In Proceedings of the International Conference on Robotics and Automation, New Orleans, LA, May 2004, pp. 2379-2385

A Multi-robot Approach to Stealthy Navigation in the Presence of an Observer

Ashley D. Tews Gaurav S. Sukhatme Maja J. Matarić

Computer Science Department
University of Southern California
Los Angeles, USA
atews|gaurav|mataric@robotics.usc.edu

Abstract— **In this paper, we propose a simple, reactive method for multiple robots carrying out sequential low-visibility navigation in the presence of an observer. Initially, the robots have no map of the environment but know the locations of the observer and goal. They generate an occupancy grid representation of the environment which is modeled using potential fields with embedded task information. These fields are combined and navigation waypoints extracted. Each robot carries out its traverse independently and shares its experience with its successor. The experience information consists of the occupancy grid and a filtered version of the traveled path used to assist the subsequent robot to traverse a lower visibility path. This produces a robust and reactive solution for stealthy navigation since there is no global path planning and the robots are not committed to any particular path. Experiments in simulation and real outdoor environments substantiate the approach and demonstrate the advantage of sharing information to reduce cumulative visibility. The experiments also demonstrate the algorithm's versatility in taking advantage of an environment that changes between robot traverses.**

Keywords-component; multi-robot; navigation; potential fields; outdoor environments; stealth

I. Introduction

In concurrent tasks, adding contributors is intuitively expected to improve task performance. In many multi-robot tasks, this is the case up to some critical limit, after which additional robots decrease overall performance due to resource conflicts and physical interference. Tasks that require an element of *stealth*, i.e., low visibility of the robots, are examples where it can be expected that increasing group size decreases task performance due to the larger profile that results. Various stealthy multi-robot tasks can be performed on or in the presence of single or multiple observers, including locating, corralling, surrounding, and navigation. In each of these, the robots need to carry out the task while maintaining as low a profile as possible. While larger teams may result in a higher *probability* that the robots will be detected, there are also advantages to using multiple robots for stealth tasks: each robot adds its own sensory data and capabilities to the team to reduce overall team exposure. It is our goal to determine these advantages of multi-robot stealth strategies and apply them to real multi-robot outdoor environments. In this paper, we present our initial approach to this problem by developing an algorithm for multi-robot navigation in the presence of a single observer. It is assumed the observer has unlimited omni-directional sensing and the environment consists of objects that can occlude the robots from the observer's sensors. The team initially has no map of the environment but the locations of the goal and the observer are known.

In our approach, the robots carry out their traverses one at a time, sequentially, and generate an occupancy grid representation [1] of the environment *en route*. The occupancy grid is modeled by potential fields [2], [3] along with task-specific information such as observer and goal position. The combination of the fields forms an abstract view of the environment from which navigation waypoints are extracted. To take advantage of multiplicity, each robot commencing a traverse is provided with the occupancy grid and filtered path information from the previous robot. The filtered path is a waypoint list generated from events that occurred along the robot's path. The successor robot uses this information to make decisions about waypoint selection and overwrites the provided occupancy grid with its ego-centric sensor data. By sharing information, each robot follows a lower-visibility path than its predecessor, and in the case of static environments, the paths are traversed more efficiently.

To demonstrate the effectiveness of this approach, we conducted experiments both in simulation and in a real outdoor environment with three Pioneer AT mobile robots. The environment configurations allow the robustness of the approach to be evaluated in terms of low-visibility path selection, the benefits of sharing information, repeatability, and reactivity to a changing environment.

This paper is organized as follows. Section II presents an overview of research related to stealthy robot navigation. Section III describes our algorithm for multiple robot stealthy navigation. Section IV presents experiments in simulation and in a real domain. Section V draws the conclusions and Section VI summarizes future directions for this work.

II. Related Work

To date, there has been little research conducted for stealthy tasks with real robots. More popular, but distantly related, are the research areas of target tracking and predator/prey. However, tasks in these domains generally require the robots to be exposed to the target or prey; stealth is rarely a consideration.

Research concerning stealthy tasks includes visibility-constrained terrain evaluation for stealthy path planning [4] [5], multi-vehicle sensor coverage algorithms [6], and single robot stealthy traverses in the presence of a known observer [7]. Each of these is examined in this section.

Teng et al. [4] present an approach to low-visibility incremental path planning for an agent in the presence of observers with linear trajectories over a digitized terrain. Their approach uses a massively parallel hypercube machine to assist in determining the visibility constraints from the observers' projected positions at each planning interval. Waypoint selection is determined by these constraints and the agent's current trajectory. The resulting waypoints are positions in the environment that are guaranteed to be hidden from the observers and reachable by the agent within the planning interval.

[5] describe methods of utilizing terrain features to plan stealthy paths through outdoor environments in the presence of an observer. The method involves extracting features from moving images that portray significant curves displaying permanence over multiple images. They rationalize that visual servoing techniques are better suited to stealthy behaviours than frontier-based methods which may fail at critical times during the traverse due to localization errors.

[6] examine the problems of locating strategic points in an environment to observe a specific area, and dynamic sensor positioning on vehicles traveling in a formation to maximise and balance coverage during a traverse. Both methods involve reasoning about the environment state, the vehicles' sensor capabilities, and the task requirements. In the former method, candidate locations are evaluated in terms of their stealth versus their potential information content. The location that offers the most visibility and coverage is selected. Dynamic sensor positioning evaluates the current environment state and the position of the other vehicles in formation to determine the most effective scan pattern for each vehicle's sensor. The scan pattern changes to focus on areas that contain potentially more information about threats such as open ground opposed to an area occluded by a nearby tree.

Our approach is similar to the single robot stealthy navigation algorithm of [7]. The assumptions of an observer with unbounded detection range and an initially unknown environment are the same. The environment is mapped as an occupancy grid which is then modelled with potential fields. Waypoints are extracted from these fields in the areas between the robot and the goal with a preference for occluded areas. Our approach extends this concept to multiple robot teams and uses a different potential field approach.

III. USING POTENTIAL FIELDS FOR STEALTHY NAVIGATION

Potential fields have been used in many areas of robotics, including manipulator obstacle avoidance [2], reactive navigation [3], and robot soccer [8]. They provide a beneficial medium for modeling various elements of the robot's task and environment. The fields can be combined and regions of interest analysed to extract important task-related information. Using a simple path-planning example, global potential fields covering the entire environment can be used to represent the distances from the goal and from the robot's location, and local high-valued fields to represent the obstacles. By combining these, the centroid of the lowest-valued region can be extracted as the next waypoint for the robot. Carried out iteratively, this ultimately results in an obstacle-free shortest path to the goal, since each step resets the robot's field to a new location.

Our approach uses the components of the above general potential field method, with an additional field to model the observer. The combination of these fields creates a virtual, task-relevant environment representation from which waypoints are extracted. In this representation, the effect of the observer's location encourages less observable traversal paths. The main potential fields we use are generated from:

- distances from the robot, observer, and goal, and
- occluded areas (shadows) behind objects with respect to the observer's position.

The distance fields represent the distance from a specified point to the boundary of the environment and are modeled as either attractors (distances from the robot and goal) or repellors (distance from the observer). Examples of these are shown in Figure 1a and Figure 1b respectively. Areas in the environment that are not visible from the observer are modeled as 'shadow' attractor potential fields, as shown in Figure 1c. These are derived from the robot's current occupancy grid.

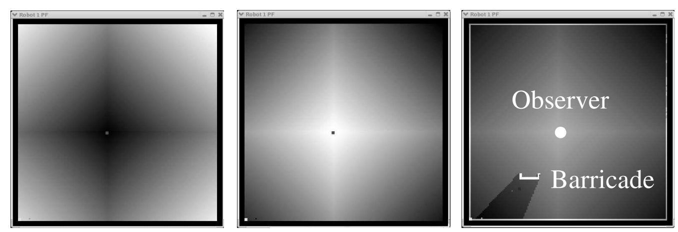


Figure 1. An example of the potential fields: a) attractive, b) repulsive and c) barricade shadow (attractive). Darker areas are more attractive for the robot to travel to.

Each field has an associated bias parameter that determines its level of influence on waypoint extraction when the potential fields are combined. These biases are adjusted empirically for the environment configuration and the robot dynamics, which vary with the environment conditions. Since there are only a few potential fields in this method, there are few parameters to adjust.

The method for generating and using these potential fields is shown in the algorithm below. It is decentralized, and executed on board each robot.

Obtain the occupancy grid and filtered path from previous robot if applicable

While not close to the goal

    Update the occupancy grid with current sensor scans

    Combine potential fields based on the occupancy grid objects, distances from the goal, robot and observer

    Add hysteresis to the last waypoint to reduce oscillations in waypoint choice

    Add local repellor potential field at robot's position to discourage nearby waypoint selection

Extract the centroid of the lowest valued region[1] in the resulting potential field as a candidate waypoint

Adjust waypoint position according to the following rules:

- In general, if the waypoint is not forward of the plane located at the robot's position and orthogonal to the goal heading, discard it and select one that is. There is a small level of variance allowed so the robot can travel away from this plane in order to negotiate a strategic barrier (illustrated in the initial part of robot 1's traverse in Figure 4b).
- If there is an object between the waypoint and the robot, the waypoint is relocated to a position between the robot and the corner or end of the object closest to the robot, but farthest from the observer.
- From the robot's position, if the waypoint is within five degrees of the next filtered path waypoint, use the filtered path waypoint instead.

Update the filtered path waypoint list if necessary

Navigate towards the waypoint

Return

Algorithm 1. The algorithm for selecting waypoints.

To summarize, before a robot commences its traverse, it requests the occupancy grid and a filtered version of the path from the previous robot, if one is available. The occupancy grid shows the explored areas and objects in the environment. The filtered path is a series of waypoints derived from events that occurred during the robot's traverse. These events are defined as deviations from the current heading by more than 30 degrees and places where the robot is about to cross a frontier between unobserved (shadow) and observed space or has crossed from observed space to unobserved. A waypoint from the filtered path of the previous robot is preferred over a potential field generated waypoint if they are along similar trajectories, as this produces smoother and faster paths.

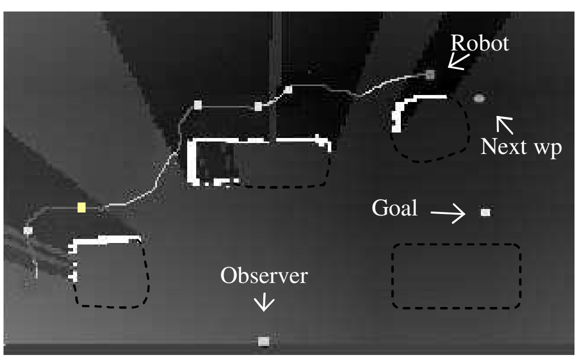


Figure 2. Example potential field view of a 55m by 55m environment extracted from the occupancy grid. The broken black lines indicate unobserved objects.

[1] A region is used rather than a point to provide a more general and rational solution, to reduce the effects of sensor noise, and to keep the waypoints away from objects. The region size is empirically set to eight occupancy grid pixels or less.

Figure 2 shows an example of the first robot during its traverse. The potential field is generated from the current occupancy grid. The objects (barricades) are the three white structures. Behind each are the 'shadow' potential fields cast from the observer's position. Since low values are considered attractive, the algorithm selects waypoints in these areas. The waypoint adjustment rules ensure the selected waypoint is not behind the robot. Otherwise, the robot would not leave shadow regions until they were filled with the high-valued robot-position local potential fields. This adjustment is evident in the figure with the next waypoint being between the robot and the goal, presumably in observed space.

Also shown in Figure 2 is the robot's path with the filtered waypoints superimposed. The path is light and dark grey, signifying where the robot assumed it was and was not being observed respectively. The filtered path waypoints in the figure are generally located before and after shadow frontiers.

A waypoint attracts the robot until a new one is chosen allowing new environment information to be integrated into the reactive path planning. Figure 3 shows another traverse in the same environment as Figure 2 with the waypoints superimposed as grey dots and a broken white line connecting them for temporal clarity. These waypoints provide a crude path for the robot but since they are forward-projected, they tend to 'tow' the robot towards them until another is selected. Also, these points are generated with the current environment information available at the time which, viewing them retrospectively in the more complete figure shown may make them appear in odd locations. For example, the waypoint inside the upper right barricade appears to be erroneously selected since the robot cannot travel inside the barricade. However, at the time it was chosen, the barricade extents had not been sensed.

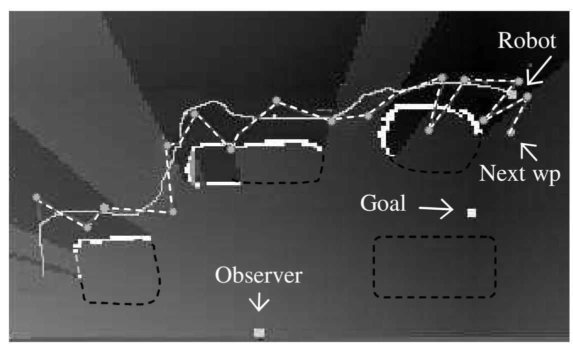


Figure 3. The effect of the waypoints (grey dots) on the robot's path (solid line).

IV. IV. EXPERIMENTS

Experiments were conducted in simulation and real environments to demonstrate and validate our approach. The simulation experiments were carried out on the Stage simulator using Player devices [9]. Apart from being a valuable development tool, the simulator is beneficial for creating different environments for testing the algorithm's robustness. The verisimilitude of the simulator should produce similar algorithmic performance to its real counterpart if it is to be useful. However, it is difficult to simulate the uncertainties of a

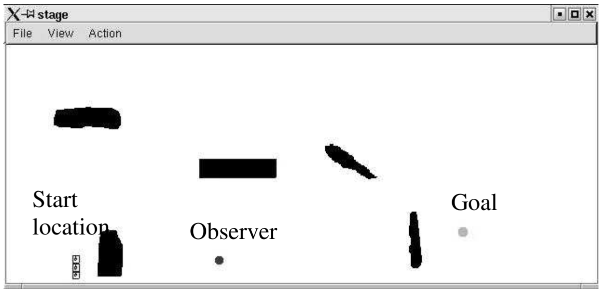
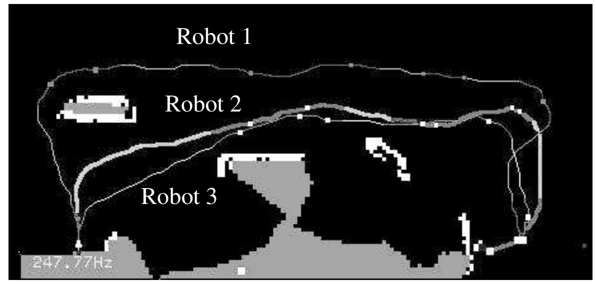


Figure 4. a) Overview of the environment for experiment 1.



b) A simulation run with all robot traverses superimposed on the third robot's occupancy grid.

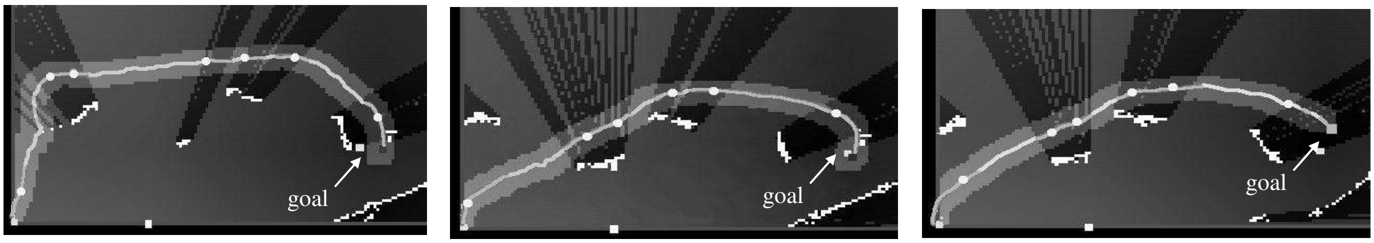


Figure 5. A typical example of the three robot traverses in the outdoor environment. Localisation variations between runs placed the goal and observer in slightly different locations. Also, there are artifacts shown near the goal in the left and centre images. Despite the noisy environment representation, the robots produced repeatable performance in all runs.

real environment. In outdoor environments, the performance of the robots is affected by various environmental conditions that can change depending on the type and time of day. Sunlight affects laser rangefinders, GPS satellite position and terrain openness affect GPS-based localization, and terrain surface variations (mud, dew, sand, etc.) affect the robot's dynamics. This is in addition to the usual noisy sensors and localization errors. Nevertheless, if the variation in these uncertainties is not extreme, we expect the robustness of the software architecture and the algorithm to accommodate them and produce repeatable and predictable performance.

The environment in these experiments is a relatively flat lawn-type area measuring approximately 35m x 35m illustrated in Figure 4a and Figure 6a. The configuration of the barricades is chosen to highlight the theme of each experiment. The size of the occupancy grid and global potential fields for the robots is 150 by 150 pixels which results in an environment resolution of 233 mm.

Three Pioneer AT robots were used in each experiment, equipped with SICK laser rangefinders (configured for 8m) for occupancy grid generation and obstacle avoidance, and DGPS and a 3DMG IMU for localization. In simulation, robots had near-perfect localization. In the real environment, localization precision was generally 1.5 m or better. The robots were controlled using the Player interface which allows near-seamless transportability between programs developed in simulation and the real environment.

Before each experiment commenced, the potential field parameters were empirically tuned for the initial environment configuration. For real experiments, the tuning was also required to account for the dynamics of the ATs in response to the current environment conditions. The parameters were copied to each robot and did not change during the experiment.

Performance in each experiment was evaluated in terms of the time each robot took for its traverse, the distance it traveled and, most importantly, the amount of time it was in the observer's sensor range. Additionally, for the simulation experiments, one-way ANOVA significance tests were performed.

A. *Experiment 1 – The benefits of multiple robots*

The goals of this experiment are to demonstrate the algorithm's path selection and the improvement of stealthy traverses from the use of shared information. The experiment was repeated three times in the real environment and ten times in simulation. Each robot started after the previous had reached the goal. This enabled the complete traverse experience to be transferred to the subsequent robot. The simulation and real environments had a similar barricade configuration as shown in Figure 4a. The main difference between the two environments is the location of the goal. In the real environment, the goal was placed higher due to the fence line shown in the lower right of the images in Figure 5.

An example simulation trial is shown in Figure 4b. The robots' paths are superimposed on the third robot's final occupancy grid. The observer's location is the shown at the bottom center of the diagram and the goal location behind the barricade on the right. The black regions indicate known areas, grey unknown and white indicates an object. Figure 5 shows an example of a real robot trial. Each robot's path is shown on the potential field generated at the end of the traverse. In both Figure 4b and Figure 5, the paths also show the filtered waypoints that were passed on to the subsequent robot.

The most noticeable difference in robot paths is between the first traverse and the others. The first traverse was least efficient since it had no initial environment information and involved blind decisions until a barricade was detected. Consequently, it traveled around the upper left barricade before heading towards the goal, hiding behind the barricades along the way.

The second robot's traverse (shown in Figure 4b and the centre image in Figure 5) takes advantage of the occupancy grid and filtered path passed to it from the first robot. Although it may appear counter-intuitive to avoid the first barricade, the result is a lower-visibility path. This is due to its path between the start position and second barricade being more direct. The robot could therefore travel quicker through the visible region. The third robot's path was similar to the second, and the effect of using the filtered path information enabled a more efficient traverse.

Table 1 and Table 2 contain the average performance statistics for the ten simulation trials and three real robot trials, respectively. The 'Time' and 'Dist' columns refer to the time taken and distance traveled for the traverse. The 'Ass. Det' column indicates the time the robot assumed it was being detected during its run. These values were calculated during the trial and are based on the incomplete environment information available at the time; they are thus not an accurate representation of the detection times. They are included to indicate how accurately the robots perceived the situation and are not used for any further analysis due to their inaccuracy. The 'Act. Det.' time is the actual time the robot was observed. This is derived from analyzing video information taken from the observer's position for the real robot experiment, and by an observer robot in simulation. The 'No stealth' traverses are where the robot traveled directly to the goal disregarding the observer's position. These are included as a baseline for comparison with the stealthy traverses.

Table 1. Experiment 1 simulation average results.

Traverse	Time s	Dist m	Ass. Det. s	Act. Det. s
No stealth	39.0	34.5	N/a	37.2
1	67.8	57.0	32.0	38.6
2	51.7	47.3	24.7	27.0
3	48.5	45.2	22.8	26.0

Table 2. Experiment 1 real robot average results.

Traverse	Time s	Dist m	Ass. Det. s	Act. Det. s
No stealth	55	32	N/a	50
1	88	47	42	50
2	65	37	37	41
3	61	37	33	38

The relationship between traverses correlates between tables which indicates the algorithm performed similarly in both domains. Each stealthy traverse improved on the previous with an overall improvement in the actual detection times over the non-stealthy case. There is also a strong correlation in the overall trend of the results between tables which reflects the simulator's verisimilitude and adds credibility to the statistical analysis.

Intuitively, the non-stealthy case should provide the shortest path to the goal and maximum visibility to the observer. The stealthy traverses should show successively shorter traverses and lower detection times, even if the robots take the same course through the environment due to the effect of sharing experiences.

One-way ANOVA significance tests carried out on the ten simulation runs support the above expectations of the algorithm's performance. In most cases, there is a significant difference ($P < 0.001$) between the results for the traverses. Considering this with the values in the tables indicates that the first stealthy traverse is not as good as the non-stealthy traverse, the second stealthy traverse is better than the first (and therefore the non-stealthy traverse) and the third is best of all. The exceptions ($P > 0.05$) are the actual detection times between: the non-stealthy and first stealthy traverse, and between stealthy traverses two and three. The discrepancy with the expected better performance of the stealthy algorithm over the non-stealthy case is due to the stealthy robot navigating around the uppermost barricade in the absence of better initial environment knowledge which increased its exposure to the observer. This indicates that although the algorithm finds an intuitive path behind the available objects during the first pass through the environment, there is no significant advantage in reduced visibility compared to a direct traverse.

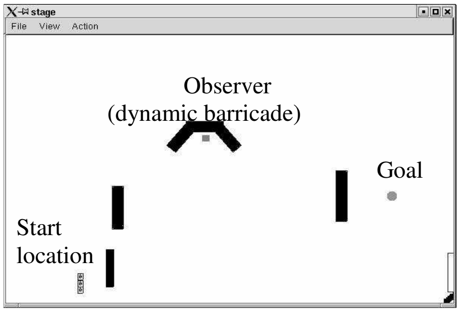
B. Experiment 2 – Dynamic environment

The purpose of this experiment was to demonstrate how the algorithm performs with an environment that changes between traverses. The environment initially contained barricades covering the start and goal locations and then a barricade was added around the observer as shown in Figure 6a.

The experiment was carried out once in the real environment and ten times in simulation. An example simulation run is shown in Figure 6b and the real robot run in Figure 7. The filtered path waypoints are annotated as dots on each robot's path.

The first robot made its traverse in near-complete visibility of the observer similarly to a non-stealthy traverse as shown in Figure 6b and the leftmost image in Figure 7. After its traverse, a barricade was placed around the observer's location as shown in Figure 6a. The addition of this barricade produced a zero-visibility path around the observer.

The second robot started with the occupancy grid and filtered path from the first robot which did not include the new barricade. Upon its discovery, the robot changed its path to travel around the observer (Figure 6b and the centre image in Figure 7). The third robot was provided with the second robot's updated occupancy grid and filtered path which allowed it to generate a smoother trajectory for the traverse and improve its



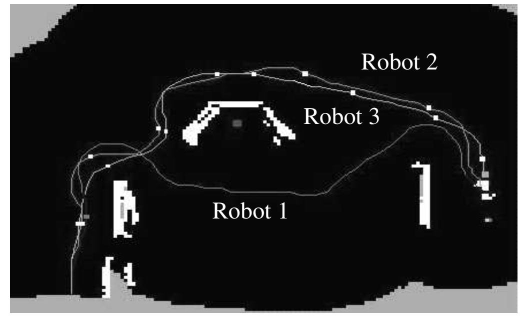
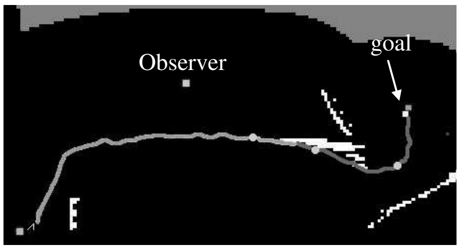
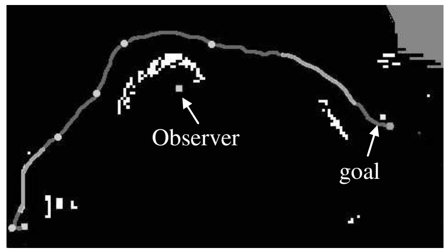


Figure 6. a) Overview of the environment used in experiment 2.

b) A typical simulation run with all traverses superimposed on the third robot's occupancy grid





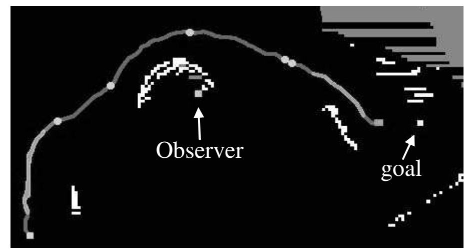


Figure 7. The real robot traverses for experiment 2. The observer's barricade was not fully detected by the robots so they assumed they were seen close to the goal when in fact, they were not.

performance further (Figure 6b and the rightmost image in Figure 7).

The performance statistics for the experiment are shown in Table 3 for simulation and Table 4 for the real environment. As before, the entries annotated as 'No stealth' are cases where the robot navigated directly to the goal, disregarding the observer. These results are shown with and without the barricade around the observer annotated as 'with barr.' and 'w/o barr.', respectively.

A statistical analysis of the results shows a significant difference ($P < 0.01$ or less) between all stealthy traverses except for the actual detection times for traverses 2 and 3, and the travel time and distances between traverses 1 and 3 ($P > 0.05$). The similarity between traverses 1 and 3 indicates that a traverse around the observer's barricade can be as efficient as a traverse directly to the goal.

Table 3. Experiment 2 simulation average results.

Traverse	Time s	Dist m	Ass. Det. s	Act. Det. s
No stealth with barr.	59.5	48.0	N/a	13.3
No stealth w/o barr.	59.6	43.0	N/a	45.5
1	51.5	45.7	40.9	41
2	59.8	49.2	20.8	0
3	53.9	46.7	14.7	0

Table 4. Experiment 2 real robot average results.

Traverse	Time s	Dist m	Ass. Det. s	Act Det. s
No stealth w/o barr.	83.2	37.7	N/a	73
1	82.9	40.6	74.5	59
2	87.6	48.0	56.4	0
3	78	44.9	63.8	0

The non-stealthy traverses are carried out with and without the observer's barricade to allow the stealthy traverses with the same environment configuration to be compared. Stealthy traverse 1 shows a significant improvement ($P < 0.01$ or less) in travel and detection times over its matching non-stealthy case without the barricade. Although it is assumed these traverses should produce similar results, the difference is mainly due to the stealthy algorithm providing a waypoint-generated path (with implicit obstacle avoidance) while the non-stealthy case uses the goal location as the only waypoint. The non-stealthy robot must negotiate the barricades as obstacles, which is less efficient.

Comparing stealthy traverses two and three to the non-stealthy case with the observer's barricade shows that most of the results are similar ($P > 0.05$) except for the time taken for stealthy traverse three, and naturally, the actual detection times ($P < 0.001$) which are worse for the non-stealthy case. Analyzing these results with the values in the table indicates that the stealthy traverses result in significantly lower visibility than the non-stealthy case. Also, it is possible to produce a path that is as efficient as the non-stealthy traverse, although this is a side effect of the low-level navigation rather than an optimization demonstrated by the stealthy algorithm.

V. CONCLUSIONS

The goals of this research are to develop a method for multi-robot low-visibility navigation in the presence of an observer and to demonstrate how multiple robots can be beneficial in such a task. The assumptions are: the robots carry out their traverses one at a time, they do not have an initial map of the environment but know the locations of the goal and observer, and the observer has an unbounded sensor range. The method developed is based on extracting waypoints from a potential field view of the task based on current environment knowledge. Additionally, traverse experiences are shared between robots in the form of an occupancy grid and filtered path information.

The method produces intuitive low-visibility paths with and without *a priori* environment information, as is evident by the results of the first and subsequent robot traverses respectively. Each traverse shows improvement over the previous in efficiency and visibility to the observer. The experiments conducted in simulation and in the real environment support the above claims. The simulation results correlate strongly to those in the real environment which demonstrates reproducible behaviour of the algorithm in different domains and the realism of the simulator. The results also demonstrate the robustness of the algorithm in the presence of a changing environment. The strength behind the approach lies in its simple and reactive use of potential fields to model the environment and the task. However, their parameters must be carefully tuned prior to use. For our application, this primarily arises from the complexity of the environment and the dynamics of the robots, particularly to varying outdoor conditions. Since we use only a few potential fields, parameter tuning is carried out quickly, even in real environments.

VI. FUTURE WORK

The approach has been rigorously tested in simulation and the real world. It shows promise for application to more demanding and complex environments such as MOUT sites. Also, the environment representation can be expanded to include natural terrain features beneficial for occluding the robots. The next stage of testing will focus on demonstrating the algorithm in these types of environments initially using the newly developed Gazebo 3d outdoor simulator [10] before verification in the real domain.

Apart from testing in different environments, features of the observer will also be varied. The observer can be made mobile with an unknown initial location. It may also reason about its situation as a rational entity by reacting to robot positions and the environment. Its sensing will be limited in range and have a periodic scan which the robots can take advantage of by modelling with potential fields and combining with those in the existing approach.

ACKNOWLEDGMENTS

This work is supported in part by DARPA grants DABT63-99-1-0015, and 5-39509-A (via UPenn) under the Mobile Autonomous Robot Software (MARS) program.

REFERENCES

[1] A. Elfes. "Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation," PhD Thesis, Carnegie Mellon University, 1989.

[2] O. Khatib. "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," The International Journal of Robotics Research, vol. 5, no. 1, 1986, pp. 90-98.

[3] R. Arkin. "Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation," Robotics and Autonomous Systems, no. 6, 1990, pp. 105-122.

[4] Y. Teng, D. DeMenthon and L. Davis. "Stealth Terrain Navigation," IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 1, 1993, pp.96-110.

[5] S. Ravela, R. Weiss, B. Draper, B. Pinette, A. Hanson and E. Riseman. "Stealth Navigation: Planning and Behaviors," ARPA Image Understanding Workshop, Monterey, CA, 1994, pp. 1093-1100.

[6] D. Cook. P. Gmytrasiewicz and L. Holder. "Decision-Theoretic Cooperative Sensor Planning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, 1996, pp.1013-1023.

[7] E. Birgesson, A. Howard and G. Sukhatme. "Towards Stealthy Behaviors," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2003, Las Vegas, pp 1703-1708.

[8] A. Tews and G. Wyeth. "MAPS: A System for Multi-agent Coordination," Advanced Robotics, vol. 14, no. 1, 1999, pp. 37-50.

[9] B.Gerkey, R. Vaughan, K. Støy, A. Howard, G. Sukhatme and M. Matarić. "Most Valuable Player: A Robot Device Server for Distributed Control," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), 2001, pp. 1226-1231.

[10] Gazebo. An Outdoor Multiple Robot Simulator. http://playerstage.sourceforge.net/gazebo/gazebo.html Last updated 19 November 2003