Minimum Distance Localization for a Robot with Limited Visibility*

Malvika Rao, Gregory Dudek and Sue Whitesides
Centre for Intelligent Machines
McGill University
3480 University St, Montréal, Québec, Canada H3A 2A7
{*rao,dudek,sue*}@*cs.mcgill.ca*

***Abstract*— Minimum distance localization is the problem of finding the shortest possible path for a robot to eliminate ambiguity regarding its position in the environment. We consider the problem of minimum distance localization in self-similar environments, where the robot's sensor has limited visibility, and describe two randomized algorithms that solve the problem. Our algorithms reduce the risk of requiring impractical observations and solve the problem without excessive computation. Our results are validated using numerical simulations.**

I. Introduction

In this paper we consider the problem of global localization, that is localization of a robot without an initial estimate of its position. In general, localization of a mobile robot may require motion to new vantage points to resolve ambiguity about the robot's position. This is true for most sensors and, in particular, for range sensors, even idealized ones, which is the case we consider here. The fundamental cause of such ambiguity is that some regions of the environment may be locally indistinguishable. Even with a perfect map, a perfect compass, and a perfect range sensor in a 2-dimensional planar world, the problem of selecting an optimum path for localization is difficult. Computing a good approximation to the optimal localizing path is tractable, but nevertheless computationally complex.

Localization is one of the most fundamental problems in mobile robotics and is the essence of more elaborate procedures such as mapping, simultaneous localization and mapping, and task execution.

The existing results regarding the complexity of global localization, and most other results regarding the complexity of related problems, deal with the case of ideal range sensors with infinite range. Likewise, existing algorithms that explicitly solve for, or approximate, the ideal path also deal with sensors with no limits on their maximum range.

In our previous work [7] we introduced two randomized approximation algorithms that solve minimum distance localization, namely, the *common overlay localization (COL)* algorithm and the *useful region localization (UAL)*[1] algorithm. In this paper, we extend these algorithms to consider the feasibility of minimum distance global localization given a range sensor with *bounded* maximum range. This problem is fundamentally more difficult than the traditional infinite-range problem due to the reduced information available to the robot and hence the much greater potential for ambiguity. Our approach to the problem is stochastic in nature, and demonstrates that good approximations to an optimal path can be feasibly computed.

A. Outline

In the next section, we briefly discuss related work. Section 3 provides a formal description of the problem and states the assumptions underlying our approach. In Section 4 we review our two algorithms, COL and UAL, and explain how we incorporated the constraint of limited sensor visibility into them. Experimental results from simulation are presented in Section 5. Finally, we close with a discussion of open problems, directions for future work and conclusions from our results.

II. Related Work

Dudek, Romanik, and Whitesides [2] show that minimum distance localization is NP-hard and propose an approximation algorithm that assumes perfect sensor visibility. One implication of the perfect idealized sensor model is that the problem will only get more difficult with more realistic, error-prone sensors. That work exploits a result by Guibas, Motwani, and Raghavan [4] who present a technique for preprocessing a map polygon P so that given a robot's visibility polygon V the set of points in P whose visibility polygon is congruent under translation to V is returned. Here, too, perfect sensor visibility is assumed. Schuierer [6] proposes a technique that uses geometric overlay trees to speed up the implementation of the greedy localization strategy put forth by [2]. While his approach reduces the time and space complexity, no implementation results are given, and it is unclear how to extend the technique to address more practical issues such as limited sensor visibility. Kleinberg [5] approaches robot localization by modeling the environment as a bounded geometric tree.

Brown and Donald [1] describe algorithms for robot localization that allow for uncertainty in the data returned

*Research supported in part by NSERC and FCAR.

[1] Acronym has been changed to UAL to avoid confusion with web url.

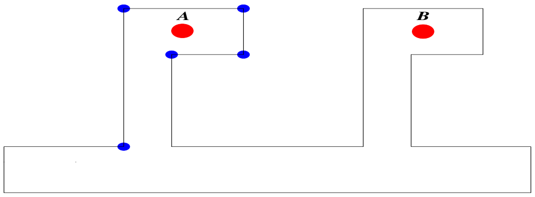


Fig. 1. Visible vertices from location A. The visibility data at locations A and B are the same.

by the range sensor. Fox, Burgard, and Thrun [3] use Markov localization to determine the position of a mobile robot from sensor data. In that work, global localization is achieved, but the length of the localizing trajectory relative to the optimum is not considered. Several authors have considered action and localization in a compound setting where the effects of actions can be accounted for probabilistically. In particular, techniques such as Markov localization and its derivatives are able to predict the belief states that may ensue from various actions [3]. Long-range path planning using such methods remains problematic, however, due to the large state spaces involved. Moreover, intractability of optimum length decision planning has already been well established.

In [7] we describe the COL and UAL algorithms for the case of unlimited sensor visibility and determine their complexity. We motivate our strategy of randomized sampling and present experimental results that show that our algorithms are effective for the ensemble of environments evaluated. The performance of our algorithms is demonstrated to improve rapidly with the number of random sample points used: typically, a limited number of samples is sufficient to obtain a near-optimal length localization trajectory.

III. Problem Specification

In this section we formally define the localization problem and state our assumptions about the robot and its environment. We are given a random environment modeled by an n-vertex simple polygon P without holes positioned somewhere in the $2D$ plane. A mobile robot is placed at an unknown initial location within P, for which it has a map. First, the robot must determine if its initial location is unique by sensing its surroundings and matching the resulting visibility data W to the map of the environment. Given P and W, the robot must generate the set H of all hypothetical locations $p_i \in P$ such that the visibility at p_i is congruent under translation to W. Next, the robot must determine its true initial location by sensing and traveling in order to eliminate all hypothetical locations but one from H, while minimizing the distance traveled.

The robot is assumed to be a point robot moving in this static, $2D$, obstacle-free environment. The robot is able to make error-free motions between arbitrary locations in the environment. The movement of the robot is restricted to the inside and along the boundary of the environment. As well, it is able to determine its orientation; otherwise, it would be impossible for the robot to uniquely determine its exact location in an environment with non-trivial symmetry such as a square [2].

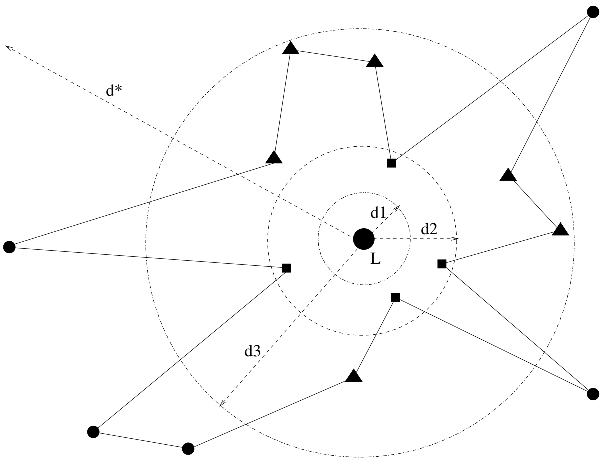


Fig. 2. What the robot sees under a range of visibilities.

The robot's sensor behaves using a "perfect" sensor in that it can precisely detect distances to those points on the boundary of the environment for which the robot has an unobstructed line of sight. However, the distance the sensor can "see" can be bounded by a constant Z. Consequently, the sensor can measure distances to those points on the boundary of the environment for which the robot has an unobstructed line of sight and which lie within a distance Z.

The visibility data W sensed by the robot is composed of the counter-clockwise ordering of vertices and edges seen by the robot (see Figure 1). Geometric relationships among the data such as vertex angles, distances, adjacencies, and the robot's relative position with respect to the data sensed are available. W can also be regarded as a visibility polygon. We assume non-null input visibility data W.

When the sensor visibility is limited, the robot can only observe a subset of the set of vertices and edges comprising its environment that it would otherwise see with unlimited visibility. Let S_i denote the set of vertices seen by the robot at visibility d_i. In general, for visibility values $d_1 < d_2 < d_3 < \cdots < d^*$, where d^* represents unlimited or infinite visibility, the corresponding sets of vertices seen by the robot satisfy the following relationships: $S_1 \subset S_2 \subset S_3 \subset \cdots \subset S^*$. Consider Figure 2. The robot, labeled L, is situated at the center of the polygonal environment. With visibility d_1 the robot sees none of the vertices of the polygonal environment. With visibility d_2 the robot sees only those vertices represented by squares. With visibility d_3 the robot sees the vertices represented by squares as well

[2] Note: the robot could still solve the problem up to rotational symmetry.

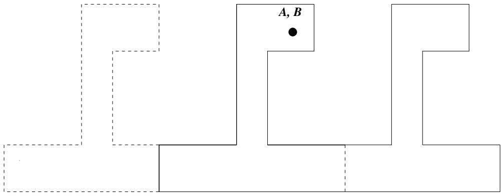


Fig. 3. Overlay arrangement for the situation shown in Figure 1.

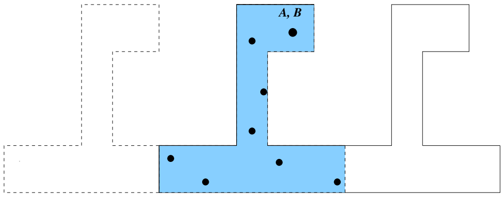


Fig. 4. Shaded area represents the overlay intersection region OI. Random points are chosen within this region.

as those represented by triangles. With unlimited visibility d^* the robot is able to see all the vertices of the polygon but not beyond since it cannot see through walls or solid edges.

IV. Approach

The original COL and UAL algorithms assume perfect sensor visibility. We now incorporate the constraint of limited sensor visibility into these algorithms. UAL is a variation of COL that is more judicious in selecting where to make observations. Before we proceed to describe the steps taken to address localization under limited visibility, we will first briefly describe some key definitions and procedures common to both the COL and UAL algorithms (see [7] for more details).

Our localization algorithms comprise two phases: *hypothesis generation* and *hypothesis elimination*. Given an environment P and visibility data sensed by the robot W, the hypothesis generation phase determines the set H of all hypothetical locations $p_1, p_2, \ldots, p_k \subset P$ such that the visibility data $Vis(p_i)$ computed at p_i matches W for $i = 1, \ldots, k$ (see Figure 1). The hypothesis elimination phase begins by computing an overlay arrangement centered around a reference point or origin (see Figure 3). We consider only the *connected component of the intersection region of the overlay arrangement that contains the origin* since it is the area known to exist in all hypotheses (see Figure 4). We will refer to this connected overlay intersection component containing the origin as OI.

A. Hypothesis Elimination

A set R of *random points* is chosen by randomly sampling the interior of OI according to a uniform distribution (see Figure 4). R is then evaluated to see if any of the random points contained in R prove *useful* - i.e. if sensing at this location is guaranteed to yield new information that distinguishes among the different hypothetical locations. For each random point picked, $r \in R$, a value function $Value(r) = info/distance_{OI}$ is computed, where $info$ is the expected number of hypotheses that could be eliminated at r, assuming all the hypothesized initial locations are equally likely, and $distance_{OI}$ is the shortest path trajectory, constrained to lie within OI, from the robot's initial location at the origin of the overlay to r.

We calculate $info$ for a point r as follows. We assume that all hypotheses are equally likely. We say two hypotheses h_i and h_j are equivalent at r if $Vis(h_i, r)$ is congruent to $Vis(h_j, r)$ and has the same orientation. $Vis(h_i, r)$ refers to the visibility data computed at a point z such that the relative position of z with respect to the hypothetical location p_i is equivalent to the relative position of r with respect to the overlay origin. If there exist b equivalence classes of hypotheses at r of sizes $s_1, s_2, \ldots, s_b$ respectively, where the total number of hypotheses $k = s_1 + s_2 + \cdots + s_b$, then

$info(r) = (s_1/k)(k - s_1) + (s_2/k)(k - s_2) + \cdots + (s_b/k)(k - s_b)$

The robot is moved to the random point r' in OI with the highest non-zero value of $Value(r')$. Those hypotheses h_i where $Vis(h_i, r')$ does not match the visibility data sensed by the robot at its new location are ruled out.

B. Limited Visibility

All visibility computations and comparisons are restricted by the distance limit d beyond which the robot is assumed to not see. When computing the initial visibility data W sensed by the robot, those vertices and edges lying outside the circular area of radius d centered at the robot's initial location are not included. Likewise, in the hypothesis generation phase, we only look for patterns of vertices that conform to W as calculated incorporating the distance limit. During hypothesis elimination, the robot's visibility at a particular random point is calculated in the same way as W. In addition, the distance limit is taken into account when we determine which random points provide non-zero information.

C. Common Overlay Localization Algorithm

We now present how we incorporated the limited sensor visibility constraint into our COL algorithm. Given an input polygonal environment P and a robot placed at an unknown initial location in P, the COL algorithm is as follows.

1) Sense visibility data W from the robot's current unknown initial location. Only vertices and edges lying within the circular area of radius d centered at the robot's initial location are included.
2) Generate the set of all hypothetical locations H in the environment P that match the visibility data sensed W. The matching pattern of vertices and edges seen from each hypothetical location must lie within the circular area of radius d centered at the hypothetical location.

3) Choose an arbitrary hypothetical location in H as the origin.
4) Construct an overlay arrangement centered on the origin.
5) Compute the connected overlay intersection component containing the origin, OI.
6) Randomly choose a predetermined number of locations or points within OI.
7) For each random point picked, r, compute the value function $Value(r) = info/distance_{OI}$. When computing $info$ for a random point the visibility range d is taken into account. Only points that can see landmarks within a radius d are assigned non-zero $info$.
8) Observe that at each overlay intersection, there is latent information to be gained that is guaranteed to eliminate some hypotheses. Therefore, if none of the random points yield non-zero $info$, then the number of random points required is increased and chosen all over again within the current overlay intersection area OI. Steps 6, 7 and 8 are repeated for a predetermined number of trials [3].
9) Move the robot to the random point r' in the overlay with the highest non-zero value of $Value(r')$.
10) Now, eliminate hypotheses by comparing visibility data sensed by the robot at r' with the visibility data computed at all the equivalent random points corresponding to all the active hypotheses. Visibility data must be computed taking into account the visibility range d as described above.
11) Call the set of eliminated hypotheses E, and compute the overlay arrangement with the reduced set of hypotheses $H - E$. Repeat Steps 3-10 until only one hypothesis, corresponding to the true initial location of the robot, is left in $H - E$.

D. Useful Region Localization Algorithm

The UAL algorithm differs from the COL algorithm with respect to the region where random points are chosen. We determine precisely the portions of OI where any random point chosen is guaranteed to provide new information.

An *internal edge* of an overlay intersection area OI is defined as an edge (one of many) that separates the inside of OI from other parts of the overlay arrangement, as opposed to those edges of OI that pertain to the outer silhouette of the overlay arrangement which separates the inside of OI from the rest of the 2D plane (see Figure 5). Given a polygon P the *weak visibility polygon* $W(e)$ of an edge $e \in P$ is defined as the set of all points $y \in P$ that are visible from some point on e (see Figure 6). Once we have determined the set of internal edges of OI, the useful portions U of OI can be computed by taking the union of all the weak visibility polygons of all the internal edges of OI.

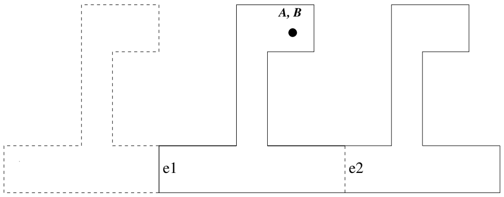


Fig. 5. e1 and e2 are internal edges.

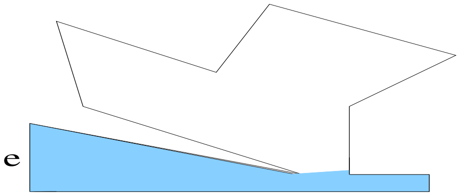


Fig. 6. Shaded region represents the weak visibility polygon of edge e.

Figure 7 depicts the useful region of polygon P. Steps 1-5 of the UAL algorithm remain the same as in the COL algorithm. Following these steps, the UAL algorithm incorporating the constraint of limited sensor visibility is as follows.

6) Compute the useful region U of OI. The weak visibility polygons comprising U must be clipped by a factor d. This is accomplished by aligning an infinitely long rectangle along each internal edge, where the width of the rectangle (the axis aligned to the width is perpendicular to the internal edge) is equivalent to the distance limit d. The intersection of the rectangle with the original weak visibility polygon produces the clipped polygon.
7) Randomly choose a predetermined number of locations or points within U (see Figure 7).
8) For each random point picked, r, compute the value function $Value(r) = info/distance_{OI}$. When computing $info$ for a random point the visibility range d is taken into account. Only points that can see landmarks within a radius d are assigned non-zero $info$.
9) Move the robot to the random point r' in the overlay with the highest non-zero value of $Value(r')$. Note that we are guaranteed that all the random points chosen provide non-zero information for hypothesis elimination. As a result, we do not need to choose more random points repeatedly as is done in the COL algorithm.

Finally, steps 10 and 11 of the UAL algorithm also remain the same as those in the COL algorithm.

[3] In our implementation, we terminate the algorithm if no useful points are obtained after this predetermined number of trials. Hence we proceed to the next step only if useful points exist.

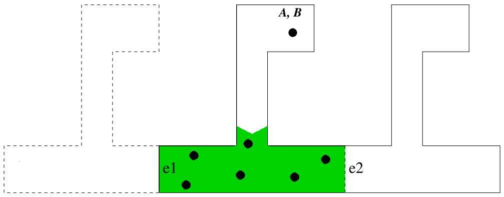


Fig. 7. Shaded area represents the useful region for the case of unlimited visibility. Random points are chosen within this region.

V. Results

A. Performance of Common Overlay Localization Algorithm

We performed the following set of experiments to measure the effect of limited sensor visibility on the localization path length for *two weighted* decision strategies. The first decision strategy is the strategy that we have described in the previous section. Here, the information gain of each random point is weighted by its distance from the robot's initial location. As before, we will continue to refer to this strategy as the weighted strategy, or the *standard weighted* strategy. The second decision strategy, which we will refer to as the *hybrid* strategy, operates by weighting the information gain at each random point by its distance from the robot's *current* location.

Twenty simulated office environments of average number of vertices approximately 400 were generated for this purpose. For each of the 20 environments, three initial robot locations were randomly selected. Each quantity of random points was repeated twice to balance out any abnormal distributions. The total number of experimental trials was, therefore, $20 \times 3 \times 2 = 120$ trials [4]. Note that the size of each grid cell in the simulated office environments we used is 100 pixel units.

Figure 8 depicts the average path length obtained for both the standard weighted strategy as well as the hybrid strategy with unlimited visibility. While both strategies exhibit an improvement in performance as the number of random points is increased, the hybrid strategy produces significantly shorter path lengths.

Figures 9, 10, and 11 show the path lengths obtained for both strategies with visibility ranges 500, 200, and 90 pixel units, respectively. For the limited sensor visibility value of 500, the path length for the standard weighted strategy gets shorter as the number of points is increased. Somewhat curious, however, is the fact that for visibility values 200 and below the path length for the standard weighted strategy appears to be more or less the same, even as the number of random points are increased. We will explain this behavior presently.

Figures 12 and 13 plot the average path length produced by the two strategies with decreasing visibility range and number of random points 40 and 250, respectively. As expected, the results indicate that as the visibility gets poorer the path lengths for effective localization get longer. The poorer the robot's vision, the closer it has to get to a distinguishing feature in order to eliminate hypotheses. As a result, it ends up traveling longer trajectories.

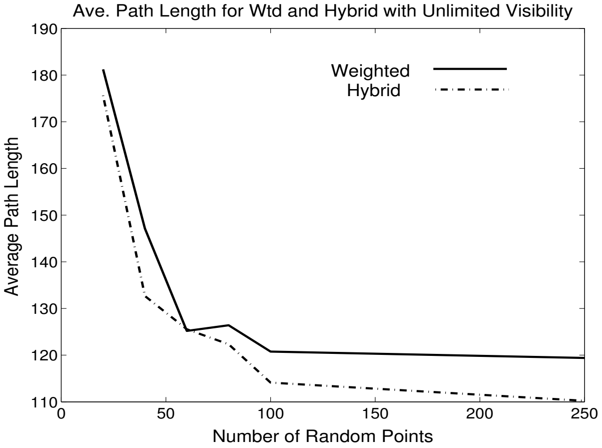


Fig. 8. Weighted versus hybrid strategies for COL with unlimited visibility. 120 trials were performed.

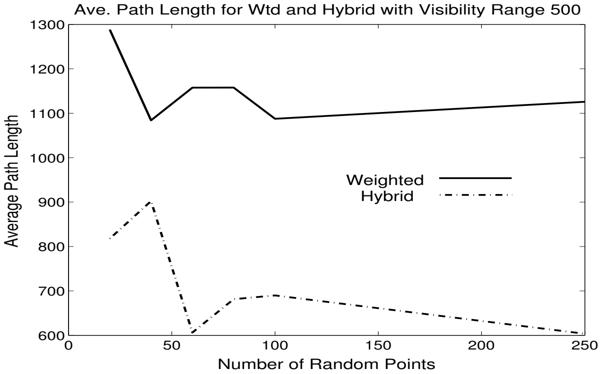


Fig. 9. Weighted versus hybrid strategies for COL with visibility range 500. 120 trials were performed.

In terms of minimizing path length, the hybrid strategy soundly outperforms the standard weighted strategy for all the values of visibility and the different numbers of random points. In fact, the difference in path length between the two decision strategies seems to get larger as the visibility gets poorer. Since the overlay intersection gets increasingly larger as hypotheses are ruled out, it follows that the robot moves progressively further away from its initial location each time it makes a decision to eliminate hypotheses. If the distance of a potential destination point is always measured from the robot's initial location, then it is possible that this could lead to extreme "zigzagging" of the robot's trajectory, depending on the shape of the environment and the placement of the initial location within the environment

[4] This set of trials will be used repeatedly for the experiments described in the remainder of this paper.

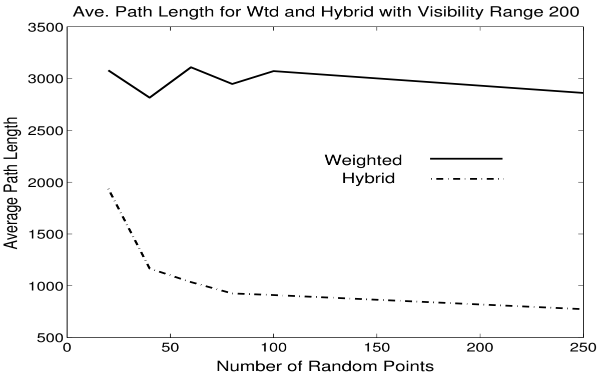


Fig. 10. Weighted versus hybrid strategies for COL with visibility range 200. 120 trials were performed.

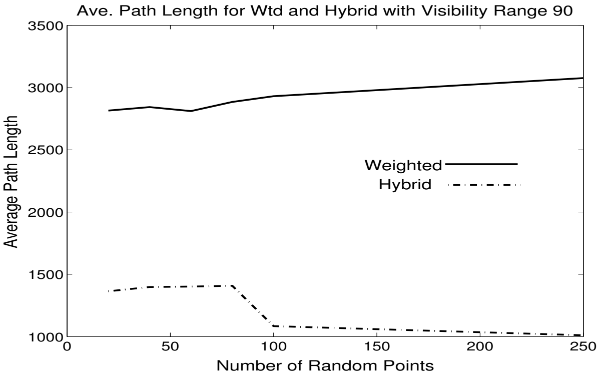


Fig. 11. Weighted versus hybrid strategies for COL with visibility range 90. 120 trials were performed.

(this is depicted in Figure 17). In addition, the distance traveled by the robot to observe a disambiguating feature gets larger as visibility decreases.

On the other hand, a weighted strategy that directs the robot to move to the most informative point nearest to its current location might avoid such a zigzagging effect on the robot's trajectory. Hence, as the visibility gets poorer the hybrid strategy produces path lengths that are shorter than those of the standard weighted strategy by a wider margin. Unlike traditional heuristics which have been demonstrated to yield potentially exponential path lengths [2], we see no compelling rationalization for a similar theoretical worst-case bound for the hybrid strategy.

B. *Performance of Useful Region Localization Algorithm*

Experiments were carried out to evaluate the performance of the UAL algorithm using weighted strategy with respect to different numbers of useful random points as well as various values of limited sensor visibility. We performed 120 trials with simulated office environments as in the case of the COL algorithm.

Figure 14 plots the variation of average path length with increasing numbers of useful random points, for different visibility ranges. Poorer visibility produces longer path lengths. And unlike the COL algorithm, with any visibility range, increasing the number of random points leads to shorter path lengths. Figure 15 shows the effect of deteriorating visibility on the average path length for 500 useful points as well as for 1 useful point. Figure 16 compares the performance of the COL and UAL algorithms with decreasing visibility range, for 250 random points.

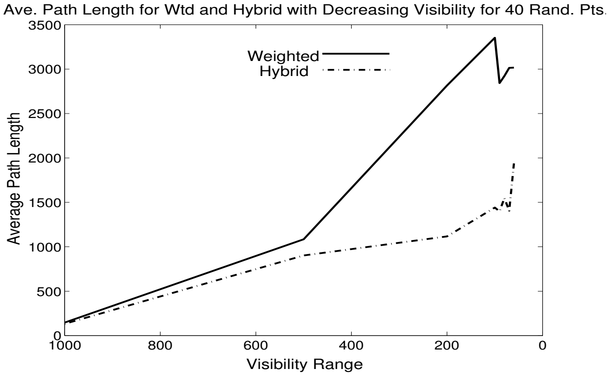


Fig. 12. Performance of weighted versus hybrid strategies for COL as visibility decreases, with 40 random points. The results are based on 120 trials.

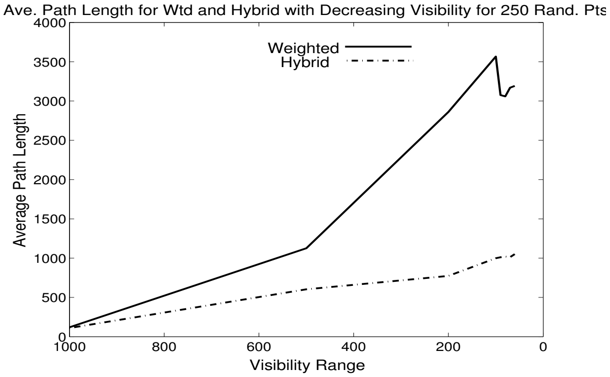


Fig. 13. Performance of weighted versus hybrid strategies for COL as visibility decreases, with 250 random points. The results are based on 120 trials.

When we examine the results for limited sensor visibility, we find that upwards of 80 random points, the UAL algorithm produces path lengths that are at least equal to those produced by the COL algorithm, and at times much shorter. For example, in Figure 16 the average path length obtained for the UAL algorithm starts out to be longer than that of the COL algorithm but very quickly becomes much shorter than that of the COL algorithm for lower visibility range values. As visibility gets poorer, the size of the useful region gets smaller since the robot must approach the disambiguating landmarks at very close proximity in order to see them. In such circumstances, choosing points from the entire overlay intersection region might result in

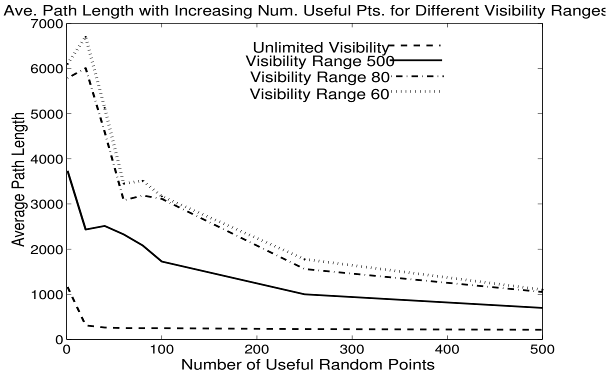


Fig. 14. Performance of the UAL algorithm using weighted strategy with different visibility ranges. The results are based on 120 trials.

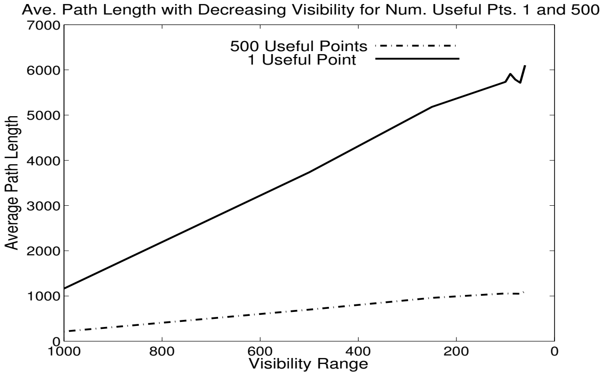


Fig. 15. Performance of the UAL algorithm using weighted strategy with only 1 useful random point versus 500 useful random points. The results are based on 120 trials.

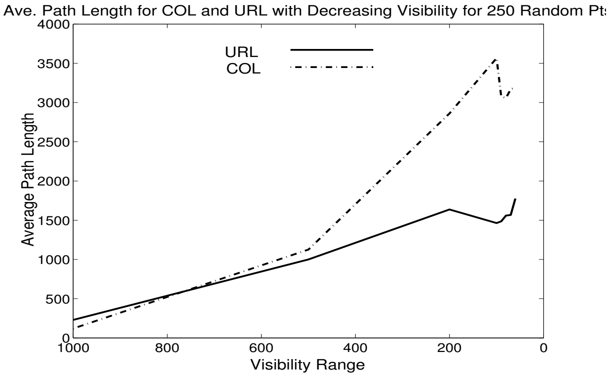


Fig. 16. Performance of the UAL versus COL algorithms using weighted strategy as visibility decreases, with 250 random points. The results are based on 120 trials.

very few of them actually coming from the useful region.

The path lengths generated by the COL algorithm for the entire range of random point quantities for limited visibility values 200 and below appear to be more or less the same. The COL algorithm functions by selecting increasing numbers of points in iterations until some are useful. Since the useful region itself gets smaller as visibility is reduced, the number of useful points uncovered by the COL algorithm remains approximately the same regardless of whether it starts off with a relatively small number of points which are augmented in iterations in order to yield useful ones, or it starts off with a relatively large number of points which might not require much reinforcement in order to uncover some that are useful. Incrementing the total number of points chosen in the overlay intersection will tend to increase the number that prove useful, but the relative increase is not sufficient to really make a difference to the path length.

It follows that the more limited the visibility of the robot's sensor, the more the number of hypothetical locations that are likely to be generated. As well, the length of the localization trajectory is likely to be longer. The increased length of the localization trajectory can be attributed to the robot's "short-sightedness" which requires it to move much closer to distinguishing features in order to be able to see them than otherwise. Simply having a larger number of hypotheses to eliminate does not necessarily imply longer path lengths.

VI. Discussion and Conclusions

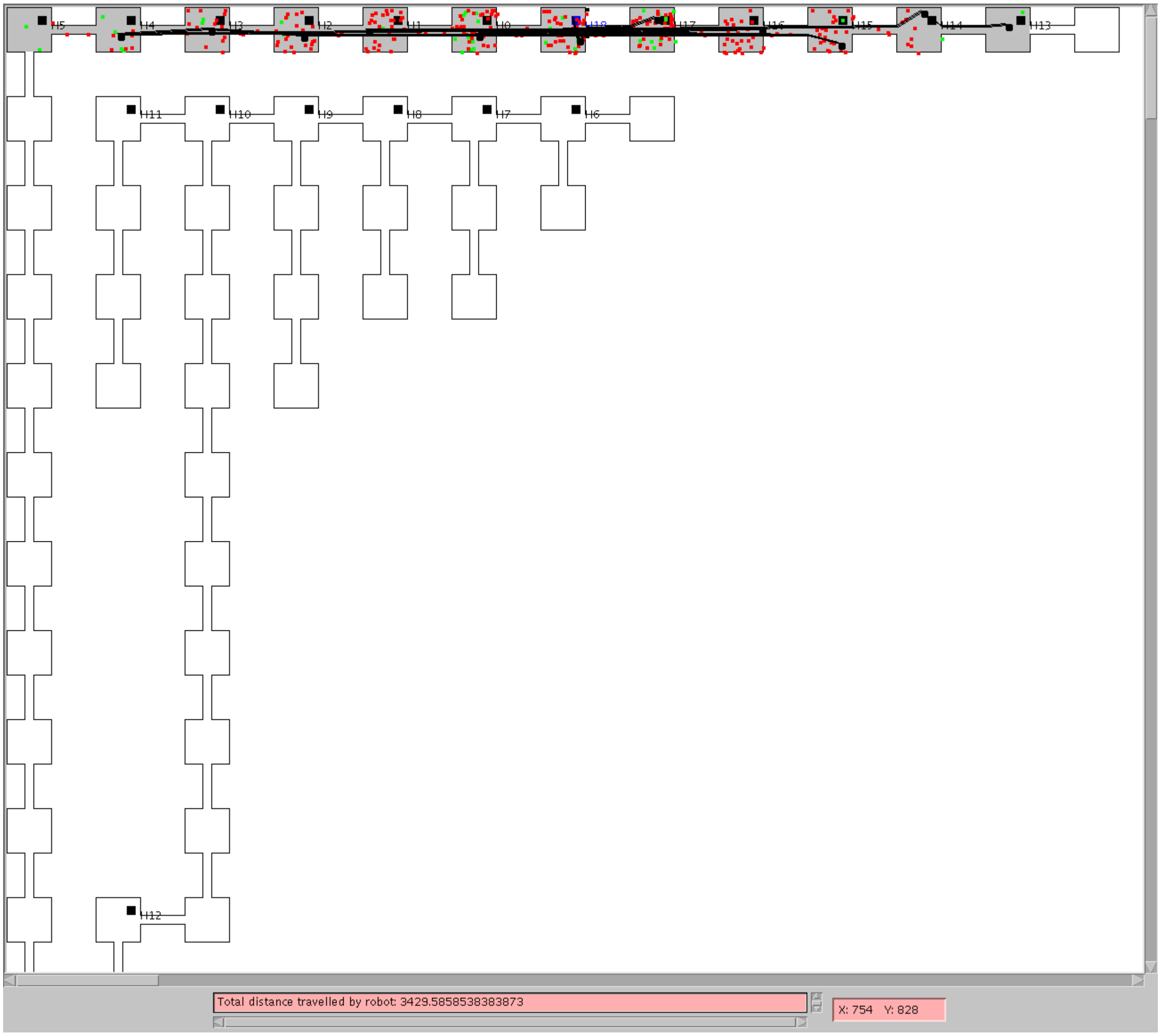
In this paper we have presented two algorithms, COL and UAL, for minimum distance localization incorporating limited sensor visibility. The algorithms produce localizing paths for a robot that are more feasible, both computationally and practically, than those that would be obtained from earlier theoretical analyses. In contrast to prior work, these algorithms and performance analyses are based on expected-time behavior rather than worst-case theoretical bounds.

Inevitably, the use of a sensor with bounded range makes the minimum distance localization problem more difficult than it would be with an infinite-range sensor. This arises due to the fact that less of the environment may be visible from many vantage points, and hence more travel may be necessary to resolve ambiguity. In this work we do not model the additional complications that would arise due to sensor noise, imperfections in the robot's map, or odometry error. While the effects of noise can be readily accommodated, finding a model that would take into account odometry errors remains an interesting open problem. In order to model explicitly the process under odometry error entails not only a suitable model of localization accuracy, but also a choice regarding how to weight large-scale ambiguity versus local pose estimation errors.

References

[1] R.G. Brown and B.R. Donald, *Mobile robot self-localization without explicit landmarks*, Algorithmica **26** (2000), no. 3/4, 515–559.

[2] G. Dudek, K. Romanik, and S. Whitesides, *Localizing a robot with minimum travel*, SIAM J. Computing **27** (1998), no. 2, 583–604.



```
Total distance travelled by robot: 3429.5858538383873
X: 754   Y: 828
```

Fig. 17. Localization trajectory in simulated office environment with limited visibility. The standard weighted decision strategy was used which directs the robot to weight the information gain at a potential probe destination with its distance from the robot's *initial* location. The robot travels in a zigzagging fashion, resulting in a long trajectory.

[3] D. Fox, W. Burgard, and S. Thrun, *Active markov localization for mobile robots*, Robotics and Autonomous Systems **25** (1998), 195–207.

[4] L. Guibas, R. Motwani, and P. Raghavan, *The robot localization problem*, SIAM J. Computing **26** (1997), no. 4, 1120–1138.

[5] J. Kleinberg, *The localization problem for mobile robots*, In Proc. 35th IEEE Conference on Foundations of Computer Science (Santa Fe, NM), IEEE Computer Society Press, 1994, pp. 521–533.

[6] S. Schuierer, *Sensing, modelling and planning*, Intelligent Robots, ch. Efficient robot self-Localization in simple polygons, pp. 129–146, World Scientific Publ., 1996.

[7] M. Rao, G. Dudek, and S. Whitesides, *Randomized Algorithms for Minimum Distance Localization*, In Proc. Sixth International Workshop on the Algorithmic Foundations of Robotics (WAFR), Utrecht, The Netherlands, 2004, pp. 265–280.