Autonomous Multi-Robot Exploration in Communication-Limited Environments

Julian de Hoog, Stephen Cameron and Arnoud Visser

Abstract— **Teams of communicating robots are likely to be used for a wide range of applications in the near future, such as robotic search and rescue or robotic exploration of hostile and remote environments. In such scenarios, environments are likely to contain significant interference and multi-robot systems must be able to cope with loss of communication. We propose a novel multi-robot exploration approach, *role-based exploration*, in which members of the team explicitly plan to explore beyond communication range limits. Rendezvous points are calculated carefully to improve the exploration efficiency. A dynamic team hierarchy leads to additional gains. The result is a hybrid centralised/distributed behaviour that adjusts reactively to communication availability and environment size.**

I. INTRODUCTION

As technologies improve and miniaturise, the number of likely applications for robots are increasing. Team of robots are already used for a variety of tasks, and will likely find more extensive use in the near future. Such applications include surveillance, target tracking, environmental monitoring, reconnaissance, as well as various domestic uses. Of particular relevance to this paper are two tasks: multi-robot exploration and robotic search and rescue.

In the multi-robot exploration task, teams of robots may be used for exploration and mapping of previously unknown environments. Such environments may include dangerous areas such as war and disaster zones, or remote areas such as underwater or other planets. In the robotic search-and-rescue task, robots may be used to enter and explore environments after disasters, such as earthquakes. The goal here is to find victims and relay useful information, such as possible entry routes or areas of toxicity, to human responders.

There are many challenges involved in such efforts, the most common ones being navigation, simultaneous localisation and mapping, multi-robot coordination, and communication. Recently there has been much work in each of these domains, and many promising approaches can accurately map small environments such as interiors of buildings. Much work remains to be done however if complex environments such as disaster zones are to be explored.

A particularly significant challenge is that of communication. While wireless communication between robots is now commonplace, environments of interest are unlikely to contain any sort of communication infrastructure, and complex environments are likely to contain significant interference. Potential communication drop-out and failure must be taken into account.

In this paper we present an overview of our research to date regarding this problem and describe our proposed solution, "Role-Based Exploration". The rest of this paper is organised as follows: In section II we discuss related work. Section III details our basic approach. Sections IV and V detail two major improvements we make to the basic approach, involving calculation of improved rendezvous points and dynamism in the team hierarchy, respectively. We describe our custom-built simulator and present simulation results in section VI. Finally we discuss the implications of our work in VII and conclude in VIII.

II. RELATED WORK

Multi-robot exploration has received considerable attention in recent years but only a small number of approaches have taken limited communication into account.

In early approaches, a line-of-sight constraint was used to keep robots within communication range [1], [9]. This has been extended to robots reactively choosing a direction that will most likely keep them within sight of the rest of the team [14].

Several authors propose multi-robot exploration strategies based on market principles, in which robots place bids on subtasks of the exploration effort [18], [5], [24], [17]. These bids are typically based on values such as expected information gain and travel cost to a particular location in the environment, and may be assigned in a distributed fashion among team members, or by a central agent. When strength of communication is factored into the bids, robots avoid areas outside of communication range.

Another common strategy for robotic exploration is to use frontiers [23], which can easily be extended for use by multiple robots [3], [6], [15], [20]. Similar to bids described above, utilities of individual frontiers may include a factor related to likelihood of communication success, so robots are less likely to explore areas that take them out of the team communication range.

Further approaches include the use of 'energy fundamentals' to maintain network connectivity [16], results from graph theory to keep individual robots in 'comfort zones' [19] and the application of synthetic 'spring forces' to keep robots close to one another [12].

While several of these approaches have proven successful in maintaining team connectivity during the exploration effort, they are usually limited by the constraint of having to keep team members within communication range. Even

J. de Hoog and S. Cameron are with the Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, UK. {julian.dehoog, stephen.cameron} @comlab.ox.ac.uk

A. Visser is with the Intelligent Systems Laboratory, University of Amsterdam, Science Park 107, NL 1098 XG Amsterdam, Netherlands. a.visser@uva.nl

if members of a team are dispersed to the maximum extent that their communication ranges allow, in large and complex environments unexplored areas will remain.

A solution to this problem is to allow robots to autonomously explore beyond communication range limits. This can be implemented in terms of 'robot pack' or clustering behaviour, in which groups of robots stay close together as they explore the environment [15], [17], [6].

However, little work has been done towards the typical search-and-rescue problem of gathering information in a severely communication-limited environment at a single location as efficiently as possible.

III. ROLE-BASED EXPLORATION

A. Problem Description

The problem that we are particularly interested in is the consolidation of the knowledge of all robot team members at a single location. In a search-and-rescue scenario this corresponds to human responders' point of entry, while in reconnaissance or surveillance this corresponds to the base station where information is gathered and analysed. We assume no prior knowledge of the environment.

Given recent developments in robotics and simultaneous localisation and mapping (SLAM), we make three further assumptions:

1) The robots are equipped with a SLAM module. This may be optical or sonar, but would more likely involve laser range-finder data. Laser range-finders are now very common on ground-based platforms and have recently been demonstrated on a UAV [7].
2) This SLAM module provides reasonably accurate localisation. Recent approaches such as scan-matching [13] or particle filters [6] make this a realistic assumption. Localisation does not need to be perfect and there is some room for error. However, robots need to be able to find their way to within communication range of agreed rendezvous points.
3) Maps created by the SLAM module keep track of explored, free space. In our work we assume occupancy-grid based maps, but the approach could be tailored to topological maps as well. The notion of free space is essential for calculation of rendezvous points.

Our main goals are to (i) explore the environment as efficiently as possible; (ii) relay new information to the base station as quickly and as often as possible; and (iii) minimise the time that team members spend out of range of the base station. This must be achieved without placing an unrealistic burden on team communication systems.

B. The Basic Approach

In role-based exploration each member of the team is assigned one of two roles:

1) *Explorer*. Explorers are meant to explore the farthest reaches of the environment. To communicate their findings, they return periodically to previously agreed rendezvous points where they pass their knowledge to a relay.
2) *Relay*. Relays ferry information back and forth between explorers and the command centre. This is achieved by meeting the explorer periodically at aforementioned rendezvous points, exchanging all relevant knowledge, and then returning to the command centre. If a relay discovers information about the environment while relaying, this is added to the team knowledge, but exploration is only a by-product of the relay's movement.

The team hierarchy is a tree with a robot at every node; the base station is the tree's root and explorers are the tree's leaves. The tree may have arbitrary depth, *i.e.* there may be a chain of multiple relays between the base station and an explorer. Currently we use a branching factor of 1 only (other than for the root, which may have any number of children), but we hope to experiment with higher branching factors in the future.

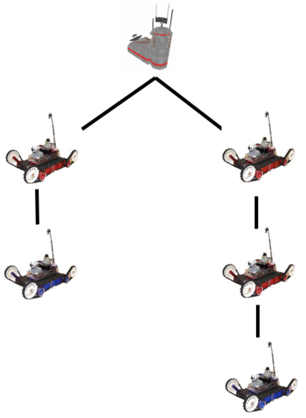


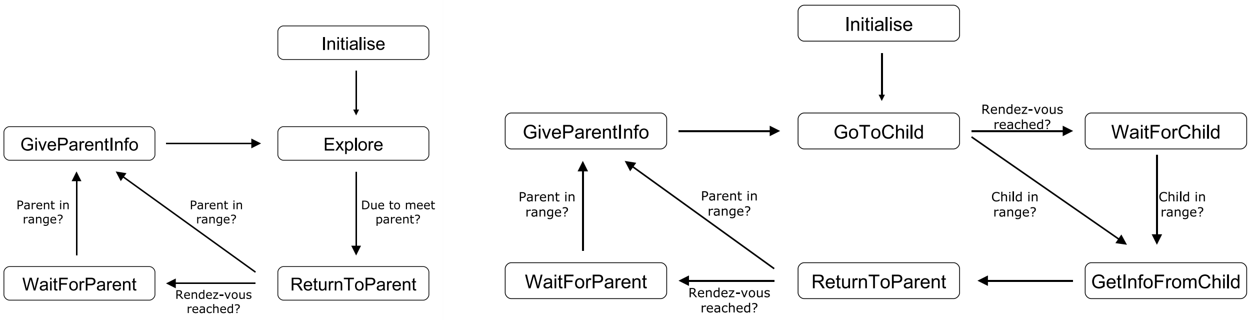
Fig. 1: A possible hierarchy for role-based exploration. The base station (top) is the root of the hierarchy tree, explorers (blue) are leaves, and there may be one or more relays (red) in a branch.

Such a configuration means that new information gathered in the environment by explorers is propagated up the tree via intermediate relays. New information gathered in parallel branches, along with control commands, can be sent down the tree from base station to explorers via the same relays.

If all robots are within range of one another, this is performed via multi-hop communication. If unexplored areas remain beyond communication range limits, the communication chain may be broken: explorers explore the far reaches of the environment, and relay robots become mobile relays, ferrying information back and forth between explorers and the base station. In short, the team responds reactively to the size of the environment and available communication levels.

State transition diagrams for Explorers and Relays are presented in Figure 2. An Explorer and Relay do not need to reach rendezvous to transition to the next state. If there is a chance meeting between the two earlier than expected, it is advantageous to replan at that moment, rather than wait until both reach the agreed rendezvous location.

Note that this approach is both centralised and distributed: both explorers and relays behave autonomously and, aside



(a) Explorers

(b) Relays

Fig. 2: State transition diagrams

from needing to share information with their parent and child, do not rely on a global team strategy. At the same time, control commands may be issued top-down from the base station that may override individual robots' behaviours. For example, if an environment is no longer of interest in a search-and-rescue effort, commands from a base station, distributed to all team members via relays, could lead to a pull-out of the whole team.

C. Frontier Assignment

Assuming that the team hierarchy has been determined and each robot assigned a role, how does exploration actually take place? For this, we apply simple frontier exploration [23], which is among the most popular and promising approaches today. Frontier exploration is heavily influenced by how utilities are calculated for individual frontiers. For every frontier f we calculate a utility $U(f)$ as follows:

$$U(f) = A(f)/C^n(f)$$

where $A(f)$ is the area of frontier f, $C(f)$ is the path cost from the robot to that frontier, and exponent n determines the exploration behaviour. High values of n lead to exploration of nearby frontiers (such as rooms) whereas low values mean that robots are more likely to pursue larger frontiers (such as hallways) [21]. For experiments reported later in this paper we use $n = 2$.

An additional consideration is that it is undesirable to send two robots into the same frontier. Elsewhere segmentation and the Hungarian method have been proposed [22], but we use a simple agent-frontier assignment algorithm detailed in [21]; in short, every robot determines frontier utilities for itself and its nearby teammates, and iteratively calculates a robot to frontier assignment that maximises joint utility. While this method is not necessarily optimal, it is fast, and in our experience entirely sufficient for distributed exploration.

D. Teammate Modeling

When two teammates meet, they exchange all relevant knowledge of the environment. After exchange, each robot will have the same map, and know exactly what its teammate knows at that point in time. Since relays' movement is highly predictable and both robots use the same path planner, an explorer can calculate exactly how long a relay will need to return to the base station (or its parent relay), turn around, and make its way back to the next jointly agreed rendezvous point. Thus the explorer knows exactly how much time it has to continue exploring before having to turn around and rendezvous once again, and subsequent meetings can be timed in such a manner that neither relay nor explorer waste time waiting for the other to return to the rendezvous point – both should reach the rendezvous point at almost the same time.

Moreover, if the explorer stores the map exchanged at rendezvous separately from its own evolving map, then it can at any point predict the relay's likely position, even when not in communication range (since the relay's map is unlikely to change much). Explorer and relay can also agree on fallback rendezvous points, in case the preferred rendezvous point can unexpectedly not be reached. This has significant implications for rendezvous in dynamic environments, discussed in more detail in section VII.

IV. RENDEZVOUS POINT CALCULATION

Role-based exploration is heavily reliant on robots periodically meeting one another for information exchange. In our early work, new rendezvous points were set by an explorer at the moment it turned around to meet a parent relay; in other words, rendezvous locations were equivalent to (usually) the most forward point reached in the environment. In some situations this led to poor performance, such as when a rendezvous point was chosen behind a door, or in the far corner of a room.

We now use a much improved approach: subsequent rendezvous is calculated by the explorer while it is in communication range of the relay, and uses thinning on the free space in the map. Thinning is a technique from digital image processing that is meant to reduce a shape to its skeleton by making the shape as thin as possible while keeping it connected and centred (there are many parallels between thinning, skeletonisation, and Voronoi diagrams). A wide range of thinning techniques have been proposed since the 1960's, having various advantages or disadvantages (for a review, see [11]).

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Fig. 3: Traversal function $T(p_1)$ is the number of 0,1 patterns in the sequence $p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_2$

In our approach we use Hilditch's algorithm [8], since it is fast, returns a connected skeleton, and is easy to implement. A typical skeleton calculated using Hilditch's algorithm is presented in the bottom right inset of Figure 7.

Hilditch's algorithm requires the calculation of a *neighbour traversal* function $T(p_1)$, described in Figure 3. This function can also be used to find junction points in the skeleton: any point p_1 that is a junction in the skeleton will have $T(p_1) \geq 3$. A skeleton may contain long stretches without junction points, for example along a hallway – to fill out the resulting graph, we iterate over all points in the skeleton and add those that are a minimum distance from all existing rendezvous points (*filling*). On the other hand, complex parts of the environment may contain a large number of junction points in a small area – to simplify calculations we choose only one point per given density (*pruning*). This gives a nice set of possible rendezvous points, distributed fairly evenly over the known environment and including all junctions. The full algorithm for rendezvous point calculation is presented in Algorithm 1.

```
List skeletonPoints = hilditchThinning(map);
List rendezvousPoints = new List;
foreach sp $\in$ skeletonPoints do
    if neighbourTraversal(sp) $\geq$ 3 then
        rendezvousPoints.add(sp);
    end
end
foreach sp $\in$ skeletonPoints do
    boolean addToList = true;
    foreach rp $\in$ rendezvousPoints do
        if sp.distanceTo(rp) ¡ threshold $T_1$ then
            addToList = false;
            break;
        end
    end
    if addToList then
        rendezvousPoints.add(sp);
    end
end
foreach rp1 $\in$ rendezvousPoints do
    foreach rp2 $\in$ rendezvousPoints, rp2 $\neq$ rp1 do
        if rp1.distanceTo(rp2) ¡ threshold $T_2$ then
            rendezvousPoints.remove(rp1);
        end
    end
end
 Return rendezvousPoints;
```

Algorithm 1: Calculation of rendezvous points.

Now that we have a list of potential rendezvous points, which is the best one? We examined a number of different utilities and combinations thereof: estimated communication range at the rendezvous point, proximity to nearest frontiers, and path cost. Since we want the relay to follow the explorer, however, it turned out that the most important consideration is the explorer's next choice of frontier. In other words, placing the next rendezvous inside the next frontier that the explorer plans to enter, while ensuring that the rendezvous point has a strong communication range, gave the best results. (A large communication range is a desirable characteristic for a rendezvous point since as two robots approach it, they will be able to detect and communicate with one another earlier. Communication range at a particular point can be easily estimated using the communication model described in section VI-A).

More specifically, in our implementation we choose a rendezvous point by considering only a small number of points near the explorer's next frontier of choice and choosing the one having highest neighbourTraversal value (since this is the most important junction). If multiple points have equal neighbourTraversal values, we choose the one with the best estimated communication range.

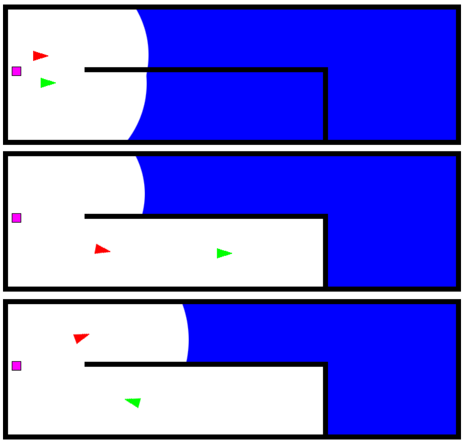


Fig. 4: A simple example of a situation in which a dynamic hierarchy is desirable. Initially, the explorer (green) sets out to explore and the relay (red) follows (top). Fairly quickly the end of the bottom hallway is reached and both must turn around to explore the top hallway (middle). At this point, the relay is much closer to the new frontier than the explorer, so it makes sense for the two of them to trade roles (bottom). The red agent is now an explorer and the green agent is now a relay.

V. DYNAMIC HIERARCHY

Even with the novel rendezvous point calculation, there are still certain scenarios where improved performance can be achieved. For example in an environment with a loop, an explorer may reach the base station (after completing the loop) and must then return back into the environment to rendezvous with its parent relay. At that point, the relay is deeper in the environment than the explorer, so the two should switch roles. Another simple example is provided in figure 4, and there are many other, similar situations where it is advantageous to have a dynamic team hierarchy.

To allow for dynamic role swapping in the team hierarchy, we have decided on a single, simple rule, the "role swap rule". Consider two robots A and B, each having goals G_A and G_B, respectively. Let $\gamma(u, v)$ represent the path cost from location u to location v in a given map. When u and v are known, this value is easy to calculate using standard path planners (such as A*) on the map. Suppose A and B have encountered one another and established a communication link. If

$$max(\gamma(A, G_A), \gamma(B, G_B)) > max(\gamma(A, G_B), \gamma(B, G_A))$$

then let A assume B's role, state, and location in the tree, and let B assume A's role, state, and location in the tree. For purposes of visualisation, this basically means that the longest path among the four paths computed is eliminated, so the robots travel shorter distances to reach the destinations of interest. The rule is applied equally to relays and explorers, both within the same branch and across branches of the hierarchy tree.

VI. EVALUATION

A. Simulator

To evaluate role-based exploration at various stages of development and to compare it to existing methods, we have developed our own JAVA-based simulator, the Multi-Robot Exploration Simulator (MRESim)[1]. MRESim allows for custom configuration of environments, either manually or by import of binary image.

The simulation framework handles collisions, sensor data and communication as follows: At every time step, the simulation framework requests from each agent a new desired location. If the location is valid, the agent is moved to this location, and new sensor data is simulated and sent to the agent. Following the movement of all agents, the communication model is used to determine whether any agents are within range of one another, either directly or via multi-hop. If yes, all relevant knowledge of the environment is shared between all communicating agents.

At any point a simulation may be paused and agents individual knowledge bases may be examined. This includes all known free space, safe space, frontiers, calculated paths, communication ranges, map skeleton, rendezvous points, robot's role and state, and team hierarchy.

We have also implemented and tested a variety of communication models in our simulations. For experiments reported here we use a standard path loss model with a wall attenuation factor as described in [2]:

$$S = P_{d_0} - 10 \times N \times log_{10}(\frac{d_m}{d_0}) \begin{cases} nW \times WAF & nW < C \\ C \times WAF & nW \geq C \end{cases}$$

where P_{d_0} is the reference signal strength, N is the path loss rate, d_m is the distance, d_0 is the reference distance, nW is the number of obstructing walls, WAF is the wall attenuation factor and C is the maximum number of walls to consider. This model is widely used in simulation, including the popular USARSim simulator [4]. A typical communication range for an agent is displayed in figure 7.

B. Results

While we cannot reproduce all of our results here due to lack of space, we present a representative set that both compares role-based exploration to existing approaches and examines the improvement a dynamic hierarchy makes as compared with a static hierarchy. To do this we compare three exploration approaches:

A) Greedy frontier-based exploration, where frontiers are chosen based on a utility function that takes into account information gain and path cost [20]. This approach is similar to those used in [3], [6], [15].

B) Role-based exploration as described above with a static team hierarchy.

C) Role-based exploration with a dynamic team hierarchy, using the role swap rule described in section V.

Experiments were conducted with a variety of team sizes and in a variety of environments. Here we present results that we believe to be representative of most of our experiments. As an environment, we used the vasche_library_floor1 floor plan from the Radish data set[2]. For each of the approaches, 10 robots were used. Both the static and dynamic role-based approaches used a hierarchy that contained 5 pairs of robots, *i.e.* a branching factor of five at the root, and one relay for each explorer. Figure 5 shows the full results of this run, and a screenshot is provided in figure 7.

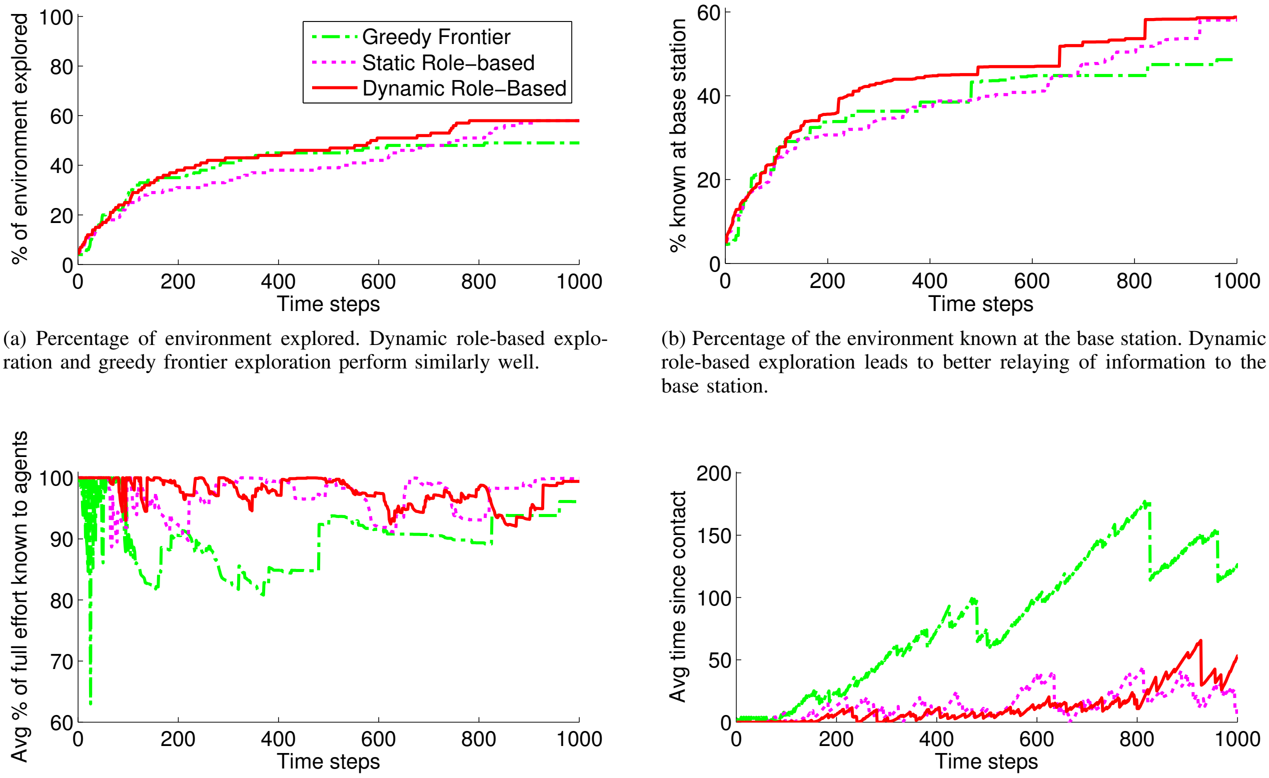
Dynamic role-based exploration leads to faster coverage of the environment than greedy frontier-based exploration, in spite of the fact that only half as many robots are actively exploring (the other half are relays). This is due to the fact that poor inter-team awareness in greedy frontier-based exploration means that robots are likely to cover areas that have already been explored.

Dynamic role-based exploration also outperforms static role-based exploration in every metric. It leads to faster exploration (figure 5a), greater awareness of the exploration at the base station (figure 5b), greater inter-teammate awareness (figure 5c) and quicker responsiveness to the base station (figure 5d). Only late in the experiment does connectivity to base station seem weaker, but this is mainly due to the fact that more of the environment has been discovered and robots must travel longer distances.

Overall, dynamic role-based exploration leads only to a small improvement in terms of speed of exploration. The main gains, however, are inter-robot awareness and team responsiveness. For applications such as search and rescue, where instant control over the robots is highly desirable, this is an important characteristic.

[1] MRESim is available upon request from the authors.

[2] This data set was obtained from the Robotics Data Set Repository (Radish) [10]. Thanks go to Ashley Tews for providing this data.



(a) Percentage of environment explored. Dynamic role-based exploration and greedy frontier exploration perform similarly well.

(b) Percentage of the environment known at the base station. Dynamic role-based exploration leads to better relaying of information to the base station.

(c) Average percentage of currently known space known to all team members, in other words ability of the team to share information among teammates.

(d) Average time, over all agents, since last contact with the base station, either directly or via a mobile relay.

Fig. 5: Simulation results: a comparison of greedy frontier-based exploration, static role-based exploration, and dynamic role-based exploration.

VII. DISCUSSION AND FUTURE WORK

The main advantages and disadvantages of role-based exploration can be summarised as follows:

Advantages: There is no need for an exact communication model. Explorers and relays adjust to size of the environment and communication availability reactively. Provided sufficient power is available, the approach leads to full exploration of environments regardless of how much interference or how short communication ranges are. Equal numbers of robots lead to similar exploration, but considerably better teammate awareness and team connectivity.

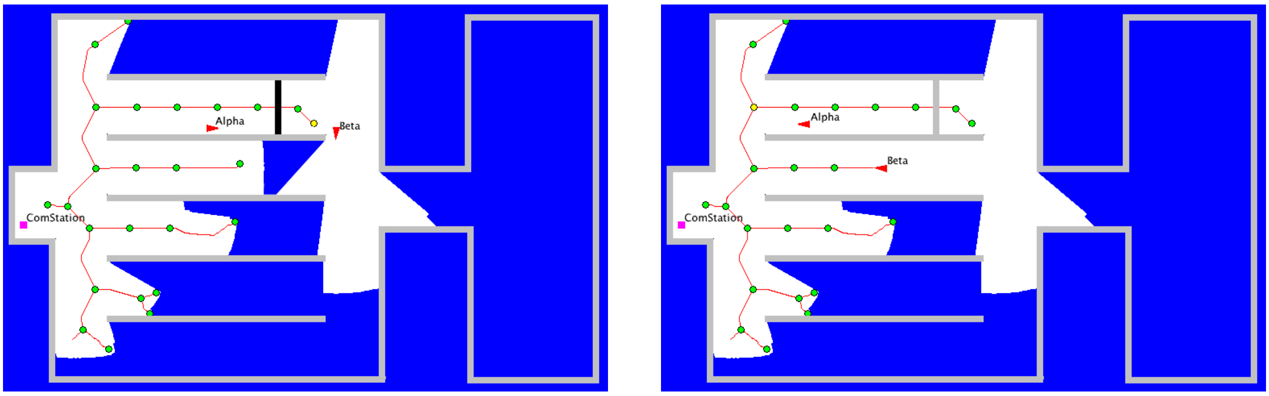
Disadvantages: Since individual robots or groups of robots may be out of range of the base station, control over the full team may not be instantaneous. If a robot or a group of robots becomes lost or incapacitated, this information may not reach the base station (other than by lack of response). The approach is heavily reliant on reasonably accurate mapping and localisation.

We envision numerous possible extensions to the current approach:

Dynamic Hierarchy Structure: In our current approach, even with dynamic role swaps within the team hierarchy, the overall structure and depth of the hierarchy tree does not change. In certain scenarios (e.g. environments with long hallways) it may be desirable to lengthen and shorten branches in the tree as required. We hope to expand ideas presented here and look at a wider range of options to enable effective team hierarchy structural adjustments.

Replanning in Dynamic Environments: The set of possible rendezvous points is shared by a relay and explorer when they meet. In the applications we are interested in, there is a risk of dynamic environments. For example, in a search-and-rescue scenario, an unstable beam may fall, a roof may collapse, or rubble may shift or burn as the exploration effort is ongoing. Since relay and explorer will have a shared set of rendezvous points, this means that they may recalculate to find one another at a different location if a previously agreed rendezvous point becomes unexpectedly blocked. This can be done either as a recalculation on the shared map, or by storing one or more 'backup' rendezvous points that may be used if the primary rendezvous cannot be reached. An example is provided in figure 6.

Heterogeneous teams: The current implementation does not take into account potential heterogeneity in the team. It is possible that different types of robots with different sensor loads may be involved in the same effort, in which case it may be desirable for certain types of robots to play particular roles (e.g. relays could be fast, simple robots while explorers could carry more intricate sensors). In such a scenario, the role swap rule would need to be adjusted to take robot types



(a) A sudden wall (previously not there) blocks Alpha from reaching the rendezvous point (yellow). Beta waits a specific amount of time for Alpha to appear.

(b) Alpha recalculates, and chooses the rendezvous point having highest degree (a junction point). Beta recalculates, assumes Alpha will head for the same junction point, and pursues frontier exploration with a preference for frontiers near the new rendezvous point. Eventually the two find one another again.

Fig. 6: Use of rendezvous points to deal with unexpected obstacles in dynamic environments

and their ideal roles into account.

Extension to three dimensions: Work to date has focussed on two-dimensional environments, but we do evaluate every aspect of the approach with an eye towards possible extension to three dimensions. Potential bottlenecks in the calculations include 3D path planning and 3D skeletonisation (for calculation of rendezvous points). However, we hope that advances in each of these areas, along with recent rapid advances in 3D mapping and localisation, will allow for dynamic role-based exploration or some variant thereof to be applied to both ground- and air-based robot systems in the future.

VIII. Conclusions

In this paper we presented a novel approach to multi-robot exploration in which exploration beyond communication range limits is explicitly planned for. Rendezvous point calculation using thinning on the map, along with a dynamic role swap rule, lead to improved performance of the exploration algorithm. While dynamic role-based exploration does not lead to vastly faster exploration than currently popular utility and frontier based approaches, it has other important advantages. Using the same number of robots, similar exploration can be achieved while maintaining considerably better team connectivity. In large and communication-challenged environments, this is very helpful.

Every year robots are becoming smaller, more powerful and more intelligent, and in 5 to 10 years, teams of small rolling, crawling or flying robots are likely to be used for exploration of unknown terrain and for robotic search and rescue. Autonomous exploration beyond team communication range limits remains a young field of study. It is hoped that the ideas explored in this paper provide an early contribution to coordination of future multi-robot systems.

References

[1] R. Arkin and J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. *Advanced Motion Control, 2002. 7th International Workshop on*, pages 455–461, 2002.

[2] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. *IEEE Infocom 2000, Tel-Aviv, Israel*, 2:775–784, March 2000.

[3] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376 – 378, 2005.

[4] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Usarsim: a robot simulator for research and education. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1400–1405, April 2007.

[5] M. B. Dias and A. T. Stentz. A free market architecture for distributed control of a multirobot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, July 2000.

[6] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339, July 2006.

[7] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[8] C. Hilditch. Linear skeletons from square cupboards. *Machine Intelligence*, 4:403–420, 1969.

[9] A. Howard, M. Mataric, and G. Sukhatme. An incremental deployment algorithm for mobile robot teams. *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, 3:2849–2854 vol.3, 2002.

[10] A. Howard and N. Roy. The robotics data set repository (radish), 2003.

[11] L. Lam, S. Lee, and C. Suen. Thinning methodologies: A comprehensive survey. 14(9):869–885, September 1992.

[12] A. Mosteo, L. Montano, and M. Lagoudakis. Multi-robot routing under limited communication range. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1531–1536, May 2008.

[13] 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.

[14] M. Powers and T. Balch. Value-based communication preservation for mobile robots. In *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[15] M. N. Rooker and A. Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.

[16] B. Santos Pimentel and M. Fernando Montenegro Campos. Multi-robot exploration with limited-range communication. In *Anais do XIV Congresso Brasileiro de Automtica (CBA'02)*, 2002.

[17] W. Sheng, Q. Yang, J. Tan, and N. Xi. Distributed multi-robot coordination in area exploration. *Robot. Auton. Syst.*, 54(12):945 – 955, 2006.

[18] R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. L. S. Younes. Coordination for multi-robot ex-

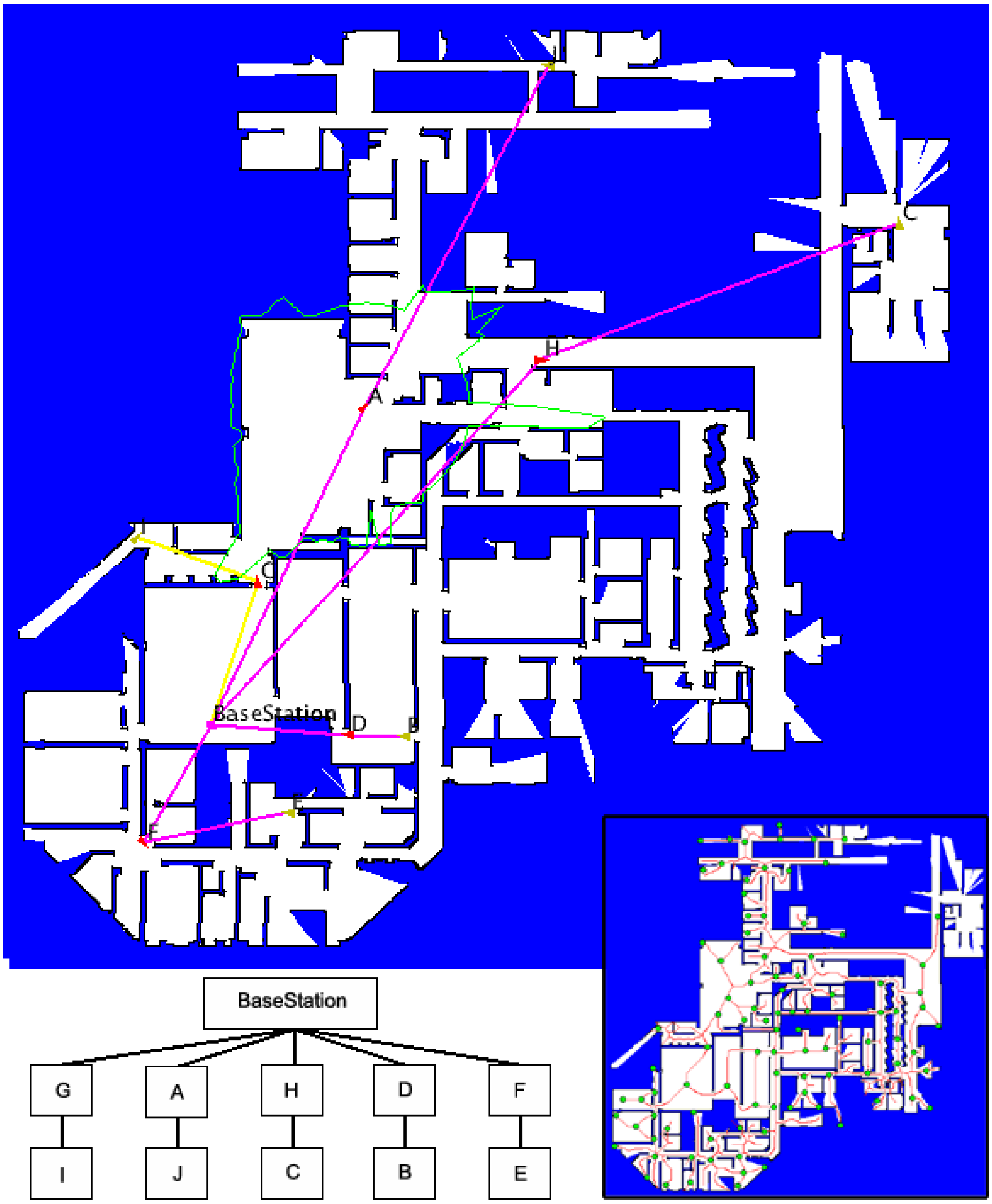


Fig. 7: A screenshot of dynamic role-based exploration in our MRESim simulator, after 1000 time steps. Top: purple and yellow lines indicate team hierarchy, with yellow connections indicating current communication link. An example communication range has been presented for robot A (thin green polygon). Bottom left: the team hierarchy as a tree. Bottom right: an example skeletonisation of the map, as performed by robot J. Green dots indicate possible rendezvous locations.

ploration and mapping. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 852–858. AAAI Press / The MIT Press, 2000.

[19] J. Vazquez and C. Malcolm. Distributed multirobot exploration maintaining a mobile network. *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, 3:113–118 Vol.3, June 2004.

[20] A. Visser and B. Slamet. Including communication success in the estimation of information gain for multi-robot exploration. *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pages 680–687, April 2008.

[21] A. Visser and B. A. Slamet. Balancing the Information Gain Against the Movement Cost for Multi-robot Frontier Exploration. In *European Robotics Symposium 2008*, Springer Tracts in Advanced Robotics, pages 43–52. Springer-Verlag, February 2008.

[22] K. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1160–1165, Sept. 2008.

[23] B. Yamauchi. Frontier-based exploration using multiple robots. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 47 – 53, New York, NY, USA, 1998. ACM.

[24] R. Zlot, A. Stentz, M. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 3:3016 – 3023, 2002.