot motions for range-image acquisition and automatic 3d model construction. In *Integrated Planning for Autonomous Agent Architectures*, AAAI Fall Symposium Series, 1998.

[8] H. H. González-Baños and J. C. Latombe. Robot navigation for automatic model construction using safe regions. In *Proceedings of International Symposium on Experimental Robotics*, pages 405–415, 2000.

[9] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Royal Statistical Soc.*, page 271:279, 1989.

[10] Sanjiv Kumar. Models for learning spatial interactions in natural images. *The Robotics Institute, School of Computer Science, Carnegie Mellon University*, 2004.

[11] Kiriakos N. Kutulakos and Charles R. Dyer. Recovering shape by purposive viewpoint adjustment. In *Proceedings IEEE Conference Computer Vision and Pattern Recognition, CVPR*, Los Alamitos, California, 1992. IEEE Computer Society.

[12] S. L. Laubach. *Theory and Experiments in Autonomous Sensor-Based Motion Planning with Applications for Flight Planetary Microrovers*. PhD thesis, California Institute of Technology, California, CA, May 1999.

[13] S.L. Laubach and J. Burdick. An autonomous sensor-based path-planner for planetary microrovers. In *Proceedings of the IEEE Conference of Robotics and Autonomous*, 1999.

[14] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, Tokyo, 2001.

[15] Jasna Maver and Ruzena Bajcsy. Occlusions as a guide for planning the next view. In *IEEE Transactions Pattern Analysis Machine Intelligence*, volume 15(5), 1993.

[16] Stewart Moorehead. *Autonomous Surface Exploration for Mobile Robots*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2001.

[17] K. Murphy. Bayesian map learning in dynamic environments. In *Proceedings conf. on Neural Information Processing Systems NIPS*, 1999.

[18] B. Nabbe and M. Hebert. Toward practical cooperative stereo for robotic colonies. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3328–3335. Omnipress, May 2002.

[19] Bart Nabbe and Martial Hebert. Where and when to look. In *IROS2003*. IEEE, October 2003.

[20] David Nister. An efficient solution to the five-point relative pose problem. In *Conference on Computer Vision and Pattern Recognition*, June 2003.

[21] J. Rosenblatt. *DAMN: A Distributed Architecture for Mobile Navigation*. PhD thesis, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, 1997.

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

[23] C. Shoemaker and J. Bornstein. The demo iii ugv program: A testbed for autonomous navigation research. In *IEEE Int. Symp. Intelligent Control*, page 644:651, 1998.

[24] A. Stentz. *The NAVLAB System for Mobile Robot Navigation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, March 1990.

[25] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3310–3317, 1994.

[26] A. Stentz. The focussed d* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1652–1659, 1995.

[27] M.J. Swain and M.A. Stricker (editors). Promising directions in active vision. In *International Journal of Computer Vision*, volume 10, pages 109–126, 1993.

[28] S. Thayer, B. Digney, M. Dias, A. Stentz, B. Nabbe, and M. Hebert. Distributed robotic mapping of extreme environments. In *Proceedings of SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, November 2000.

[29] Nicolas Vandapel, Raghavendra Rao Donamukkala, and Martial Hebert. Experimental results in using aerial ladar data for mobile robot navigation. In *International Conference on Field and Service Robotics*, 2003.