Where and When to Look

How to extend the myopic planning horizon

Bart Nabbe and Martial Hebert

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{bana,hebert}@cs.cmu.edu

***Abstract*— In this paper we describe an approach towards integrating mid-range sensing data into a dynamic path planning algorithm.**[1]

The key problem, sensing for planning will be addressed in the context of outdoor navigation.

An algorithmic approach is described towards solving these problems and both simulation results and initial experimental results for outdoor navigation using wide baseline stereo data are presented.

I. Introduction

At the core of all autonomous robotics systems is a mobility system that takes data from sensors as input, reconstructs the 3-D geometry of the terrain around the vehicle, assesses the drivability of the terrain, detect obstacle regions, and modifies its currently planned path to avoid newly discovered non-drivable areas. The cycle is repeated many times a second 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 grid which is used for planning.

Irrespective of the implementation details of such mobility systems, their performance is always severely limited by the so-called myopic planning effect (Figure 1); 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.

For autonomous navigation over long distances, this issue 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 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.

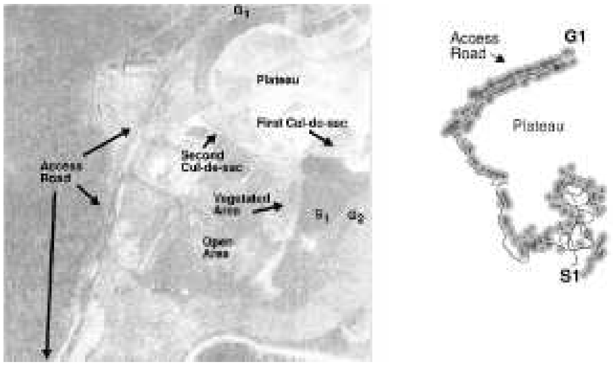

Fig. 1. Typical example of poor performance due to lack of sensor planning and mid-range sensing (Left: Overhead view of terrain; Right: Executed path with detected obstacles shown as shaded regions). The path from S1 to G1 intersects a large hill which is discovered only when the vehicle enters a large 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 [14])

A lot of attention has been given recently to the development of sensing solutions that would provide partial three-dimensional reconstruction of the vehicle's environment at longer range, e.g., up to 500m. These solutions are based either on reconstructing structure by integrating long sequences of monocular images acquired as the vehicle moves, 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. For example, in previous work [10], we have shown that a wide baseline stereo system is capable of acquiring range data up to 500m. In the rest of the paper we refer to such sensing approaches collectively as "mid-range sensing". In addition we refer to sensing systems for obstacle avoidance as "mobility sensing", these include sonars, small baseline stereo and laser range finders which will provide data up to a maximum of a few tenths of meters.

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

In principle, such sensors would provide enough data for a path planner to anticipate obstructions in the environment in order to make early decisions as to the optimal path to the goal. There are two critical issues in using the new, longer-range, sensors. First, it is obviously not practical to attempt to reconstruct a map of the terrain in all directions around the vehicle since it would require a prohibitive amount of computation and special hardware to achieve sensor coverage. Therefore, a strategy must be designed to decide *where to look* given the current state of the vehicle, its planned path, and the current representation of the world. 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 practical to attempt to continuously reconstruct the terrain structure at mid-range. Instead, we need to define a strategy that minimizes the extra cost incurred in sensing by dynamically deciding *when to look* based on the current state of the system.

In this work, we show how near-optimal strategies for when and where to look can be integrated with a state-of-the-art dynamic route planning system. We show how the performance of the system, measured in the length of the path executed between a start and a goal point, is greatly reduced at the expense of minimal extra computational cost.

II. Prior work

A large body of prior work exists in robotics in the general area of sensor planning. The first class of approaches is motivated by active vision problems in which a sensor is actively controlled by a robot. Most of the work in active vision concentrates around hand-eye coordination [17] or view planning for dense reconstruction [8][5]. All these techniques exploit the fact that the environment is constrained and that there is a specific object of interest.

The second class of approaches deals with exploration/coverage. The classical art gallery problem [11] falls in this category as well. There is an extensive body of work that deals with this problem [2], however the art gallery problem deals with planning views for complete coverage which brings a different solution to the table as required by our problem, view planning for navigation. The work of Moorehead [9] is also of the exploration type, and does also focus 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 presents in [3] and [4] a method that is based on detecting regions that are safe to navigate. These regions will be extended every iteration according to newly sensed data. On boundaries of these safe regions, candidate view points are generated, the viewpoint that exposes most of the unknown boundary is select for the next viewing position. Although the focus of the paper is on exploration, the viewing for planning approach is completely integrated. A major drawback of the algorithm as it is presented is that it has no error-recovery capabilities; once sensing data is received it is used as is and can not be revised. The sensing is also solely based on coverage and can not be tailored to other navigation tasks.

Much related to the work presented here is the work by Laubach; the framework she presents explicitly deals with sensing to discover better paths. In [6] and [7] she extends the TangentBug algorithm into the WedgeBug (RoverBug) algorithm which explicitly deals with a sensor model. She adds some virtual states to the typical bug states (Motion towards 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 clean in its underlying concept, however it still suffers from the same problems as the Safe region algorithm, that there is 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 traversing cost is limited to a binary representation, traversable or non-traversable, which can not express costs of traversing for example vegetated terrain.

III. Planning with Sensing

To alleviate the limitations of WedgeBug, we propose a solution with a sensor (grid) based planner. The path planner that we use for this is the D* planner [15], [16]. The D* planner takes as input a collection of cost maps that describe 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. The "where and when to look" questions are addressed automatically by the planner, which attempts to find a path that still avoids obstacles but also maximizes the utility of sensing.

This approach of incorporating utility into a cost function is similar to the approach Rosenblatt [12], [13] takes in his arbiter. However, his utility map does only look at a few possible actions over a very limited horizon and does not deal with additional utility for sensing purposes.

In our approach this cost function consists of three individual costs (utilities). That express the current state of the world C_{mob}, the added utility for sensing purposes $C_{bennefit}$ and a heuristic cost C_{mid}.

C_{mob} 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.

$C_{benefit}$ is the (inverse) benefit of visiting the 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. The computation of $C_{benefit}$ is at the heart of deciding “where to look”.

C_{mid} expresses our knowledge about the traversability of a grid cell as discovered by the mid-range sensing system. This cost is similar to C_{mob} in that it encodes the local obstacle-ness of the terrain. The difference is that it is computed from the mid-range sensors instead of from the mobility sensors. It is useful to separate the two costs because, typically, we have less confidence in the mid-range sensors.

The combination of these costs captures the notion of utility for sensing purposes and will also make cells that are expected to be traversable based on our mid-range data more favorable: In the planner, the cost are combined as

$$C = C_{mob} + \epsilon C_{benefit} + \tau C_{mid}$$

where the parameter ϵ controls the amount of deviation from the path that is allowed for acquiring sensor data. If $\epsilon = 0$, no sensor planning takes place and the vehicle executes the default path computed from the traversability maps alone. τ captures the level of confidence in the mid-range data. If $\tau = 0$, the mid-range data is ignored and the plan is executed based only on the mobility sensors.

The computation of the combined utility value needs not be restricted to this simple linear combination, as long as this function is admissible. For example a non-linear function that thresholds if costs are too high is required in order to guarantee that the planner will not plan any paths trough non-traversable areas if there exists no path.

IV. Sensing for planning

Given the above cost definition, for each cell of the map, we need to evaluate the gain obtained by taking a sensor reading from that cell position. Consider a position (x, y) in our map, at this position, we can point our sensor at an angle θ. Given the field of view and minimum and maximum sensing distance of the sensor, we can define $\mathcal{G}$ as the set of all possible labelings for these cells that fall within the sensor footprint. For each labelling of traversable/non-traversable cells (G_j) there is a probability $P(G_j)$ that this labelling occurs. Given a labelling, one can compute the cost of getting to the goal by not making the observation at (x, y): C_{ns_j}, and the cost of getting to the goal making the observation: C_{s_j} (Figure 2). Intuitively, C_{ns_j} is the cost of the path executed if no sensors other than the short-range mobility sensors are used and the map is in configuration (G_j). The utility for each position in the map $E_{(x,y)}$ can now be expressed as:

$$E_{(x,y)} = \underset{\theta}{\operatorname{argmax}} \left(\sum_j \frac{P(G_j)(C_{ns_j} - C_{s_j})}{C_{ns_j}} \right) \tag{1}$$

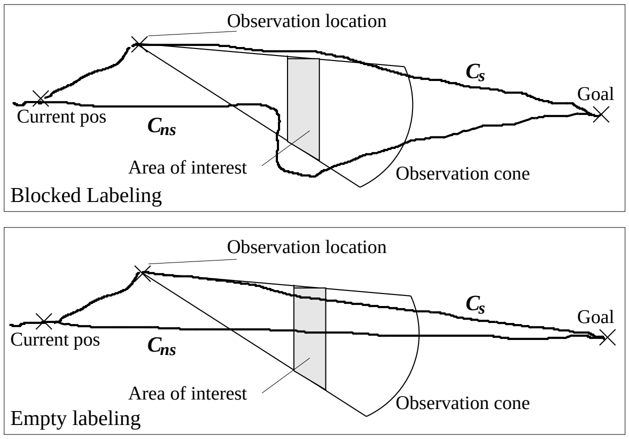

Fig. 2. Illustration of the path computed with (C_s) and without sensing (C_{ns}) in two different configurations of the world. One configuration is obtained by filling up the world with obstacles in the field of view of the sensor; the other one assumes no obstacles at all. In principle, the utility of sensing is evaluated over all possible configuration of the map in between these two extremes.

In principle, the utility can be computed everywhere in the map and the resulting utility map (see also Figure 4) can be used by the planner to plan for the most favorable sensing position. C_s and C_{ns} can be evaluated by running the planner on different “virtual” configuration of the world map corresponding to different configurations G_j. In reality, however, this would require the enumeration of *all* possible configurations of the world in the sensor footprint, 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 heuristics must be used to reduce the computation while retaining near-optimal evaluation of the utility.

Specifically, we propose to reduce the computation in Equation 1 in two ways:

- Restrict search over θ: Given a current cost map configuration and current proposed path computed from the map, we propose to process the map in order to identify those locations those locations that are of most immediate interest for mid-range sensing. Given a candidate sensor position, we consider only the θ that would aim the sensor in the direction of the high-interest locations, thus eliminating the need to search over sensor orientations.

- Restrict possible configurations: We consider only the extreme configurations of G in which either no new obstacles are added or, conversely, the entire area is blocked, thus avoiding searching through a combinatorially large set of possible configurations. In the future, intermediate configurations may be considered, but this first cut provides us with a baseline on the performance of our sensor-based planning strategy.

We now describe in detail how these two heuristic reductions are implemented.

The key in eliminating the combinatorial nature of the summation is to avoid summing over all possible configurations of the world. To do this, we first compute an intermediate map I such that $I(x', y')$ presents the utility in capturing data in the vicinity of (x', y'). Then for each potential observation point (x_o, y_o) the result from computing I is used to find the utility $E(x_o, y_o)$ of taking a measurement from (x_o, y_o).

We define an additional utility value I for each cell (x', y'), which represents the interest in observing the area A in the vicinity of this location. A is a fixed-size region of support for which $I(x', y')$ is computed. We will use this I to compute a single viewing angle θ, so we can eliminate the need to evaluate E for all possible viewing angles. For each $(x', y') \in A$ we define the probability $P_{occupied_{x',y'}}$ of the cell being occupied, given the current knowledge of the world. $P_{occupied_{x',y'}}$ is computed such that the closer (x', y') is to a cell with high cost (C_{mob} or C_{mid}), the higher $P_{occupied_{x',y'}}$ is. This heuristic of course cannot anticipate obstacles in large empty regions but it does guarantee that the algorithm focuses on those cluttered areas in which failure to detect an obstruction in time may lead to a large detour. This is widely adapted technique for obstacle expansion in current path planners [14].

The second heuristics limits the area of the world in which the sum is evaluated to those areas that are of most immediate interest, namely the areas close to the current path, D_{toPath}. This is implemented by an iterative procedure, similar to the distance transform in potential fields, that expands and weighs the current path. This heuristic has the effect of giving less weight to those cells that are far from the path, which may not have immediate effect on the path. Finally, cells that have already been observed with high certainty are not taken into account in the summation under the assumption that a new sensor reading would not change their status. The observed cells are marked by a flag $Observed(x', y')$. Combining those heuristics lead to the modified utility function at cell (x, y):

$$I(x,y) = \sum_{(x,y)\in A} \Big[D_{toPath}\Big(1 - Observed_{(x',y')}\Big) \Big(1 - \Big|2\Big(P_{occupied_{x',y'}} - \frac{1}{2}\Big)\Big|\Big)\Big]$$

Finally, since we are more interested in detecting obstacles that are imminent, we define the maximum of I within minimum and maximum mid-range sensing range and closest to the current robot position as (x_{poi}, y_{poi}). Given a potential observation point (x_o, y_o) for which we want to evaluate $E(x_o, y_o)$, $\theta_{(x_o,y_o)}$ can then be computed as the angle for the sensor that will bring (x_{poi}, y_{poi}) at the center of the field of view (Figure 3,4).

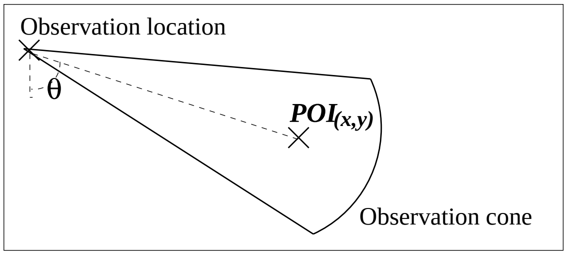

Fig. 3. The relation between θ and (x_{poi}, y_{poi})

Currently we also limit the evaluation of E to only the two extreme cases of all possible configurations in G: the open configuration G_{open} and the closed configuration G_{closed} and there corresponding probabilities of occurrence: $P(G_{closed}), P(G_{open})$. The sum in E reduces now to only computing the costs C_{ns_j} and C_{s_j} for these two configurations, which leads to the following reduced expression for E.

$$E_{(x,y)} = \left(\frac{P(G_{closed})(C_{ns_{closed}} - C_{s_{closed}})}{C_{ns_{closed}}}\right) + \left(\frac{P(G_{open})(C_{ns_{open}} - C_{s_{open}})}{C_{ns_{open}}}\right)$$

The computation of the utility value at each cell addresses the problem of where to look: The cells with the highest utilities are favored by the planner as sensing locations. Since it necessary to limit the number of observation location ("when to look"), only those locations that have sufficiently high utility are retained.

Figure 4 shows some typical utility maps for two different robot positions (A and B, marked by an x). This figure shows clearly that when the robot reaches position B, it has discovered a corridor through which it is anticipating to traverse and therefore decides to gather more information to verify this hypothesis.

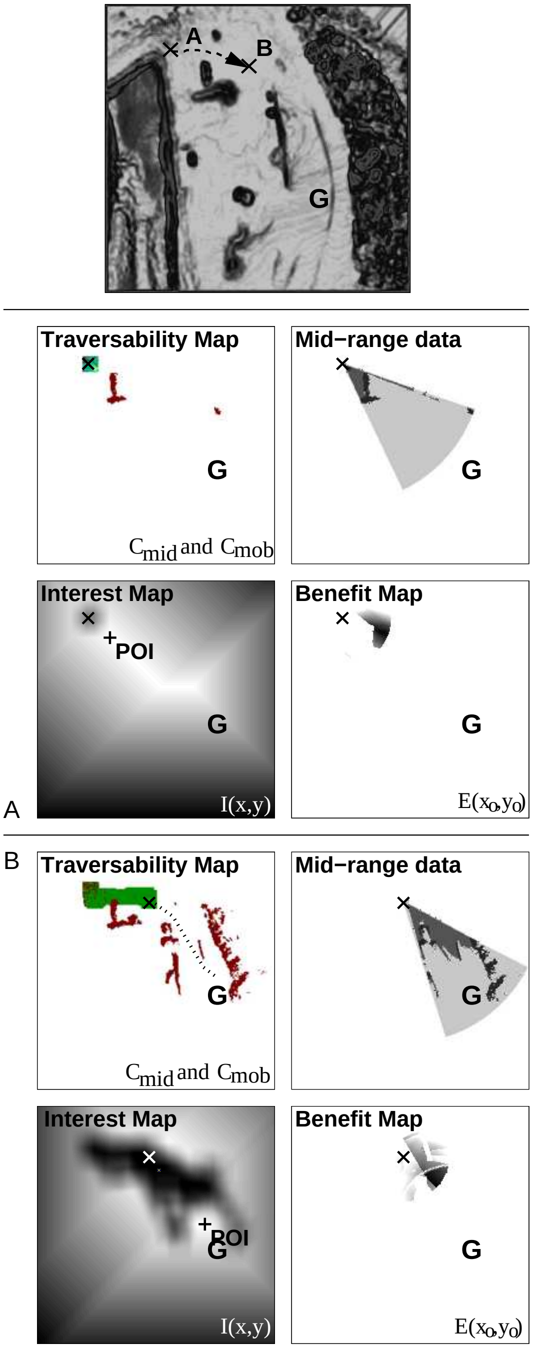

Fig. 4. The associated Cost/Benefit maps displayed for the robot positions A and B (marked with a X) along the way towards goal G. In the Interest map, a high measure of interest is represented with a bright color. Whereas in the Benefit map, a high benefit measure is represented by a dark color. The darkest point in the Benefit map is the one used as the next sensing position.

V. Discussion

In the discussion of our results, the following settings of parameters have been used: FOV $45°$, $\epsilon = 0.25$, $\tau = 0.25$ and both "Priors", $P_{closed} = P_{open} = 0.5$.

We conducted an example experiment in an outdoor environment in which we, as before, 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 6. Along this path we have collected a few pair of stereo images, from which a sparse set of 3D points were computed with a wide baseline stereo algorithm [19][10]. 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. Figure 5 shows a typical stereo image pair that was used to create this data. The baseline of the system was typically 3 to 4 meters, which yielded range data up to 30 meters. 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 6. The path as found by the mid-range sensing approach had a total length of 65 meters and is 57% shorter than the path generated by mobility sensing only.

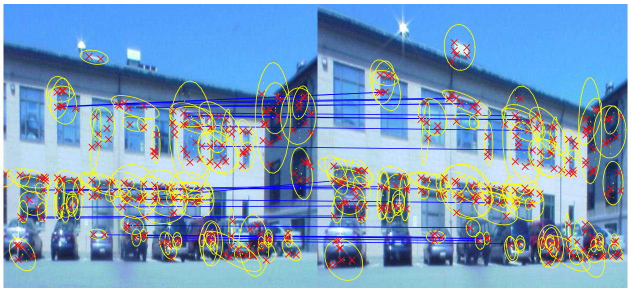

Fig. 5. The range data is computed with a wide-baseline stereo system, using a region based approach.

Further experiments were conducted using the environment of Figure 7, an open mine. For controlled experiment, we ran simulations of the system for different start and goal locations and different system configurations. A complete elevation map of the environment is used for generating sensor data in the simulation, but it is not known by the robot initially. Figure 8 shows one example of paths executed using the strategy described above.

The path executed by using only the mobility sensors shows that the vehicle squirts the terrain obstruction instead of completely avoiding it. When sensor planning and mid-range sensing is added, the vehicle avoids the obstructions long before it reaches them, yielding a path that is both substantially shorter but also involves fewer changes of directions, thus allowing greater execution

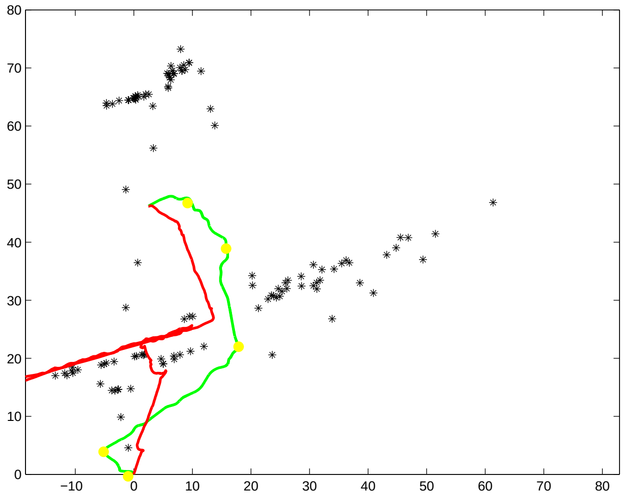

Fig. 6. a birds-eye view of the real test scenario, in which the mobility sensing navigation skirts around the building (dark) and the Mid-range sensing/planing algorithm (light) finds a passage and heads straight towards it. The dots (·) indicate positions were the robot captured mid-range sensor data.

Fig. 7. Typical example of real terrain.

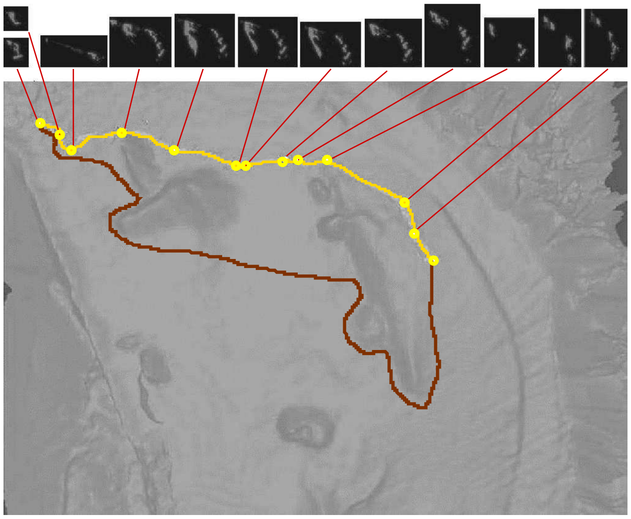

Fig. 8. Difference between paths executed without (dark curve) and with (light curve) sensor planning. The locations at which new sensor data was acquired are shown on the corresponding path along with the world reconstructed from the data sensed from those locations.

speed.

We have repeated this experiment for 100 more trial runs with random chosen start and goal positions; within the constraint that there exists a path from start to goal.

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 9, in which the runs are sorted in order of increasing gain for the sensor-based planning method. This first type of analysis is necessary 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 4%. However, more interesting is that our results show that 39% of the runs exhibit positive gain (up to 73%). And that of those that have negative gain, the maximum loss is only -10%, furthermore, as the left illustration in the figure shows, these cases are those for which the paths in the environment are unobstructed.

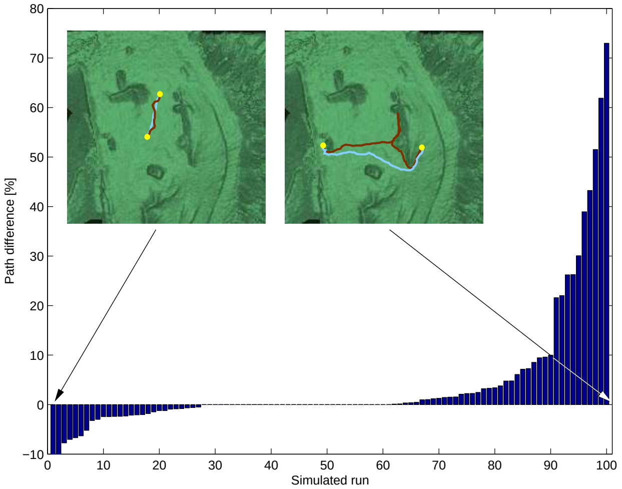

Fig. 9. The Mid-range planning sensing algorithm compared to the standard navigation approach. A positive percentage shows how much shorter the path from the mid-range sense/planning is over the mobility only approach. The average gain in path length is 4%. However, more interesting is that 39% of the runs exhibit positive gain (up to 73%) and that of those that have negative gain, the maximum loss is -10%. This is due to the overhead introduced by the explore behavior that tries to find better paths. For 34% of the runs there was no gain or loss.

Second, it is important to 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, 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 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. This part of the analysis is summarize in Figure 10, 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 indicate that the lengths are similar, i.e., the values are scattered near the diagonal (the correlation coefficient is $\rho = 0.99$). This result verifies 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.

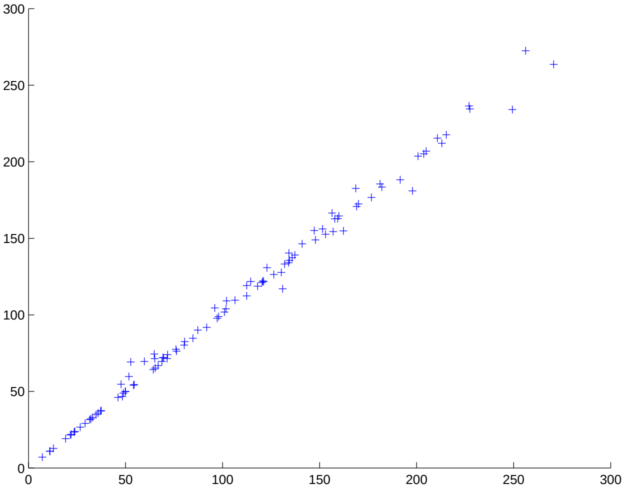

Fig. 10. The Mid-range planning sensing algorithm compared in a scatter plot to a hypothetical continuous sensing method with an unlimited range and 360° view range finder (no viewpoint planning). The horizontal axis is the path length from continuous sensing and planning, the vertical axis the Mid-range sensing/planning path length.

Finally, a third type of analysis uses the paths generated by an omniscient planner that 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 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 limited sensor horizon. This analysis is shown in Figure 11, in which the paths generated by using our mid-sensor planning strategy and the paths executed by using mobility sensing only are plotted again as a scatter plot. The graph shows that the majority of the paths are longer when using mobility sensing alone, as was already clear from Figure 9, but it also shows that the paths generated with mid-range sensor planning are closer to the optimal from the omniscient planner, i.e., the points corresponding to the paths computed with sensor planning are closer to the diagonal in the graph of Figure 11. Specifically, the correlation coefficient is $\rho = 0.81$ for the paths planned with mobility sensing alone, and $\rho = 0.98$ for the paths planned with mid-range sensor planning.

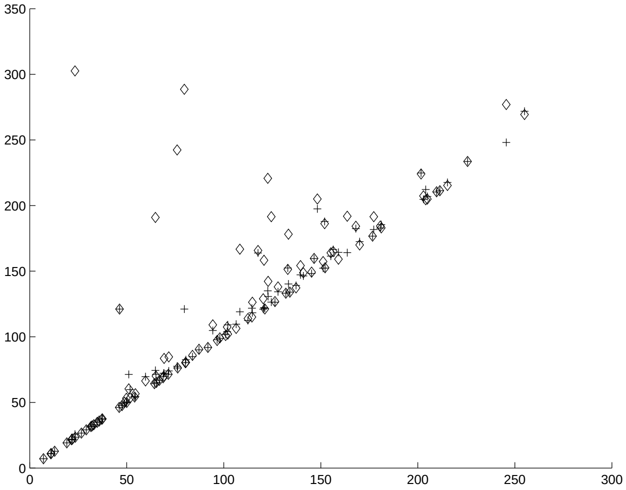

Fig. 11. A scatter plot comparing the Mobility sensing method (◇) to the true shortest path and the Mid-range sensing/planning algorithm (+) compared to the true shortest path. The horizontal axis is the path length from the omniscient planner, the vertical axis is the path length from our Mid-range sensing/planning method.

VI. Conclusion

In this article, we have presented our initial results on a combined sensing/planning approach that can alleviate the well know sensing horizon problem. Our method will be beneficial especially for outdoor environments in which it is expensive to collect sensing data. We have shown that while incurring a small extra cost in open environments, a great benefit can be achieved in more cluttered environments.

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 elaborate reasoning and sensing. Negotiation between multiple vehicles for optimal sensing strategies in 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 [1] and [18].

References

[1] 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.

[2] H. Gonz´alez-B´años 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.

[3] H. H. Gonz´alez-Ba¯nos 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.

[4] H. H. Gonz´alez-Ba¯nos and J. C. Latombe. Navigation strategies for exploring indoor environments. Submitted to the International Journal of Robotics Research, 2001.

[5] Kiriakos N. Kutulakos and Charles R. Dyer. Recovering shape by purposive viewpoint adjustment. In *Proceedings IEEE Conference Computer Vision and Pattern Recognition, CVPR*, pages 16–22, Los Alamitos, California, 15–18 1992. IEEE Computer Society.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] J. O Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, 1987.

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

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

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

[15] 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.

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

[17] 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.

[18] 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.

[19] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local affinely invariant regions. In *Proceedings British Machine Vision Conference*, 2000.