

Including communication success in the estimation of information gain for multi-robot exploration

Arnoud Visser and Bayu A. Slamet
Intelligent Systems Laboratorium Amsterdam
Universiteit van Amsterdam (UvA)
Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands
<http://www.science.uva.nl/research/isla>

Abstract—This article investigates the effect of incorporating knowledge about the communication possibilities in an exploration algorithm used to map an unknown environment. The mission is to explore a hypothetical disaster site with a small team of robots. The challenge faced by the robot team is to coordinate their actions such that they efficiently explore the environment in their search for victims. The coordination can only be optimal when the robots share the same map. With a limited communication range the map cannot be shared in all circumstances. This article concentrates on the effect of a distributed map, where each robot has only has knowledge of a part of the global map and has no guaranteed connection to the other robot or the operator.

I. INTRODUCTION

This paper¹ will investigate the influence of a limited communication distance on a multi-robot exploration approach that was designed with the disaster sites of the RoboCup Rescue Competitions in mind. These scenarios can be simulated in the real world [1] or virtually within the USARSim simulator [2]. In either case, the team of robots is challenged to explore the site and locate victims in a constrained amount of time. Afterwards, the efforts of the robot team are evaluated on the amount of covered area, the quality of the produced map and most importantly the number of located victims. See [3] for a more detailed discussion on this scoring process.

The exploratory efforts exposed by our robots have so far been governed by strictly reactive behavior (2006) and tele-operation (2007). Although the autonomous behavior has demonstrated good robustness and obstacle avoidance, any seemingly “intelligent” autonomous exploration effort was due to randomizations that were inherent to the behavior control design [4].

A well-known paradigm to address the multi-robot exploration challenge in a more intelligent fashion is *frontier*-based exploration. The frontiers are typically defined as the boundaries of the currently mapped free area where the robot can enter yet unexplored area [5]. Collaborating robots can use these frontiers to coordinate their actions [6]–[8], i.e.: assign

robots to frontiers such that the robots simultaneously explore multiple yet unexplored parts of the environment. This shifts the exploration problem to a frontier assignment problem.

Most approaches use a *cost* measure to evaluate the utility of a frontier. The Manhattan distance and the anticipated traveling distance are examples of such cost measures. In [9] and [10] however, frontier evaluation approaches were presented that focus on the opposite measure: the *information gain* that can be expected if the frontier would be explored. This gain is expressed as an estimate for the amount of area that lies beyond the frontier. While [9] uses a sampling method that extrapolates the current map to estimate the information gain, the approach of [10] directly measures the expected gain from the *current* map.

This paper will incorporate a measure which estimates the probability of communication into the *information gain*. For a single robot this probability will be mainly a function of the distance from the base station. When the observations are relayed between the robots, the position of the other robots has to be included into the measure. In the following Section we will introduce our robotic and simulation platform. In Section III our simultaneous mapping and localization approach is given. In Section IV our multi-robot exploration strategy is explained. Subsequently, the value of information sharing is demonstrated in a number of exploration experiments in Section V. This will lead to the conclusions in Section VII.

II. EXPERIMENTAL PLATFORM

A. ROBOTIC PLATFORM

The robotic platform utilized in this paper is realistic simulation of the P3AT robot. The same robot was used by the SPQR team in the Rescue Robot league [11].

For our algorithm the choice of the robotic platform is of minor importance, because the method is on a much higher level than the actual movements and odometry measurements of the robot. The SLAM approach mainly relies on the measurements collected by the laser scanner, and is applicable to any robot which can carry a laser scanner (the *SICK LMS 200* weights 4.5 kg), as demonstrated in [3].

¹The research reported here is performed in the context of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

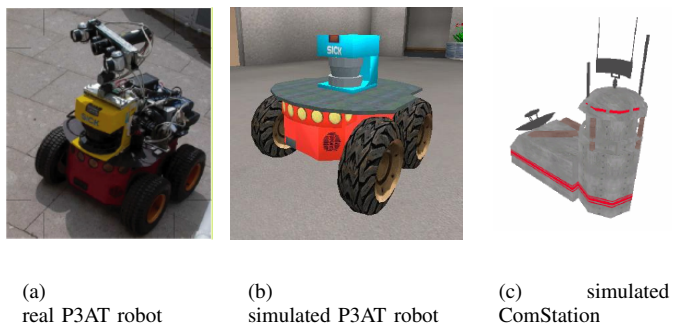


Fig. 1. The Pioneer All-Terrain Robot platform is used in the Rescue Robot league and Virtual Robot competition. The SICK laser range scanner on top and the sonar rings are clearly visible. The stereo camera on top of the real robot was not used in this research. On the right a simulated ComStation is given.

B. SIMULATION FRAMEWORK

The robots are simulated in the USARSim framework². Care has been taken to validate the simulator as good as possible [12], an effort which still continues. The USARSim framework provides a development, testing, and competition environment that is based on a realistic interpretation of a disaster scenario. The current version of USARSim [2] is based on the UnrealEngine2 game engine that was released by Epic Games as part of Unreal Tournament 2004. The Unreal engine handles most of the basic mechanics of simulation and includes modules for handling input, output (3D rendering, 2D drawing, and sound), networking, physics and dynamics. Multiplayer games use a client-server architecture in which the server maintains the reference state of the simulation while multiple clients perform the complex graphics computations needed to display their individual views. USARSim uses this feature to provide controllable camera views and the ability to control multiple robots.



Fig. 2. One of the P2AT robots in an USARSim simulated world.

When a robot is instantiated through USARSim, for each component of the robot an object is created which provides information and can be controlled. For instance objects like

²The USARSim framework can be downloaded from <http://sourceforge.net/projects/usarsim>.

wheels and sensors are part of the API to USARSim. For each class of objects there are class-conditional messages that enable a user to query the component's geography and configuration, send commands, and receive status and data. Permissible calls into the game engine and complete details on the API may be found in the USARSim Reference Manual [13].

An important aspect for this paper is the simulation of the wireless communication. All information between the robots has to go via Wireless Communication Server, which performs a sanity check and drops messages and connection when they are not longer feasible. Also the information for the operator is directed via the Wireless Communication Server. Only the information that reaches a ComStation (see Fig. 1(c)) which is physically present in the environment may be used by the operator. Only the physical layer of communication is simulated, protocol and routing issues are ignored.

III. LOCALIZATION AND MAPPING

The mapping algorithm is based on the manifold approach [14]. Globally, the manifold relies on a graph structure that grows with the amount of explored area. Nodes are added to the graph to represent local properties of newly explored areas. Links represent navigable paths from one node to the next.

The mapping algorithm is not dependent on information about the movement of the robot for the creation of links. A good estimate of the displacements can be derived only from scan matching. In practice the displacement as reported by the inertial navigation sensor is valuable as initial estimate for the scan matching. The displacement is estimated by comparing the current laser scan with laser scans recorded shortly before, stored in nearby nodes of the graph. In principle the scan matcher can also perform a comparison with measurements elsewhere in the graph, but such a comparison is only made under specific circumstances (for instance during loop closure, as illustrated in Figure 3). At the moment that the displacement becomes so large that the confidence in the match between the current scan and the previous scan drops, a new node is created to store the scan and a new link is created with the displacement. A new part of the map is learned.

As long as the confidence is high enough, the information on the map is sufficient and no further learning is needed. The map is just used to get an accurate estimate of the current location. The localization algorithm maintains a single hypothesis about where the robot currently is and does an iterative search around that location when new measurement data arrives. For each point the correspondence between the current measurement data and the previous measurement data is calculated. The point with the best correspondence is selected as the center of a new iterative search, until the search converges. Important here is the measure for the correspondence. Lu and Milios [15] have set the de-facto standard with their Iterative Dual Correspondence (IDC) algorithm, but afterward many other approaches have been proposed. Our approach is based on the Weighted Scan Matching (WSM) algorithm [16], which works

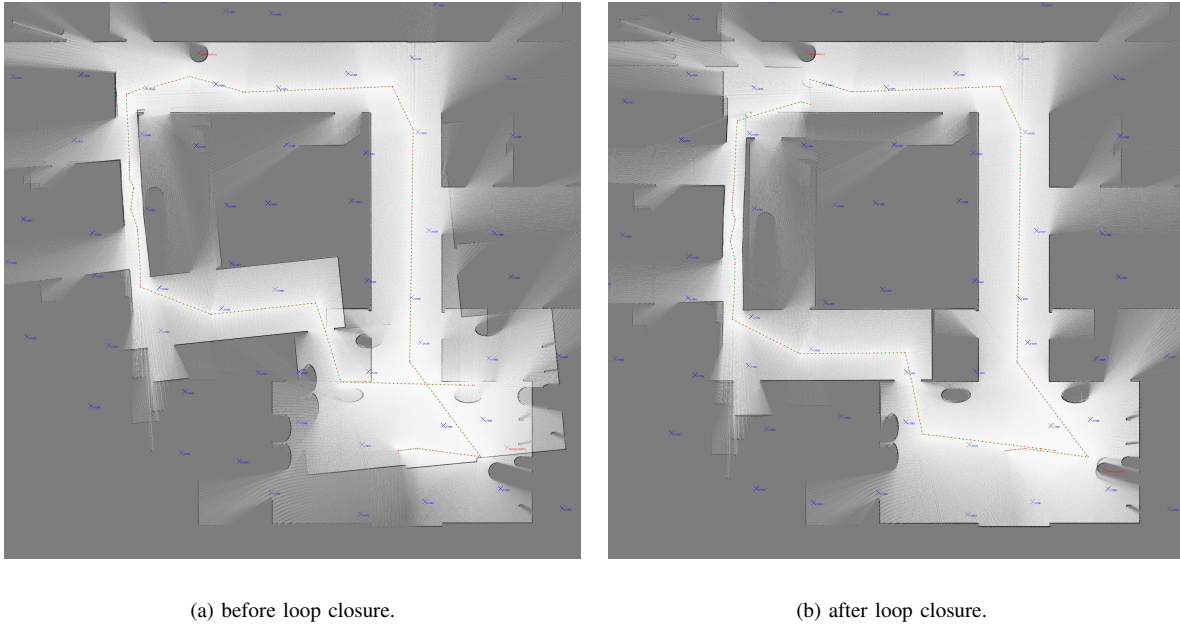


Fig. 3. An occupancy map generated by our mapping algorithm. In this example the robot drives a loop of 40 meters. The robot starts at the lobby at the bottom right and moves up. Then the robot turns left several times until it returns in the lobby. In Fig. 3(a) the lobby seems to be visible twice, once from the observations at the start of the loop and once from the observations at the end of the loop. During the loop small localization errors are accumulated, resulting in a final error of a few meters and several degrees. When the robot realizes that it is on a location visited before (by re-observing a particular landmark) it can close the loop by matching the measurements at positions in the graph.

fast and accurate in circumstances where dense measurements are available and the initial estimate can be trusted³.

The graph structure allows to have multiple disconnected maps in memory. In the context of SLAM for multiple robots, this allows to communicate the graphs and to have one disconnected map for each robot. Additionally, it is possible to start a new disconnected map when a robot loses track of its location, for example after falling down stairs.

When there seems to be an overlap between two disconnected maps, this hypothesis can be checked by scan matching. The displacement and correspondence between the measurements of two nearby points in each overlapping region are calculated. When the correspondence is good, a loop closing operation may be performed to refit all points on the two maps for improved accuracy. An example of a merged map is shown in [4].

The graph structure of the manifold can be easily converted into occupancy grids with standard rendering techniques, as demonstrated in [17] and Figure 3.

IV. MULTI-ROBOT RESCUE SITE EXPLORATION

Exploration addresses the challenge of directing a robot through an environment such that its knowledge about the environment is maximized [18]. A mobile robot typically maintains its knowledge about the external world in a map

m . Increasing the knowledge represented by m is achieved by either reducing the uncertainty about current information, or by inserting new information. The latter occurs when the map coverage is extended as the robot visits areas in the external world not yet covered by m before.

The approach in [4] was to passively acquire the information to store in the map, i.e. while the robot was wandering around pursuing other objectives like finding victims. In this work however, the focus is on *active* exploration with limited communication range: to explicitly plan the next exploration action a which will increase the knowledge about the world the most (taken into account if this knowledge can still reach the ComStation). In this paradigm victim finding becomes the side-effect of efficient exploration.

Occupancy grids [19] are a convenient representation for m in order to address the exploration challenge as they lend themselves excellently for storing probabilistic information about past observations. Each cell x corresponds to a region in the external world and holds the value $p(x)$ that denotes the aggregated probability that this region in the real world is “occupied”, i.e. is (part of) an obstacle. The objective of active exploration can then be seen as to minimize the *information entropy* $H(m)$ [20] of the probability distribution defined over all $x \in m$:

$$H(m) = - \sum_{x \in m} p(x) \log(p(x)) \quad (1)$$

Initially each grid cell has unknown occupancy, so $p(x) = 0.5$ for all $x \in m$ and the entropy of the map $H(m)$ is max-

³MatLab code of the Weighted Scan Matching algorithm was made available by Samuel T. Pfister *et al.*. C++ and Visual Basic versions are implemented by Max Pfingsthorn and Bayu A. Slamet and downloadable via <http://www.robocuprescue.org>.

imum. For exploration purposes the absolute value of $H(m)$ is not of interest, what is relevant is the difference in entropy before $H(m)$ and after $H(m|a)$ a particular exploration action a : the *information gain* $\Delta I(a)$ [21]–[23].

$$\Delta I(a) = H(m|a) - H(m) \quad (2)$$

Note that the exploration action a could be a complex maneuver, consisting of a number of controls u_i and observations z_i that spans multiple time steps i . Hence, for predictions about $\Delta I(a)$ that lie multiple timestamps in the future, the set of possible exploration actions can grow rather fast. In many current exploration strategies this issue is addressed by evaluating only a limited set of future states. These approaches consider only the situations where a robot actually enters yet unexplored area, which are by definition the locations where open area borders on unknown area: the *frontiers* [5].

Given the occupancy grid map m the concepts of open area and unexplored area are well-defined by the occupancy values $p(x)$ for all the grid cells $x \in m$ [5]. The unexplored area involves all the cells x for which the occupancy $p(x)$ is still at its initial value $p(x) = 0.5$. The free space are the cells for which $p(x)$ is sufficiently close to zero.

A. Estimating Area Beyond Frontiers on Occupancy Grids

A good autonomous exploration algorithm should navigate the robot to optimal target observation points [24]. The approach presented in [10] enables a robot to distinguish these locations using a method that is based on “safe regions”. The idea is that the robot simultaneously maintains two occupancy grids: one based on the maximum sensing range r_{\max} of the range sensing device and another one based on a more conservative safety distance r_{safe} . Typical values for r_{\max} and r_{safe} are 20 meters and 3 meters respectively. The result is that the safe region is in effect a subset of the open area as visualized in Figure 4(a). Frontiers can then be extracted on the boundaries of the safe region where the robot can enter the free space, as illustrated in Figure 4(b). Subsequently, Figure 4(c) illustrates how the area beyond the frontier can be estimated directly from the current map by measuring the amount of free space beyond the safe region.

Greedy exploration could continuously focus the robot to the frontier f with largest area $A(f)$ and which will ultimately lead to a complete coverage of the environment. More efficient exploration can be expected when also the distance $d(f)$ is considered in a utility function $U(f)$ that trades off the costs of moving to the frontier with the expected information gain. In our experiments we used the equation:

$$U(f) = A(f)P(f)/d(f) \quad (3)$$

The information gain $\Delta I(a_f)$ of the action a_f going to frontier f , is estimated by the product $A(f) * P(f)$. Here $P(f)$ is the probability that the robot can still communicate at the location of the frontier. How this probability is estimated is explained in the following section.

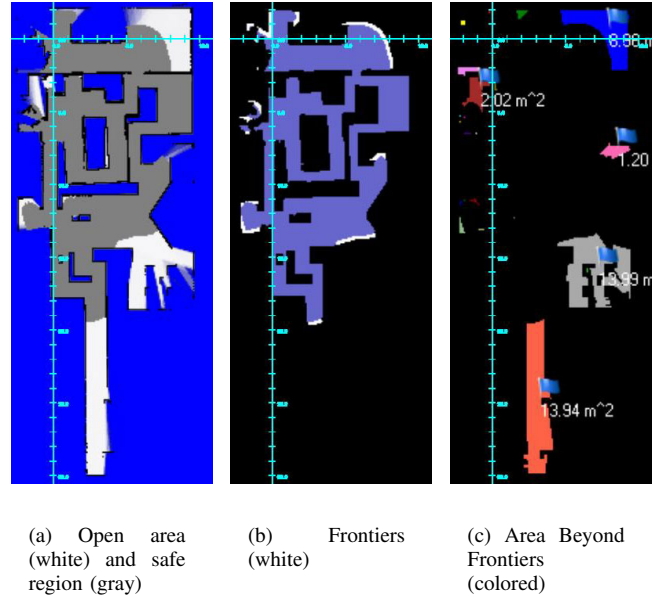


Fig. 4. Frontier Estimation: based on the open area and safe region in a) the frontiers in b) can be extracted. The regions beyond the frontiers can then be estimated directly from the map as in c).

B. Estimating the Success of Communication

During the exploration, the signal level of the communication towards the ComStation is measured at regular intervals. The location of the robot is provided by the localization, the position of ComStation is broadcasted. This means that at those regular intervals the tuple (distance, signal level) can be stored in a datastructure⁴. When the signal level of an unknown distance is needed, this can be looked up in the datastructure by finding the nearest neighbor. To find the nearest neighbor also a heuristic estimate of the signal level $S(d)$ is needed, which can be calculated from the current signal level $S(d_0)$ plus the distance component from the equation:

$$S(d)[dBm] = S(d_0)[dBm] - 10n \log\left(\frac{d}{d_0}\right) \quad (4)$$

This estimate only takes the dominant effect of path loss into account. Other effects, such as attenuation, reflection, refraction and diffraction of the signal in an indoor environment, are not taken explicitly into account. Instead, those effects (mainly dependent on the obstacles in the line of sight) are partly incorporated in the current signal level $S(d_0)$ and partly incorporated in the stored values in the datastructure. The attenuation of the current environment is “learned” in this datastructure. With the estimated signal level an estimate of the Bit Error Rate can be made, which can be translated in probability of a Message Error Rate. This Message Error Rate cuts down fast at a certain signal level (for these experiments fixed on -93 dBm).

⁴As datastructure a quadtree [25] for fast retrieval of nearest neighbors is used

C. Multi-Robot Coordination

After the frontier extraction method illustrated in Figure 4 and the utility function from Equation (3) we are left with the challenge to intelligently assign frontiers to the members of a multi-robot team.

Given the set of frontiers $F = \{f\}$ and team of robots $R = \{r\}$ the full utility matrix $U = [u_{ij}]$ can be computed that stores the utility u_{ij} for all possible assignment of robots $r_i \in R$ to frontiers $f_j \in F$. This matrix U is calculated with an Euclidian distance measure $d_{eu}(f)$. The Euclidean distance d_{eu} gives a lower bound of the actual distance to be traveled. This means that the utility values $[u_{ij}]$ are optimistic. The actual distance to be traveled can be calculated by performing a path-planning operation, but this is typically a computation intensive operation. The efficiency of our algorithm is optimized by recalculating only a few elements of the matrix; only the highest utility $u_{ij} \in U$ is recalculated. Thereafter it is checked if this value is still the maximum value. Otherwise the new maximum value is recalculated with a distance measure based on path-planning. When the path-planned value is the maximum, the frontier f_j is assigned to robot r_i and the rows and columns of the utility matrix U are pruned. This process continues until a frontier is assigned to the current robot r_c . The pseudo code of this algorithm is given in Algorithm 1.

Note that in some specific cases this could lead to a slightly sub-optimal assignment, i.e. a different non-greedily selected assignment could exist which has higher joint utility. Also, when there are less frontiers than robots some robots will not be assigned a frontier. However, in our experience these occasions are rather rare as robots usually quickly find more frontiers than they can close.

D. Planning Safe Paths

In the second part of the algorithm a check is made if an obstacle free path exists to a frontier. The same occupancy grid that was used to extract the frontiers can also be used to plan safe paths that avoid obstacles. If paths would be planned on the free-space robots may be guided to locations that are dangerously close to obstacles. This is a well-studied problem in robotics [26] and several solutions exist.

Because path-planning has to be performed for several robot-frontier combinations, a simple method has been applied that gives fast reasonable path. The occupancy map is converted into a safety map by convoluting the obstacles with the shape of the non-holonomic robot. This is approximated by employing a Gaussian convolution kernel. On this approximated safety map path-planning is performed with a breath-first algorithm.

This completes the set of tools necessary to enable coordinated frontier-based exploration by a team of robots. In the subsequent section the method will be applied to guide robots through to a virtual disaster site in several experiments.

V. EXPERIMENTS AND RESULTS

Our experiments will be based on the ‘‘Hotel Arena’’ that was used extensively during the RoboCup Rescue competi-

```

Data: the identity of the current robot  $r_c \in R$  and the
map  $m$  as known by  $r_c$ .
Data: the set of robots  $r_i$  in  $R$ . Each  $r_i$  consist of the
tuple  $(x_{r_i}, y_{r_i}, \theta_{r_i})$ .
Data: the set of frontiers  $f_j$  in  $F$ . Each  $f_j$  consist of the
tuple  $(x_{f_j}, y_{f_j}, A_{f_j}, P_{f_j})$ .
Result: the pair  $r_c, f_c$  and the path  $p_c$  to the location
 $(x_{f_c}, y_{f_c})$ 
for each robot  $r_i$  in  $R$  do
|   for each frontier  $f_j$  in  $F$  do
|   |    $d_{eu} = \sqrt{(x_{f_j} - x_{r_i})^2 + (y_{f_j} - y_{r_i})^2}$ ;
|   |    $u_{ij} = A_{f_j} P_{f_j} / d_{eu}$ ;
|   end
end
 $u_{max} = \max u_{ij}$ ;
repeat
|   for robot  $r_i$  and frontier  $f_j$  of  $u_{max}$  do
|   |    $p = \text{PathPlanning from } (x_{r_i}, y_{r_i}) \text{ to } (x_{f_j}, y_{f_j}) \text{ on}$ 
|   |    $\text{map } m$ ;
|   |    $d_{pp} = \text{length of path } p$ ;
|   |    $u_{ij} = A_{f_j} P_{f_j} / d_{pp}$ ;
|   end
|   if  $\max u_{ij} = u_{max}$  then
|   |   Assign  $f_j$  to  $r_i$ ;
|   |   Prune  $U$  from  $i$  and  $j$ ;
|   end
|    $u_{max} = \max u_{ij}$ ;
until robot  $r_c$  is Assigned ;
 $p_c = \text{last path } p$ 

```

Algorithm 1: The algorithm for the assignment of frontiers to robots

tions of 2006⁵. Figure 5 shows a blue-print of this office-like environment. The wide vertical corridor in the center connects the lobby in the south with several horizontal corridors that go east and west. Numerous rooms border on all the corridors. For the competition runs, robots would typically be spawned in the lobby or at the far ends of corridors, e.g. in the north-east, north-west or south-east corners.

Following the same setup as in the RoboCup competition each experiment will involve a run of 20 minutes. So the comparisons will focus on the amount of area that the robots were able to explore in this fixed time-window.

A. Results based on Coordinated Exploration

These experiments involve multi-robot exploration using the presented approach. We used three spawn positions in the Northern corridor: at both ends and at the T-junction near the middle. Each spawn position was used as the fixed location of the ComStation, with the robots at the two other spawn positions. The Northern corridor has a length of 25 meters,

⁵USARSim simulator, the simulated worlds used in these experiments and documentation are available on <http://www.sourceforge.net/projects/usarsim>. The maps used in 2007 are unfortunately not publicly available.



Fig. 5. A blue-print of the Hotel Arena, the virtual environment used for our experiments.

which allows error-free communication in this corridor. Communication problems occur in rooms and corridors further away.

The first experiment is used as reference scenario. The robots relay their positions to each other, but they do not share their maps. Their observations are only combined at the operator, and this information is not feed back to the robots. The resulting maps can be seen in Figure 6. On the maps the following color-coding is used:

- blue indicates unknown terrain,
- shades between light-blue and white indicate the probability that the area is free from obstacles
- black indicates obstacles
- solid grey indicates “safe region”, as introduced in Section IV-A
- red indicates a victim
- light-green indicates the path of the robots

In Figure 6(b) the robots started in the north-west and north-east end of the corridor. The ComStation was located at position N in the middle near the T-junction. In Figure 6(a) the ComStation was located in the north-west location NW and in Figure 6(c) in the north-east location NE. Although the resulting maps are already quite good (the results are comparable with the 2006 competition results, with 6 robots and no limit on the communication), it can be noticed that there is some inefficiency in the exploration. Some corridors are explored several times, which can be seen from the many green paths in those corridors.

In Figure 6(a) the first robot (called Hercules) starts exploring near the T-junction, while the second robot (called Achilles) starts at the NE location. Hercules heads west towards the ComStation. Near the ComStation Hercules turns back and follows the corridor to the east. That was precisely the point where Achilles started. Achilles goes south at the T-junction. In the lobby it discovers the door towards the north-west. During this passage communication is lost for short period, but is reestablished short afterwards. This is

a good indication of a typical communication range in this environment. In the mean time Hercules also reached the lobby, which means that the same corridor was explored twice.

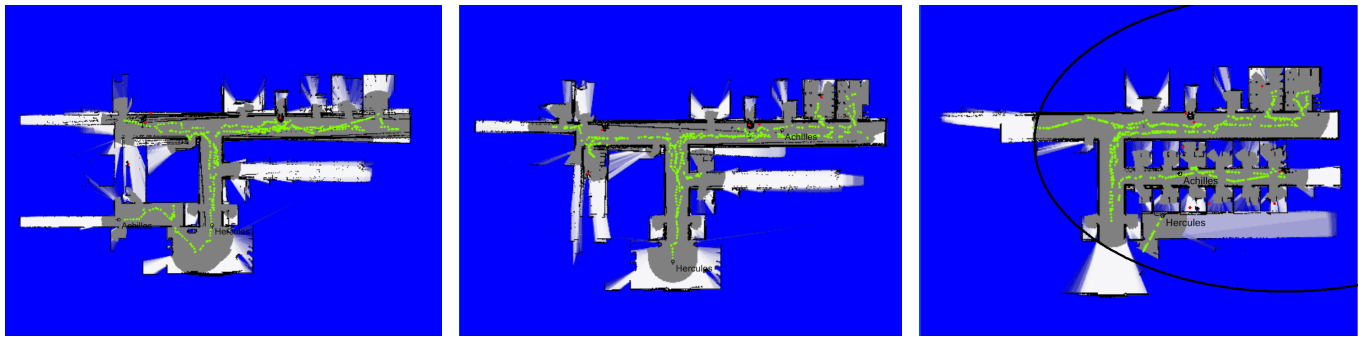
In Figure 6(b) the robot Hercules starts in the north-east. Hercules explores the first two rooms and heads towards the lobby. Achilles starts at the north-west, checks the room below and the small corridor to the west. After that Achilles heads towards the T-junction (where the ComStation is located), checks the first two rooms of the east-corridor but continues towards the north-east corridor (which was already explored extensively by Hercules).

In Figure 6(c) Hercules started near the T-junction. Hercules goes west (towards the ComStation). Hercules explores the two rooms near the ComStation and heads towards the lobby. In the lobby it discovers the passage to the north-east. For a long time the wireless connection is broken, but closer to the ComStation the connection is reestablished. More exploration is possible here, but the time limit of 20 minutes is reached.

It should be clear from these examples that the exploration choices of the robot are good, but not coordinated. Sometimes the communication link drops because the robots plan too far ahead, and cannot always predict what this means for the communication link. Reducing this planning horizon is maybe good to guarantee communication, but not for the exploration efficiency.

The next set of experiments demonstrates the effect of sharing map information between the robots. It should be clear that there will be less overlap between the areas explored by the robots. The expectation is that the robots will explore less corridors in the north and more corridors in the south. The resulting maps can be found in Figure 7.

In Figure 7(a) Hercules starts at the T-junction and directly heads towards the lobby. There the north-west passage is found. This information is relayed to the ComStation via Achilles. After the passage the robot gets stuck for a while, and heads towards the south-west corridor. Achilles explores the first two rooms and continues to explore the northern corridor.

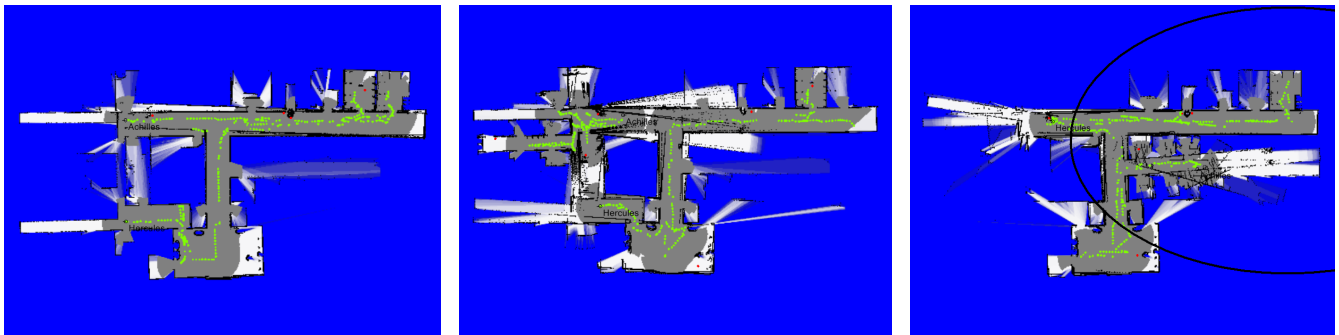


(a) ComStation NW (535 m^2 , 3 victims)

(b) ComStation N (556 m^2 , 4 victims)

(c) ComStation NE (545 m^2 , 6 victims)

Fig. 6. Exploration from the Northern corridor with two robots which do not share their map



(a) ComStation NW (531 m^2 , 3 victims)

(b) ComStation N (642 m^2 , 5 victims)

(c) ComStation NE (568 m^2 , 4 victims)

Fig. 7. Exploration from the Northern corridor with two robots sharing their map

In Figure 7(b) Achilles starts at the location NW. Achilles explores the narrow north-west corridor. After that Achilles is able to enter the cubicle area and to come back. Also the room south of the start location is partly explored. During all those narrow passages some orientation error is accumulated. Hercules in the mean time explores one room, continues towards the T-junction, heads towards the lobby and find the north-west passage.

In Figure 7(c) Achilles directly heads east. At the T-junction Achilles goes south, explores part of the lobby. The results of the lobby are relayed via Hercules. After the lobby Achilles goes back to the corridor in the east and finds 6 rooms. Hercules explores in the mean time the northern corridor and one of the rooms.

The difference between the two sets of experiments is twofold. Firstly, with shared maps less area is explored double, which can be seen from the green paths through the corridors. Further, more information collected in the southern regions, outside the direct communication range. A rough indication of the communication range is drawn in Figure 7(c) and 6(c). This is possible because the information reaches the operator via a relay (when the other robot is near). More important, the shared map-information forces the other robot to take

less obvious choices for observation points, because a portion of nearby frontiers is already explored by the other robot. Unfortunately, this also means that the local navigation of the robot is tested in narrow passages.

Notice not only the explored area but also the number of victims found by the two robots in the previous experiments is comparable to the number of victims found in the semi-final by teams with four to eight robots [3].

VI. DISCUSSION

This paper investigates the influence of a limited communication distance on a multi-robot exploration approach. The proposed algorithm incorporates wireless constraints in the selection between frontiers. Compared to other research, as for instance [27] and [24], the planning looks far ahead (both in time and space). In [27] also a utility function is used to select an optimal action, but there the action is a single move over an occupancy grid with cells of a few centimeters. In this paper planning looks several meters ahead.

Planning so far ahead makes it also difficult to estimate the probability of communication success, because the signal level indoors can drop sharply when traveling several meters ahead. Our hypothesis was that the emergent behavior of robots would

be to first explore the area inside the communication range, followed by an exploration outside the communication range. The experiments performed here indicate exploration outside the communication range occurs already before the whole area inside is explored, because the precise location of the communication range can only be learned by crossing that line several times (by accident). Further we had hoped that the exploration outside the communication range would occur for both robots at the same time, and that both robots could benefit from each other efforts. The limited number of experiments indicates that hope about the timing could be correct, but that this is no guarantee that both robots benefit from each other. It is clear that the role as relay station should be explicitly planned.

VII. CONCLUSIONS

This paper investigated the influence of a communication range on a frontier-based exploration approach that can be used to coordinate a team of robots. The approach assigns utilities to frontiers using a measure of the information gain that can be estimated directly during exploration. In this paper, an approach is proposed to incorporate a measure for the communication probability in the estimate of the information gain. This information gain is balanced by the movement costs. Subsequently, a frontier with the highest utility is assigned to the members of a robot team.

In a number of experiments the importance of reliable communication is indicated, and the need to include the communication probability inside the planning of multi-robot exploration. Further experiments are needed, but before these experiments are performed, the underlying navigation should be optimized.

In our experiments we have shown that this approach leads to efficient rescue site coverage. In future work we would like to investigate the possibilities for multi-robot coordinated exploration with more than two robots, study the influence of a-priori data and the effect of explicitly planning relay-positions to optimize exploration outside the communication range of the ComStation.

REFERENCES

- [1] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2003.
- [2] S. Balakirsky, C. Scrapper, S. Carpin, and M. Lewis, "Usarsim: providing a framework for multi-robot performance evaluation," in *Proceedings of PerMIS 2006*, August 2006, pp. 98–103. [Online]. Available: http://www.isd.mel.nist.gov/PerMIS_2006/proceedings/PerMIS_Program.pdf
- [3] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue," *Journal of Field Robotics*, to appear.
- [4] B. Slamet and M. Pfingsthorn, "ManifoldSLAM: a Multi-Agent Simultaneous Localization and Mapping System for the RoboCup Rescue Virtual Robots Competition," Master's thesis, Universiteit van Amsterdam, December 2006.
- [5] B. Yamauchi, "A frontier based approach for autonomous exploration," in *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 10-11, 1997., 1997.
- [6] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–378, 2005.
- [7] B. Yamauchi, "Frontier-based exploration using multiple robots," in *AGENTS '98: Proceedings of the second international conference on Autonomous agents*. New York, NY, USA: ACM Press, 1998, pp. 47–53.
- [8] R. Zlot, A. Stentz, M. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.
- [9] H. H. González-Baños and J.-C. Latombe, "Navigation Strategies for Exploring Indoor Environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002. [Online]. Available: <http://ijr.sagepub.com/cgi/content/abstract/21/10-11/829>
- [10] A. Visser, Xingrui-Ji, M. van Ittersum, and L. A. G. Jaime, "Beyond frontier exploration," in *Proceedings of the 11th RoboCup International Symposium*, July 2007. [Online]. Available: <http://www.science.uva.nl/~arnoud/publications/BeyondFrontiers.pdf>
- [11] D. Calisi, A. Censi, A. Farinelli, L. Iocchi, D. Nardi, and G. Tipaldi, "Spqr real rescue team description paper," in *Robocup 2006: Robot Soccer World Cup X*, Bremen, Germany, June 2006, (Proceedings CD-ROM).
- [12] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, and J. Wang, "Quantitative assessments of usarsim accuracy," in *PERMIS*, 2006.
- [13] J. Wang and S. Balakirsky, "Usarsim v 3.1.1," <http://sourceforge.net/projects/usarsim>, May 2007.
- [14] A. Howard, G. S. Sukhatme, and M. J. Matarić, "Multi-robot mapping using manifold representations," *Proceedings of the IEEE - Special Issue on Multi-robot Systems*, 2006.
- [15] F. Lu and E. Milius, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *Journal of Intelligent and Robotic Systems*, vol. 18, pp. 249–275, 1997.
- [16] S. T. Pfister, K. L. Kriechbaum, S. I. Roumeliotis, and J. W. Burdick, "A weighted range sensor matching algorithm for mobile robot displacement estimation," *IEEE Transactions on Robotics*, 2007, (to appear).
- [17] M. Pfingsthorn, B. Slamet, and A. Visser, "A scalable hybrid multi-robot slam method for highly detailed maps," in *Proceedings of the 11th RoboCup International Symposium*, July 2007. [Online]. Available: http://www.science.uva.nl/~arnoud/publications/slam_method.pdf
- [18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [19] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, pp. 61–74, 1988.
- [20] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, March 1998.
- [21] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation - mobile robot navigation with uncertainty in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 34–40.
- [22] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005.
- [23] R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *AAAI/IAAI*, 2000, pp. 852–858.
- [24] T. Kollar and N. Roy, "Trajectory Optimization using Reinforcement Learning for Map Exploration," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008. [Online]. Available: <http://ijr.sagepub.com/cgi/content/abstract/27/2/175>
- [25] H. Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 187–260, 1984.
- [26] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.
- [27] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.