ARL-TR-7423 • Sep 2015

US Army Research Laboratory

Experiments Toward the Application of Multi-Robot Systems to Disaster-Relief Scenarios

by Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-7423 • SEP 2015

US Army Research Laboratory

Experiments Toward the Application of Multi-Robot Systems to Disaster-Relief Scenarios

by Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2015	Final	July – October 2014

4. TITLE AND SUBTITLE
Experiments Toward the Application of Multi-Robot Systems to Disaster-Relief Scenarios

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-CII-A
Adelphi, MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-7423

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We present a formal problem statement for the application of multi-robot systems to intelligence gathering in disaster-relief scenarios. The design and implementation of a system capable of performing experiments in a relevant field environment is addressed along with a suite of autonomy-enabled behaviors that support operation in a communications-limited setting. Finally, we present extensive experimental results evaluating this system.

15. SUBJECT TERMS
autonomous systems, intelligent behaviors, task-level autonomy, disaster relief

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jason Gregory
Unclassified	Unclassified	Unclassified	UU	54	19b. TELEPHONE NUMBER (Include area code) 301-394-5628

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Intentionally left blank.

1. Introduction

Humanitarian assistance and disaster relief (HA/DR) has long been appreciated as one of the most compelling applications of robotics technology, giving responders tools to sense and act in dangerous environments.[1] For example, the use of robots in the aftermath of the Fukushima Daiichi nuclear disaster has been well documented,[2] and analysis of the response suggests that action at one of several "inflection points" of the crisis would have probably averted further catastrophe[3] if those actions had not been deemed too dangerous at the time. Partly inspired by these implications, the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge was conceived to catalyze the focused development of solutions for solving the myriad of challenges related to locomotion, manipulation, perception, and human interface that are needed to build a robot that can act as a stand-in for humans at such "inflection points" in the future. Though this "avatar" concept inspires the imagination, we would argue that robotics has an even more important role to play in the broader HA/DR mission as the backbone for the required information-gathering activities that lie at the heart of any coordinated response. As an illustration, the *Foreign Humanitarian Assistance* manual published by the US Department of Defense[4] identifies that the military will primarily assist in a few ways to a disaster requiring government response: with the first-responder Crisis Action Team tasked as the immediate responder and *assessor* for the regional commander; and with the Humanitarian Assistance Survey Team whose primary responsibility is assessment, such as dislocated populations, degree of property damage, and remaining communications infrastructure. These are all activities that feed into the planning phase that must happen *before* any larger action can be carried out. Though not quite as exciting as a humanoid robot that wades through a flooded disaster site to extinguish a critical fire, we believe a heterogeneous, multi-robot team that can quickly navigate through an environment to quantify an emerging situation is more important to the timeliness and success of the larger response.

Two important focal points of multi-robot systems deployed in a primarily information-gathering sense have been the Robocup Rescue League[5] and the Multi Autonomous Ground-robotic International Challenge (MAGIC) 2010 competition.[6,7] From these activities, we learn that, although physical platform capabilities do play a role, the majority of the system complexity is

derived from the overarching operational problems of team management and communication.

Toward this end, this work focuses on establishing a preliminary formal problem description that places an HA/DR-inspired, information-gathering mission in an operations research context (Section 2) and describing a multi-robot system capable of performing such a mission in a relevant field environment. We present the design of such a system (Section 3), a set of autonomy-enabled behaviors that can be used to address the HA/DR mission (Section 4), and a graphical user interface (GUI) that allows a human operator to task the system (Section 5). Finally, we report extensive experimental results, which address the current capabilities of our system with respect to the implementation of a solution to the HA/DR mission (Section 6).

2. Problem Statement

Within the scope of information-gathering activities required for planning a response to a HA/DR scenario, we focus on simultaneously solving 2 specific problems: evaluating of damage to infrastructure in the environment, e.g., traversability of roads; and localizing particular targets of interest, e.g., a potentially injured "very important person" (VIP) who we discover through sensing a radio signal, such as a cell phone. This problem statement contains both *a priori* goals (key assessment sites established from prior maps) and *dynamic* goals (the existence and possible locations of targets), and a solution must focus on effectively balancing between these 2 types of goals. Moreover, we address the issues of unreliable autonomy and limited communications through incorporation of dynamically uncovered costs, and we cast the entire problem as a dynamic variant of the Capacitated Team Orienteering Problem (CTOP) with details discussed below.

If we considered only the problem to efficiently visit a set of locations derived from prior maps of the environment, a classical formulation would suffice. Initially it could be as a well-studied Vehicle Routing Problem (VRP): with known travel costs between sites, find paths for multiple vehicles to visit all sites that minimize total travel costs. However, since we may assume that the mission is time-critical and some sites are likely to be more interesting than others, we could instead formulate it as a Team Orienteering Problem (TOP):

with known travel costs between sites and known rewards for visitation, find paths that maximize the total gathered reward with a fixed cost bound.[8] The environment limitations suggest 1 final modification.

Because the environment is communications-limited, we conjecture that as we send robots to visit sites and gather information, we need them to eventually return to communications range in order to offload their information before it becomes too outdated. This is most closely modeled as a CTOP: as a TOP but with a constraint on the total reward that any individual vehicle may gather on a single trip.[9]

A key component of the problem is the dynamic goals that arrive because of detecting unknown targets. We model these as dynamically updated rewards available at the visitation sites of the CTOP, and we assign the value of these rewards according to the expected information gain about the target location using the available sensing, similar to information-guided exploration strategies.[10] If we assign a distribution to these rewards initially or as the mission progresses, there is prior work on solving TOPs with stochastic rewards[11] that could apply.

The last challenge is to incorporate the effects of unreliable autonomy; in our experience, this mostly manifests itself when performing autonomous navigation between 2 waypoints and the intervening terrain has features that make perceiving or avoiding obstacles difficult. Two waypoints along a road usually present no problem, even if they are far apart, but navigating between 2 waypoints in a grassy field may cause the navigation system of a ground platform to believe that there are tiny obstacles everywhere as laser scans pick up individual blades of grass and consequently fail to find a path. We model these effects as unknown travel costs between visitation sites: we may have some intuition about how likely it is for a given site-to-site navigation to be successful, but ultimately we build a navigation risk model during operation in the environment. It is important to note that failed navigation is not necessarily fatal because we assume we have backup behaviors to return to a known safe location. If we assign a distribution to these costs, there is prior work on solving TOPs with stochastic costs[12] that could apply.

Our preliminary formal problem formulation is thus as a CTOP with stochastic

(unknown) costs and rewards. We ask, what value is it to have such a formal problem given that we are not developing an online planner to demonstrate through these experiments? The answer is that having the solution for any specific mission instance gives us an upper-bound on how well any autonomy or human could perform at the task and therefore gives us a metric to know when the system is improving. Even for the case of unknown costs and rewards, we can solve the plan as if the costs/rewards were known up front or solve it in a receding-horizon fashion as information is uncovered. Developing these upper-bounds for this experiment remains future work.

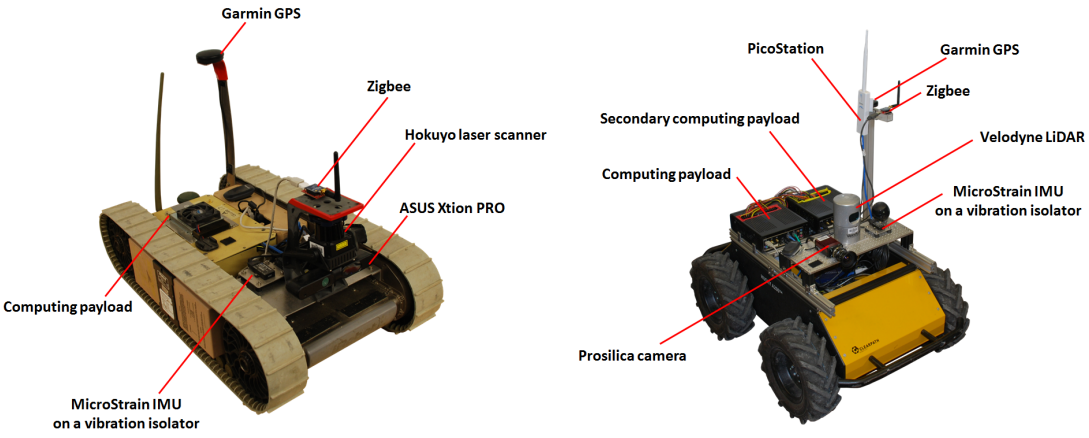
3. System Overview

We present a heterogeneous, multi-robot system capable of field testing basic-research algorithms focused on a wide range of military applications in relevant environments, e.g., military operations in urban terrain (MOUT) training facilities. In particular, our focus is on moving from small-scale systems operating in controlled laboratory environments to the study of interacting systems and the development of algorithms that can operate robustly in real-world scenarios. To that end, our design decisions regarding hardware and software infrastructure are driven by the need for these systems to "survive the field" and allow for reliable evaluation of autonomy-enabling algorithms flexibility and *sufficient* reliability are our objectives.

3.1 Hardware

Two robotic platforms are used in this work: an iRobot PackBot[13] and a Clearpath Robotics Husky.[14] The PackBot, seen in Fig. 1a, is a military-grade, tracked platform capable of speeds up to 2 m/s and traversing both indoor and outdoor terrains. To enable autonomous operation, the PackBot is outfitted with a processing payload containing a Quad-Core Intel i7 ICOM express board and a 256-GB solid-state drive (SSD). The PackBot collects three-dimensional (3D) point cloud data by nodding a Hokuyo UTM-30LX-EW LiDAR[15] with a Dynamixel servo. This Hokuyo LiDAR has a 270° field of view, 30-m range, and 1-mm resolution. Accurate state information is achieved using a MicroStrain 3DM-GX3-25 inertial measurement unit (IMU)[16] mounted on a custom-made vibration isolator. A Garmin 18x PC global positioning system (GPS) sensor[17] is elevated on a mast in an effort to receive better GPS measurements. Finally, an ASUS Xtion Pro Live provided red, green, blue (RGB) data.[18]

The second robot used in this work, the Clearpath Husky seen in Fig. 1b, is a larger, wheeled platform that is limited to a maximum velocity of 1 m/s and is best suited for outdoor operations. Similar to the PackBot, the Husky employs a MicroStrain 3DM-GX3-25 IMU and a Garmin 18x PC GPS. The Husky is equipped with 2 Quad-Core Intel i7 Mini-ITX processing payloads, each with a 256-GB SSD. The Husky has a Velodyne HDL-32E LiDAR,[19] which generates a 360° point cloud of 700,000 points per second at a range of 70 m and an accuracy of up to ±2 cm. Finally, the Husky collects imagery data using a Prosilica GT2750C, 6-megapixel charge-coupled device (CCD) color camera.[20]



(a) (b)

Fig. 1 The hardware configurations of (a) the iRobot PackBot and (b) the Clearpath Husky

Both robots use Ubuntu 14.04 (Trusty) and leverage the open-source *Robotics Operating System* (ROS) Indigo[21] to support higher-level algorithms for mapping, navigation, and autonomous capabilities.

3.2 Mapping and Navigation

While the focus of this work is on the evaluation of high-level site-visitation tasks, the capability for a mobile robot to robustly operate autonomously hinges on its ability to understand its environment, know *where* it is within that environment, and successfully move within that environment, the problem of mapping and navigation.

3.2.1 Mapping

Though the GPS coupled with low-cost IMUs can provide accurate localization capabilities under certain circumstances including unobstructed line of sight to a large number of satellites, a more robust method for managing a world model or map is necessary for 2 reasons. First, ground robots are inherently useful because they can operate in cluttered environments that are specifically not visible from aerial assets, i.e., areas with degraded or denied access to the GPS. Second, the ability of a robot to move autonomously through an environment depends on its ability to understand the structure and traversability of that environment this requires precise, consistent knowledge of the robot's trajectory as it gathers sensor measurements. The simultaneous localization and mapping (SLAM) problem aims to address these capabilities and has been studied for some time in the robotics literature.[22,23] We adopt a modern *graph-based* solution to the SLAM problem based on the square-root smoothing and mapping ($\sqrt{\text{SAM}}$) technique[24] and the *GTSAM* software library developed at Georgia Tech.[25] Our technique leverages the Generalized Iterative Closet Point (ICP) algorithm[26] for dense inter-frame matching of point cloud data and loop closure constraints. GPS measurements, when available, are robustly incorporated into our solution based on the techniques described in our previous work.[27]

We refer to our SLAM system as *OmniMapper* due to its ability to integrate sensor data from a variety of sensor sources including Velodyne 3D laser scanners, Microsoft Kinect 3D cameras, Hokuyo two-dimensional (2D) laser scanners, which are mounted on a tilting platform, as well as planar, unactuated 2D laser scanners. We divide the components of this system into a backend, the *OmniGraph*, which is responsible for solving the factor graph representation of the SLAM problem, and a frontend, the *OmniCache*, which is responsible for managing sensor data and performing computations that yield the probabilistic factors connecting nodes in the factor graph. An example factor graph like one used by the *OmniGraph* can be seen in Fig. 2. Our SLAM back-end called *OmniGraph* solves for the robot's optimal trajectory using the *GTSAM* library; the front-end tasks of data association and generating relevant measurements is handled by the *OmniCache*.

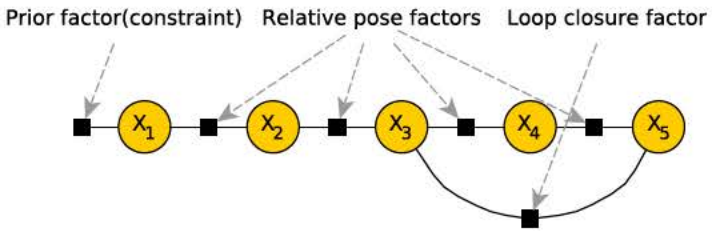


Fig. 2 An example factor graph like one used by GTSAM. Pose variables X_i are connected through measurements called *factors*.

The point-cloud *OmniCache* used in this work receives local point-cloud data aggregated over small time windows based on the odometry of the robot and serves 2 primary purposes. First, it can respond to queries about the relative pose of 2 local point-clouds via ICP algorithms in order to generate measurement factors. Second, it acts as a pipeline for generating a series of data products based on the underlying local point-cloud data. This includes a set of intrinsic products, i.e., ones that are invariant to the global pose of a local point-cloud, such as per-cloud terrain classification, occupancy grid rendering, and terrain height estimation. Other products are extrinsic, i.e., ones that must be recomputed after optimization of the factor graph yields a new optimal trajectory for the robot, including an aggregated point cloud and composite occupancy grid map. A block diagram of the relevant components of the *OmniMapper* can be seen in Fig. 3. Once an optimized trajectory is computed, each robot broadcasts its current location in a GPS-based reference frame to all clients. This broadcast is at a low enough rate so that it does not significantly impact the bandwidth available to other services on the network. The position data of other agents are inserted as obstacles into the robot's costmap, which is later used for planning and trajectory generation.

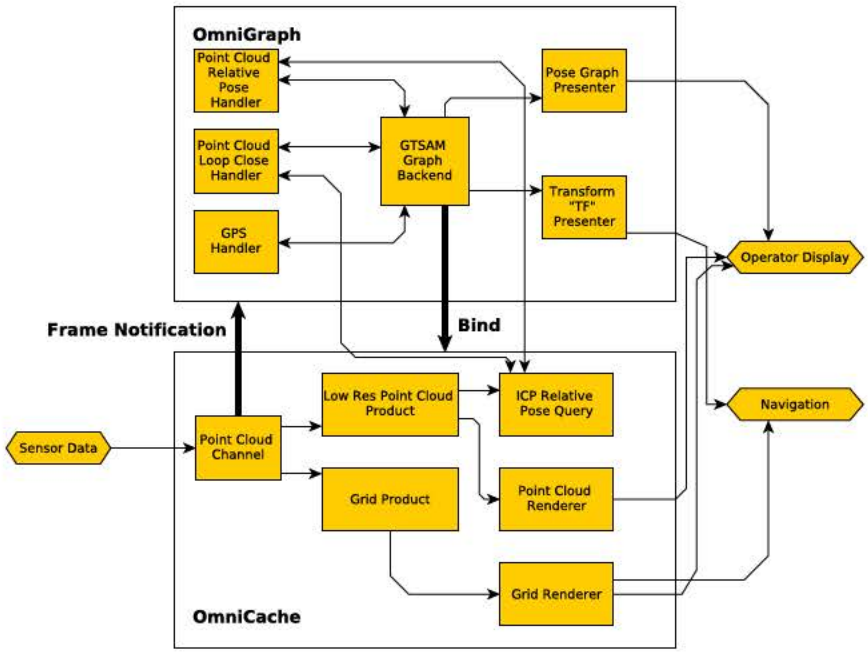


Fig. 3 The OmniMapper configuration used in this work

3.2.2 Navigation

Given an occupancy-grid map of free space and obstacles in the environment, the navigation problem consists of 2 main parts finding kinematically feasible paths and controlling to follow the chosen path. Finding kinematically feasible paths is the motion planning problem traditionally considered in the robotics literature[28] though finding detailed plans through a large environment can be computationally expensive. Finding control inputs to follow that path is, generally speaking, a nonlinear control problem. We address the coupled problem of general navigation with a 3-stage architecture including a global motion planner, a local planner, and a local controller as depicted in Fig. 4.

We use the idea of a 3-stage architecture to drive our software design within the ROS framework. That is, each stage of the navigation system is implemented as a node, or independent software process, which provides an *ActionServer* interface. *ActionServer* interfaces are a ROS construct used to deal with long-running tasks and include an internal state machine to manage the setting of goals, task feedback, and eventual completion state, i.e., success or failure. Each stage of our navigation architecture provides an *ActionServer* that re-

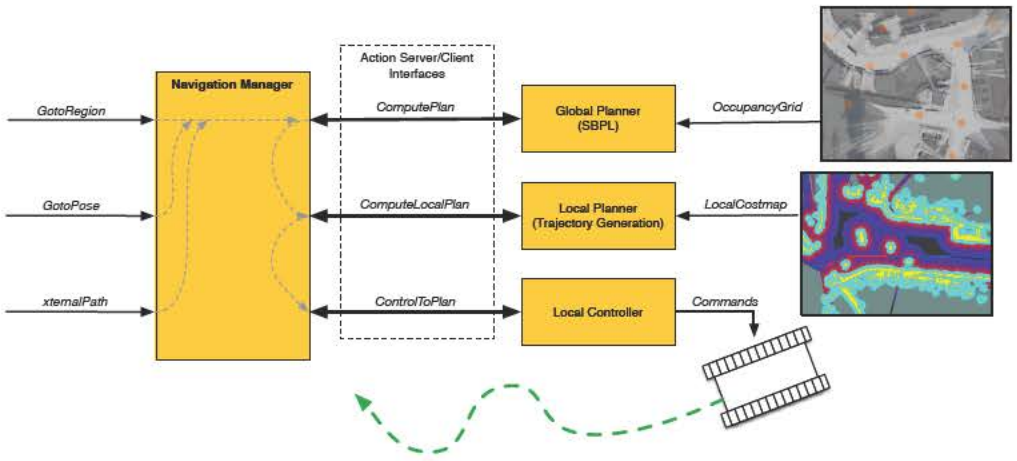


Fig. 4 Architecture for autonomous navigation tasks

sponds to an abstraction of the navigation problem. For instance, the global planner provides a *ComputePlan* action, which takes as input a starting and goal pose given the current map, it returns an optimal kinematically feasible path. The local planner provides a *ComputeLocalPlan* action, which takes a global plan as input and uses the robot's current pose and a local map of dynamic obstacles to find a short-term high-resolution path that follows the global plan. Note that the decoupling of the planning problem into a global and local component is a standard approach used in the robotics literature the local planner is capable of generating high-resolution plans over a short time-horizon while the global planner helps prevent the system from being trapped in local minima caused by non-convex environments. Finally, the local controller provides a *ControlToPlan* action, which takes the current local plan and the current state of the robot to compute control inputs, which can be sent to the underlying platform.

Sequencing of the actions is performed by a *NavigationManager* process, which presents an external interface to the user or application, e.g., the *MoveBase* action, which mirrors the traditional ROS interface to navigation to provide the capability of driving to a desired pose. The software architecture presented above is designed to maximize flexibility in implementing different solutions to not only each component of the navigation system, but also provide flexibility in how the external interface to navigation is presented the *NavigationManager* is a fairly simple state machine and allows for easy adaptations of the

traditional pose-based navigation interface. For instance, in this experiment, we relied exclusively on a variant of the *MoveBase* action and instead use a *GotoRegion* action, which drives the system to any pose within a defined region.

For this experiment, we rely on the Search-Based Planning Library (SBPL)[29] to perform global planning actions. We generate a custom set of motion primitives based on our platform's kinematics and use of 0.2 and 0.3 m occupancy-grids for the PackBot and Husky, respectively. We use the ARA* planner algorithm and compute reverse plans so that computations can be reused as the robot drives for fast re-planning actions. Re-planning allows the system to quickly correct in the event of errors in platform control or updates of the occupancy-grid map. Feasible solutions to most initial planning queries are found in less than a second with optimal solutions being found in a few seconds for most scenarios.

Local planning and control actions are currently provided by a single process, which performs optimal trajectory generation over the space of time-varying control inputs. As stated above, local control is essentially a nonlinear control problem. Based on prior work in trajectory generation,[30] we formulate a parameterization of the control input for a differential-drive platform such that a relatively small number of variables, 4 in our current instantiation, provide an expressive description of the possible trajectories available to the robot over a short time horizon of $T = 3$ s. An objective function is devised that performs a weighted minimization of the error between the robot's path and the desired global path coupled with some curvature minimization terms to prevent overly aggressive trajectories. The final optimization problem, including bounds on the parameterization of the control input, can be solved with a variety of algorithms implemented in the NLOPT library.[31] We are typically able to solve the trajectory generation optimization for a time horizon of $T = 3$ s in $5 - 10$ ms, allowing for a control frequency of 10 Hz. We are able to directly execute the optimized time-varying control inputs, thus simultaneously addressing the local planning and control problems.

3.3 Signal-Strength Modeling

The search for an injured VIP can be represented by localizing a radio frequency beacon, e.g., a cell phone. In fact, a variety of spatial information-gathering tasks, including chemical and radiation analysis, can be emulated with radio signal propagation from 1 or more beacons.

We use a low-power IEEE 802.15.4 XBee radio, shown in Fig. 5, to broadcast a beacon once per second at 2.4 GHz. Each robot also carries a XBee radio, as shown in Fig. 1, and records radio signal-strength indication (RSSI) when it successfully receives packets from the beacon while traversing the environment in pursuit of the other data-collection tasks. The locations that these RSSI measurements have received are also recorded.

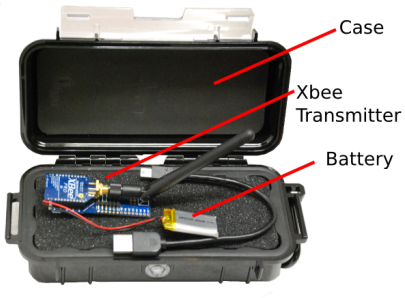


Fig. 5 XBee "beacon signal" transmitter with protective case

By aggregating the signal-strength measurements from multiple robots in many locations across the environment, the operator can infer an estimate of the beacon location from the maximum of the signal-strength field. This task is complicated by the fact that radio-signal propagation is notoriously challenging to model in complex urban environments due to the phenomena of shadowing and multi-path. Furthermore, a high frequency beacon transmission may make complete reconstruction of the signal-strength measurements at the operating station impractical. We employ a segmentation-based approach for modeling that allows each robot to maintain efficient models of the received signal strength.[32] This method relies on analyzing the obstacle costmap and iteratively creating regions that comprise a portion of the traversable area of the obstacle costmap. Regions are characterized by a single position from which all other points must be below a maximum distance and within line of sight;

each region must also be continuous. As the robot uncovers traversable areas through exploration, regions may expand into these new areas. Otherwise, size and line-of-sight constraints necessitate the creation of new regions.

It has been shown that these regions are a useful tool for grouping RSSI measurements. RSSI measurements from the same region are likely to be better correlated than measurements from other regions.[32] We use this hypothesis to justify averaging the RSSI measurements collected in a region together. The resulting list of averaged RSSI with a corresponding region center is much lower bandwidth, less noisy, and more succinct. These compressed models can be transmitted to the operator and visualized to allow adaptive exploration of the environment with the goal of accurately localizing the VIP beacon.

3.4 Wireless Communication

Robust wireless communication is an important capability for the multi-robot experiments described in this report. While the primary role of wireless communication is to support the transfer of information and commands to and from the human operators during experiments, it also plays an important role during development, enabling software updates and easy tuning of algorithmic parameters. These 2 objectives lead to a desired system design that resembles a realistic deployed network with sometimes severe limitations on the range of communication but includes the capability to quickly deploy additional radios - creating a rich infrastructure for the development paradigm.

We use embedded hardware equipped with off-the-shelf *IEEE 802.11.g* radios operating in the 2.4-GHz frequency band to provide the necessary wireless connectivity. Specifically, we use the Ubiquiti hardware listed in Table 1.

An advantage of the Ubiquiti embedded systems is that we can easily use OpenWRT,[37] an open-source Linux distribution focused on the network router application we currently use version 14.07 (Barrier Breaker).

Each wireless radio is operated in *AdHoc* mode, i.e., no central access point is defined or required for inter-node communication. End-to-end connectivity is supported by the *B.A.T.M.A.N.* mesh routing protocol version 2014.2.0.[38] *B.A.T.M.A.N.* is a Layer-2 routing protocol, meaning that it operates entirely over raw ethernet frames and all nodes on the network appear to be linked

Table 1 Embedded hardware for wireless communication

Ubiquiti RouterStation Pro[33]	Atheros MIPS CPU at 680 MHz, 4 Ethernet ports (10/100/1000), 128 MB RAM, XR2 2.4 GHz 802.11b/g radio module capable of 28 dBm transmit power[34]
Ubiquiti RouterStation[35]	Atheros MIPS CPU at 680 MHz, 4 Ethernet ports (10/100/1000), 128 MB RAM, XR2 2.4 GHz 802.11b/g radio module capable of 28 dBm transmit power[34]
Ubiquiti PicoStation2HP[36]	Atheros MIPS CPU at 180 MHz, 1 Ethernet Port (10/100), 32 MB RAM, Embedded 2.4 GHz 802.11b/g radio capable of 28 dBm transmit power

local. Layer-2 routing simplifies the network configuration of the clients using the mesh and allows us to easily employ a high-reliability wired "backbone", which drastically improves the robustness of our wireless communication system when deploying a development infrastructure.

Since the focus of these experiments was not on teaming or inter-robot communication, we allocated each robotic platform with a unique frequency for communication. For the experiments, a "base station" was located in an advantaged location, i.e., a tower approximately 20 m above the ground, and equipped with 2 radios to support communication with each of the robots being deployed. The placement of the "base station," environment complexity, and the fact that each robot's radio was placed very close to the ground, induced a communication environment within our test facility, which clearly exhibited regions of high-bandwidth reliable communication, intermittent unreliable communication, and no communication at all. While the *B.A.T.M.A.N.* routing protocol supports multi-hop communication, we restricted all communication in this experiment to be over a single wireless link in order to simplify the modeling of communication capabilities.

3.5 Multi-Robot Capabilities

The focus of this work is not primarily multi-robot coordination insofar as each robot independently pursues a series of data-collection tasks after being tasking by a centralized planner or human operator. However, the robots do operate in the same physical environment and though the centralized planning will generally yield motions that keep each robot in disjoint regions of the environment, some interagent knowledge and coordination may be necessary for efficient navigation. This stems primarily from the fact that platforms such as the PackBot are very low-profile and thus have small sensor signatures in a 3D point cloud — the primary source of traversability and obstacle detection for navigation.

We address this problem by having each robot broadcast its current location in a GPS-based reference frame to all clients in the network. This broadcast is made up of a small UDP datagram and is sent at a low enough rate, e.g. 2 Hz, that it does not significantly impact the bandwidth available to other services on the network. It should be noted that routing of UDP broadcast packets is equivalent to a flooding algorithm when using the *B.A.T.M.A.N.* mesh protocol. This makes for a very reliable form of communication — the recent positions of the robot were generally available to the mission operator even near the extent of the feasible communication range. Each robot's algorithms for autonomous navigation incorporate other agent's positions by inserting obstacles into the maps used for planning and trajectory generation based on the broadcast data.

4. Behaviors Supporting Autonomy

In the previous section, we described the basic capabilities of our multi-robot system. In this section, we describe how we build automata to sequence these capabilities in order to provide higher-level autonomous actions and begin to address the data-collection mission described in Section 2. While the behaviors described here are fairly simplistic, the underlying architecture allows for complex collections of actions.

For the purposes of this work, all of our navigation behaviors build on the canonical *GotoRegion* action in which the robot plans and drives to an arbitrary pose within a defined region of the environment. Our current implemen-

tation requires each region to be defined as a disk with center and radius, but can easily be extended to regions of arbitrary geometry. The design decision to rely on region-based navigation is based on the observation that navigation to a precise pose in the environment leads to brittle solutions and that many data-collection problems can in fact be satisfied with large degrees of flexibility. Take for example, the image collection problem there are many viewpoints from which to obtain a suitable image of a target in $\mathbb{R}^3$. While the complexity of solving this viewpoint problem is beyond the scope of this work, we believe many future data-collection problems can be generalized to a desired region in the environment.

At their core, the behaviors generated by sequencing basic capabilities are meant to aid the operator in tasking the robot when it must go outside the area of reliable communication. Thus, we begin by defining the *GuardedNavigation* behavior to be one where a *goal* region and *safe* region are defined. If execution of navigation to the *goal* fails, the robot navigates back to the *safe* region where communication is known to be reliable and the operator can continue to task the robot. Clearly, the *GuardedNavigation* behavior can be extended to support sequences of *goal* regions such that a failure at any point in the sequence results in pursuit of the *safe* region.

With the addition of a simple *Collect* action that causes the robot to record and store an image, the operator can immediately begin to address the data-collection mission from Section 2. By specifying a sequence of *goal* regions with accompanying *Collect* actions, i.e., a *goal* region and a *safe* region, the operator instructs the robot to visit a number of sites at which it will record high-resolution images. When it completes visiting the sequence of *goal* regions or deems a leg of the task to be infeasible, the robot returns to the *safe* region with its known reliable communication and transmits the images to the operator. For now, the operator selects *safe* regions based on previous locations from which the robot has successfully transmitted data.

5. Operator Interface

We rely on a simple GUI that enables a human operator to task 1 or more robots. Our GUI is based on the RViz application that is included in ROS for 3D rendering of sensor-data visualizations, tools for on-screen interactions, and an extensible plugin architecture. In addition to software components that allow for visualization of experiment-specific data, we developed tools for creating and interacting with generic graph-embeddings on $\mathbb{R}^2$, which are used to specify autonomous behaviors. It should be noted that our design and implementation of an operator interface is driven by necessity in order to evaluate our system in appropriately relevant scenarios rather than as an example of best practices in terms of human-robot interaction. Indeed, the design of an efficient interface that allows a single human operator to task many autonomous systems is a research topic in its own right and beyond the scope of this work.

5.1 Visualization of the System State

For this work, we used RViz to display a top-down orthographic view of satellite imagery of our experimental facility, as seen in Fig. 6. Along the top bar of the application, there are a collection of tools for interacting with the objects in the main view area. Beside the main view are customized panels that allow the human operators to task each robot.

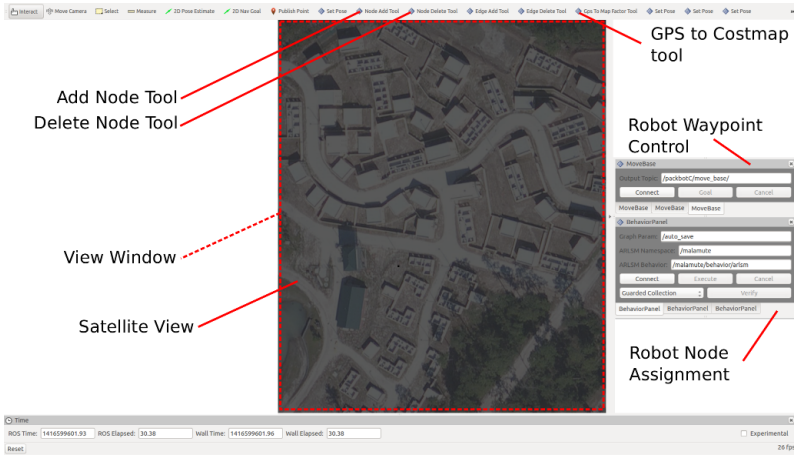


Fig. 6 The RViz panel, which contains several different windows and buttons for mission specification and verification

While our factor-graph-based algorithm for mapping does incorporate GPS measurements that fix the robot's reference frame relative to the GPS reference frame, systematic bias in these measurements can lead to errors in the alignment. Since the robots perform autonomous navigation in a local map-based reference frame, this alignment is important in order for the robot to reliably execute missions specified by a human operator based on a priori knowledge of the environment, e.g., satellite images. We have implemented a "GPS To Map Factor" tool, which allows the operator to manually align the map generated by the robot with the satellite imagery and insert alignment as a precise measurement to the underlying mapping system. In this way, a GPS to map reference frame alignment, which may require many measurements to accurately converge, can be bootstrapped in the beginning of an experiment. Fig. 7 shows a view where the satellite imagery is overlaid with an aligned occupancy grid produced by the 3D mapping techniques described in Section 3, along with the current positions of all robots in the system.

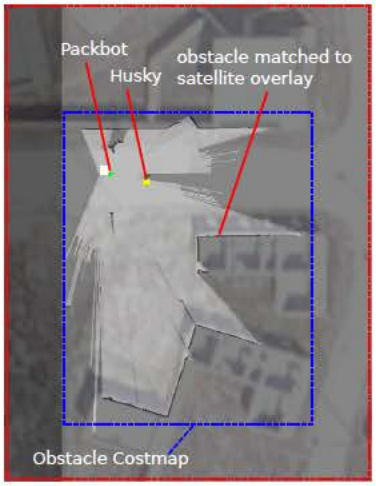


Fig. 7 An example view of the satellite overlaid with an occupancy grid

5.2 Tasking the System

We designed a generic set of tools for creating and editing a graph object, i.e., a collection of nodes and edges, that is embedded in the metric space within which the robots operate, as seen in the top bar of Fig. 6. We rely on a generic graph structure because it presents an intuitive representation for a variety of tasks including patrol, exploration, and data-collection. For the purposes of this work, we focus on the data-collection task and implicitly add edges

to create linear topologies along a sequence of nodes, which are defined by a disk with a center position and radius. After the operator has annotated each node as *safe* or *goal*, we can easily map a graph onto the behaviors described in Section 4. Figure 8 depicts an example autonomous sensing task given to a robot.

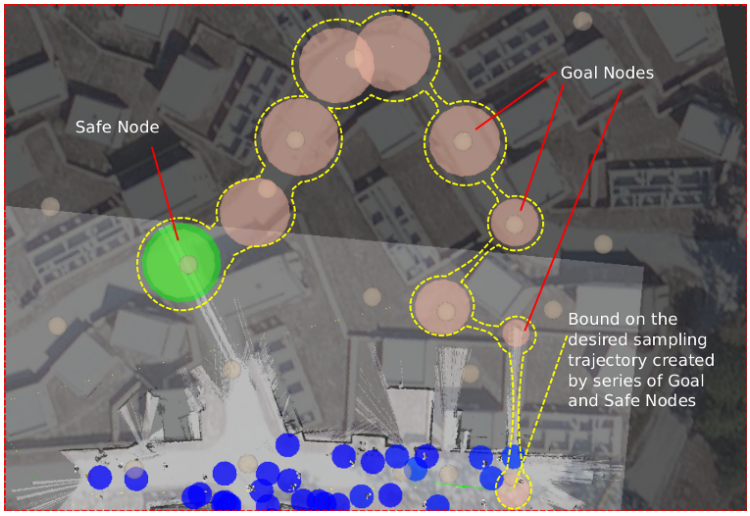
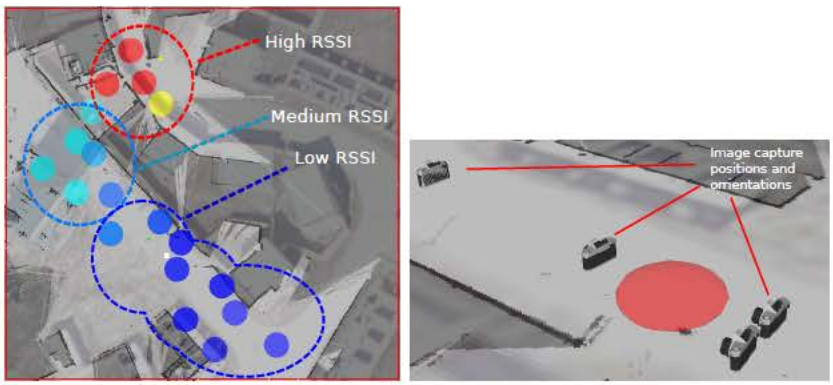


Fig. 8 An example of the user interface for a single data-collection task in a trial. The map is overlaid on top of a satellite image with small pink disks representing the predefined GPS mission nodes. The blue disks indicate that the robot has measured poor received signal strength data thus far. The large orange and green disks are the goal and safe nodes, respectively, as set by the operator. Note, the red lines, white text, and yellow dotted lines have been manually added for clarity.

After defining a graph in RViz, the system runs a verification to ensure that there are 1 or more *goal* regions and only 1 *safe* region for each task. The mission definition is then communicated to each robot where the resulting state machine is executed. Since the execution takes place onboard, the robot continues to run even if communication with the operator is lost or unreliable. At completion of the task, the color of the panel indicates success or failure, i.e., if the robot was forced to return to the *safe* region before visiting all of the *goal* regions.

5.3 Visualization of the Exploited Data

Another aspect of the GUI was visualizing the information that the robots collect during execution of their data-collection behaviors. The RSSI and images comprised the data collected, as depicted in Fig. 9.



(a) (b)

Fig. 9 Examples of (a) the average RSSI sampled in different areas traversed by the robot and (b) the images taken at the goal nodes

The color of the RSSI circles show the average RSSI in an region generally within 2 m of the marker.[32] When the experiment starts it is possible that the robots will be far enough away from the signal source that it is not possible to detect the RSSI. As the robots traverse the environment, they attempt to sample RSSI and assign samples to spatial regions. As a result, RSSI markers for areas will not be created in the GUI until several valid RSSI measurements are made for a given region in order to properly smooth the highly variable signal strength measurements. As RSSI is sampled closer to the signal source, the color of the marker for that region changes from blue to cyan to yellow to red.

When the robot enters a *goal* region, it captures a high-resolution image and stores it in a local cache. When the robot reaches the *safe* region, it transmits all of these images back to the operator where they can be visualized. Camera icons are displayed in the GUI to represent the pose from which each image was captured. The operator can then select each image in order to display the view from that pose.

6. Experimental Results

We conducted a series of experimental trials using the 175 x 175 m environment pictured in Fig. 10a to evaluate the capability of our system to address missions defined according to the problem statement in Section 2.

(a) (b)

Fig. 10 A satellite overview of the experimental facility overlaid with (a) experiment annotations (green: operating center, purple: elevated base station antenna, orange: mission-specified sites, red: VIP location for each trial, black arrow: starting location for all trials) and (b) the aggregated paths driven by robots over all trials

Each experiment consisted of 1 or 2 robotic platforms and mission operators tasked with the mission of capturing an image at as many of the defined collection sites as possible within the time limit of 20 min. Experiments were designed such that the visitation of some collection sites require traversal over a variety of terrain complexities and that robots must travel outside of communication to motivate the use of autonomy. While collecting images, each robot monitors the received signal strength from a radio beacon carried by a mock VIP that is hidden in a static location for the duration of an experimental trial. Localization of the VIP through received signal strength at the end of each 20-min experiment is an auxiliary intelligence-gathering task that further guides the exploration strategies employed by the mission operator.

While we envision a multi-robot system capable of autonomous traversal of the complete mission with high degrees of reliability, i.e., suitable for tasking by an autonomous agent that dynamically optimizes vehicle routes, this is beyond the scope of state-of-the art algorithms when implemented in a realistic field environment. The use of a safety operator not constrained by unreliable communication, i.e., following the robot through the environment who is able to

intermittently intervene and control the robot's actions, drastically improves our ability to collect information on the system performance across an entire mission execution. As such, evaluation of the frequency and duration of these interventions serves as a primary benchmark in terms of rating current autonomous capability. Additionally, each mission operator was given a joystick and allowed to intervene remotely from the base location; however, it should be noted that the absence reliable communications often made this error-prone.

We report on the results of 9 experimental trials with respect to the number of sites visited, the number of interventions, the distance traveled autonomously, and the mock VIP localization accuracy in Tables 2 through 5. The trajectories traversed by both robots across all experiments are overlaid in Fig. 10b to depict the breadth of experiments conducted. In most experimental trials, the robots drove more than 90% of their total distance while autonomously executing *GuardedNavigation*-based sub-missions designed by the human operators to gather high-resolution images and VIP signal strength data.

Table 2 The number of site visitations for each experimental trial

Trial	*PackBot*	*Husky*	*Total*
3	2	2	4
4	9	4	13
5	1	8	9
7	4	3	7
8	7	8	15
9	7	10	17
10	7	8	15
11	5	7	12
12	11	6	17

Table 3 Data comparing when an operator intervened and when the PackBot operated autonomously

Trial	*Interventions (Mission/Safety)*	*Intervention Distance (m)*	*Autonomous Distance (m)*	*Total Distance (m)*	*Percent of Mission Autonomous*
3	0/17	14.5	101.2	115.7	87.5%
4	7/13	51.6	386.4	438.0	84.1%
5	0/22	34.2	175.5	209.7	83.7%
7	4/1	162	9.9	171.9	5.8%
8	5/5	25.6	334.3	359.9	92.9%
9	1/7	26.1	403.1	429.2	93.9%
10	1/12	61.5	454.4	515.9	88.1%
11	4/19	48.1	446.3	494.4	90.3%
12	0/13	107.0	605.9	712.9	85.0%

Table 4 Data comparing when an operator intervened and when the Husky operated autonomously

Trial	*Interventions (Mission/Safety)*	*Intervention Distance (m)*	*Autonomous Distance (m)*	*Total Distance (m)*	*Percent of Mission Autonomous*
3	0/4	2.6	167.1	169.7	98.5%
4	1/6	21.9	336.4	358.3	93.9%
5	0/9	77.9	494.4	572.3	86.4%
7	0/1	0.5	169.9	170.4	99.7%
8	0/6	1.7	378.4	380.1	99.6%
9	0/16	15.2	371.4	386.6	96.1%
10	0/5	0.1	426.2	426.3	99.9%
11	0/0	0.0	342.7	342.7	100.0%
12	0/11	125.7	326.0	451.7	72.2%

Table 5 Approximate error between the estimation made by the operators and the actual location of the mock VIP beacon. Note, 0 m indicates that the operators had an image with the VIP in the field of view and thus knew the location exactly

Trial	*Localization Estimation Error (m)*
3	60
4	15
5	3
7	0
8	2
9	45
10	53
11	0
12	8

Appendix A contains a collection of figures that depict all of the experimental results. For each trial, we show 1) the maps and RSSI visualization generated by the PackBot and Husky, respectively, as seen by the mission operators at the end of a single trial; 2) the trajectories of each robot with an overlay of interpolated RSSI provided for data analysis; and 3) plots of the communications reliability and rate of collected images. The mission operators were instructed to use the map and RSSI visualization during the experiment to guide their execution of the site-visitation problem and the interpolated RSSI plots demonstrate how these data can be leveraged to provide localization information for the VIP's radio beacon. In practice for this experiment, the mission operators communicated with each other in order to coordinate their search efforts across the facility.

One noteworthy test is trial 11, seen in Fig. A-8. In this trial, both robots begin by heading east until they reach an intersection - at which point the mission planners coordinate the PackBot to travel south and the Husky eastward. Then, after traveling several meters down the road without receiving strong RSSI, both robots return to the common intersection. While traveling to the intersection the mission operators share their maps with one another and determine the only revealing RSSI measurement thus far is located near the starting location. As a result, the PackBot is tasked to return to the starting location while the Husky is sent north to map an uncharted area in the environment. During this process, both robots receive stronger RSSI measurements in the northwest region of the environment. The PackBot mission operator

tasks the robot to navigate through a building complex to reach the road that the Husky is also investigating. Together, the mission operators converge on an area that has very high RSSI and eventually locate the mock VIP's radio beacon exactly by collecting images with the mock VIP within the robot's field of view. This trial is particularly interesting because it requires the operators to use the guarded navigation capabilities to autonomously collect data in regions of dubious communications, as seen in Figs. A-8c and A-8d. Additionally, this trial demonstrates the ability of the mission operators to combine their current knowledge about the environment and cooperatively locate the mock VIP within a predefined time limit.

A similar and equally interesting trial is number 12, shown in Fig. A-9. Again, both robots begin by heading to the central intersection. The PackBot is tasked to explore south and returns to the intersection when no informative RSSI is obtained. Meanwhile, the Husky is sent north and receives the team's first revealing RSSI measurement. Using this piece of information, the PackBot mission operator decides to task the robot to navigate to a road north of its current location by travelling through a narrow alley in the complex. At this point, both robots are operating autonomously as they are outside of communications range. While the Husky explores inside the complex, the PackBot navigates down the north-most road and the 2 robots converge on an area with strong RSSI readings. Eventually all communications are lost between the base station and the Husky so no additional data are received from this robot. Finally, the 20-min time limit expires and the mission operators are required to mutually agree upon their estimation for the mock VIP beacon location. Although incorrect, the operators' assessment for the mock VIP beacon location was within 10 m of the actual location. This trial, like the previous example, illustrates the usefulness of the guarded navigation capability to allow for data collection in areas that would otherwise be unnavigable due to the absence of communications. Likewise, this trial emphasizes how beneficial it is to use more than 1 robot for solving the site-visitation and VIP-localization problems, even when 1 of the robots is lost during the mission.

7. Conclusions and Future Work

We have presented a series of field experiments that explore the capability of a heterogeneous multi-robot system when applied to intelligence-gathering tasks in a post-disaster scenario. Our results demonstrate autonomy-enabled operation when communication reliability is not sufficient for teleoperation. Furthermore, by allowing the operators to on-the-fly compose behaviors and define sub-missions that respond to new conditions such as navigation failure, we enable safe operation completely outside the range of reliable communication.

From Table 5 we conclude our system enabled mission operators to localize a mock VIP beacon within 3 m of a static location in 4 of the 9 reported trials. Furthermore, the mission operators localized the beacon exactly in 2 of these trials by acquiring images with the VIP in view. It should be noted that the mission operators were given 20 min for each trial and could presumably decrease localization estimation error in cases of poor performance if given additional time.

By comparing the communications reliability plots in Appendix A, we note that, in general, the Husky experienced more reliable communications to the base station than the PackBot. This improved communication is expected due to the increased height of the Husky and, in turn, its antenna for wireless communication, relative to the radio installed on the PackBot. Indeed, it is well known that the so-called "ground effect" can drastically affect radio signal performance when transmitters are placed to close to the ground. After several trials, the mission operators noticed this difference in communication reliability and adjusted their task allocation accordingly.

Similarly, from the image collection plots in Appendix A, we see that 2 distinct methods for allocating goal waypoints arise. The mission operator commanding the PackBot chose to assign several goal waypoints so that the robot would transmit a larger number of images at each safe region. The mission operator tasking the Husky, on the other hand, only assigned 1 or 2 goal waypoints for each safe region, resulting in a more linear transmission of imagery data. The latter technique can be considered a safer approach as fewer images will be lost in the event the robot loses communications with the base station, but

also requires more effort and oversight by the operator to task short missions more frequently.

The results from the reported trials shed light on a number of improvements that can be made to the existing system as well as avenues for future work. In terms of the data collection process, images are taken at the boundary of the waypoints assigned by the mission operator and are often not in the desired direction for maximal information gain. This drawback is most apparent in trials such as 4, seen in Fig. A-2b, where the Husky drives within 2 m of the beacon location, but fails to capture imagery data containing the mock VIP. As a direct result, the mission operators were unable to successfully localize the beacon. We note that the system would benefit dramatically from additional task-level information, specifically for the VIP localization mission - that is, the waypoints assigned by the operator should incorporate an orientation in addition to location information. This way the operator can investigate specific areas of the environment by receiving very focused images. Along the same lines, the robot could present the mission operator with even more information if an omnidirectional camera were installed. An alternative to this solution would be a specific procedure that the robot executes at each waypoint where it rotates in place and takes a picture approximately every 45°. An additional drawback to the current data collection process is the downtime the operators have while waiting for the robots to finish executing navigation tasks. To alleviate this issue, we suggest that the robots attempt to transmit each image as soon as it is taken rather than waiting until the safe regions are achieved. In the event the robot is within communications range, the mission operator will receive the image sooner and can be visually searching for the VIP or planning the next iteration of exploration.

Data visualization was not a primary objective for this work and we hypothesize that improved tools in this area would have a positive impact on the mission operator's decision-making process. First, we suggest providing the mission operators with the ability to select each RSSI region and receive more data. For example, each RSSI region could provide the maximum, minimum, and average RSSI measurements for that region. These data would only be displayed when the operator clicked on the RSSI region so as to not clutter the visualization window. Additionally, the system should build some visualization involving the interpolation of RSSI data, as seen in Figs. A-1c - A-9c,

on-the-fly so the mission operator can more efficiently localize the VIP beacon. A similar communications visualization could also be developed to show the quality of communications using the measured latency when transmitting data between the robot and base station. Throughout the navigation process, it was often unclear what the order of waypoints the robot would attempt to achieve and, in the current configuration, the mission operator was required to remember the order of placement if they wanted to reuse these nodes for the next task. By displaying some visualization such as numbers or arrows connecting the nodes, the mission operator would have a much more clear understanding of what the robot was attempting to accomplish. In some circumstances the robot's map would become corrupted due to error in a sensor measurement. Some visualization tool that allowed the mission operator the ability to correct a map by removing or adjusting sensor measurements could be the difference between a successful and unsuccessful mission. Finally, a tool we believe could help improve communications between the robot and base station is a diagnostic visualization that shows statistics on each piece of data that is transmitted. With this tool the mission operator would know exactly how much bandwidth each piece of information requires in order to be received and then make more informed decisions when tasking the robot with collecting data. For example, if the operator recognizes that images require a large percentage of the available bandwidth and the robot is in a communications-limited area, the mission operator may task the robot with fewer waypoints so fewer images are transmitted. Similarly, the mission operator may decide the robot should refrain from transmitting some specific data to free up bandwidth for transferring some other, more important, piece of information.

It should be noted that there is a subtle increase in the reliability of our system afforded by the operator's ability to incorporate a priori knowledge, e.g., the road network, and intuitive uncertainty management to specify region-based navigation as seen in Fig. 8. Encoding the intelligence that goes into incorporating this a priori knowledge will be key to the application of future autonomous planners that schedule the collection mission specifications for multiple robots operating in challenging environments.

In the end, the series of experiments presented here offer a wealth of insight into how a multi-robot system operating with state-of-the art algorithms for autonomous perception, navigation, and intelligence, but subject to environ-

mental uncertainty and communication constraints, can be used to address data gathering missions in the wake of a natural disaster. We see this work as a starting point for 2 distinct lines of inquiry. First, the performance of system acts as a benchmark for the capability of our multi-robot system in conjunction with human operators - we expect that as the reliability and performance of our underlying component systems increases, so will our performance on the overall site-visitation and beacon-localization tasks. Second, the experimental results captured in this work provide data that can be used to model the reliability, performance, and efficiency of our current multi-robot autonomous behaviors. These models will enable future intelligent task allocation algorithms to find robust solutions to a wide range of site-visitation problems that take into account realistic field-environment conditions. Ultimately, the experiments presented here lay the ground work for future systems that allow a minimal set of human operators to intelligently task large numbers of robotic platforms for intelligence-gathering tasks in disaster-relief scenarios.

8. References

1. Murphy RR. Disaster robotics. MIT Press; 2014.

2. Guizzo E. Fukushima robot operator writes tell-all blog. In IEEE Spectrum; Source: `http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries`.

3. Strickland E. 24 hours at fukushima. IEEE Spectrum. 2011;48(11):35 42.

4. US Department of Defense. Foreign Humanitarian Assistance. Joint Publication 3-29; 2014.

5. Balakirsky S, Carpin S, Kleiner A, Lewis M, Visser A, Wang J, Ziparo VA. Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. Journal of Field Robotics. 2007;24(11-12):943 967.

6. Olson E, Strom J, Morton R, Richardson A, Ranganathan P, Goeddel R, Bulic M, Crossman J, Marinier B. Progress toward multi-robot reconnaissance and the magic 2010 competition. Journal of Field Robotics. 2012;29(5):762 792.

7. Butzke J, Daniilidis K, Kushleyev A, Lee DD, Likhachev M, Phillips C, Phillips M. The university of pennsylvania magic 2010 multi-robot unmanned vehicle system. Journal of Field Robotics. 2012;29(5):745 761.

8. Vansteenwegen P, Souffriau W, Oudheusden DV. The orienteering problem: A survey. European Journal of Operational Research. 2011;209(1):1 10.

9. Archetti C, Feillet D, Hertz a, Speranza MG. The capacitated team orienteering and profitable tour problems. Journal of the Operational Research Society. 2008;60(6):831 842.

10. Stachniss C, Burgard W. Exploring unknown environments with mobile robots using coverage maps. In: IJCAI; p. 1127 1134.

11. Ärlhan T, Iravani SMR, Daskin MS. The orienteering problem with stochastic profits. IIE Transactions. 2008;40(4):406 421.

12. Campbell AM, Gendreau M, Thomas BW. The orienteering problem with stochastic travel and service times. Annals of Operations Research. 2011;186(1):61 81.

13. PackBot. http://www.irobot.com/For-Defense-and-Security/Robots/510-PackBot.aspx;.

14. Clearpath Robotics Husky. http://www.clearpathrobotics.com/husky/;.

15. Hokuyo LiDAR. http://www.hokuyo-aut.jp/02sensor/07scanner/download/products/utm-30lx-ew/;.

16. MicroSrain IMU. http://www.microstrain.com/inertial/3DM-GX3-25;.

17. Garmin GPS. https://buy.garmin.com/en-US/US/oem/sensors-and-boards/gps-18x-oem/prod27594.html;.

18. ASUS Xtion Pro Live. http://www.asus.com/us/Multimedia/Xtion_PRO_LIVE/;.

19. Velodyne LiDAR. http://velodynelidar.com/lidar/hdlproducts/hdl32e.aspx;.

20. Prosilica Camera. http://www.alliedvisiontec.com/us/products/cameras/gigabit-ethernet/prosilica-gt/gt2750.html;.

21. Quigley M, Conley K, Gerkey B, Faust J, Foote TB, Leibs J, Wheeler R, Ng AY. ROS: an open-source Robot Operating System. In: International Conference on Robotics and Automation; Open-Source Software workshop

22. Leonard J, Durrant-Whyte H. Simultaneous map building and localization for an autonomous mobile robot. In: Intelligent Robots and Systems' . . . ;

23. Thrun S. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. The International Journal of Robotics Research. 2006;25(5-6):403 429.

24. Dellaert F, Kaess M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. The International Journal of Robotics Research. 2006;25(12):1181 1203.

25. Dellaert F. Factor graphs and GTSAM: A hands-on introduction. GT RIM; 2012. Report No.: September.

26. Segal AV, Haehnel D, Thrun S. Generalized-ICP. In: Robotics: Science and Systems;

27. Rogers J, Fink J, Stump E. Mapping with a ground robot in GPS denied and degraded environments. In: American Control Conference;

28. LaValle SM. Planning algorithms. Cambridge university press; 2006.

29. Likhachev M. Search-Based Planning Library. `https://github.com/sbpl/sbpl`;.

30. Howard TM, Kelly A. Optimal rough terrain trajectory generation for wheeled mobile robots. The International Journal of Robotics Research. 2007;26(2):141 166.

31. Johnson SG. http://ab-initio.mit.edu/nlopt. http://ab-initio.mit.edu/nlopt; 2014.

32. Twigg JN, Fink J, Yu PL, Sadler BM. Efficient Base Station Connectivity Area Discovery. The International Journal of Robotics Research. 2013;.

33. RouterStation Pro. `http://wiki.ubnt.com/RouterStation_Pro`;.

34. Ubiquiti XR2. `http://dl.ubnt.com/xr2_datasheet.pdf`;.

35. RouterStation. `http://wiki.ubnt.com/RouterStation`;.

36. Ubiquti PicoStation2HP. `http://dl.ubnt.com/pico2hp_ds.pdf`;.

37. OpenWRT. `https://openwrt.org/`;.

38. Better Approach to Mobile Ad-Hoc Networking (B.A.T.M.A.N.). `http://www.open-mesh.org/projects/open-mesh/wiki`;.

Intentionally left blank.

Appendix A. Trial Results

This section contains experimental results for each of the reported trials. In each trial, we present 1) the visualization as seen by each of the mission operators; 2) the corresponding RSSI data processed for additional analysis; and 3) plots indicating the measured communications reliability and rate of images collected for each robot. Note the blue lines in (c) and (d) of each trial correspond to the PackBot and the red lines correspond to the Husky.

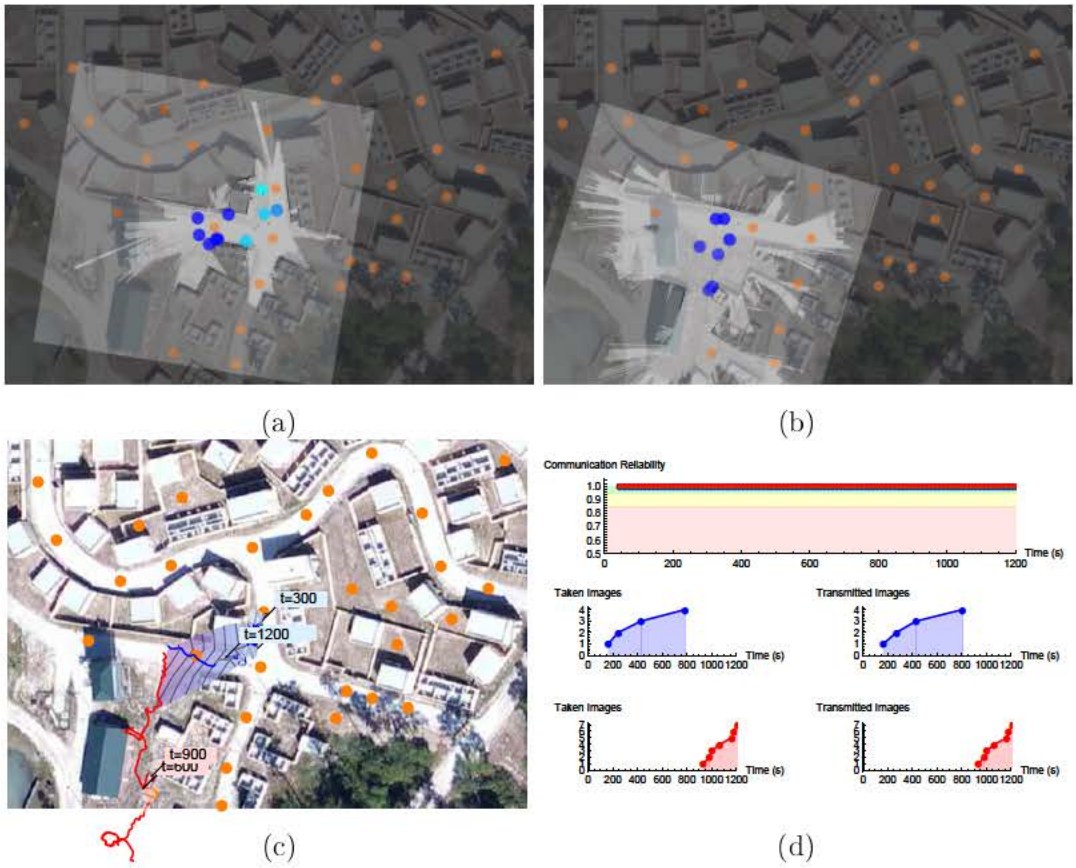


Fig. A-1 Experimental trial 3. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment

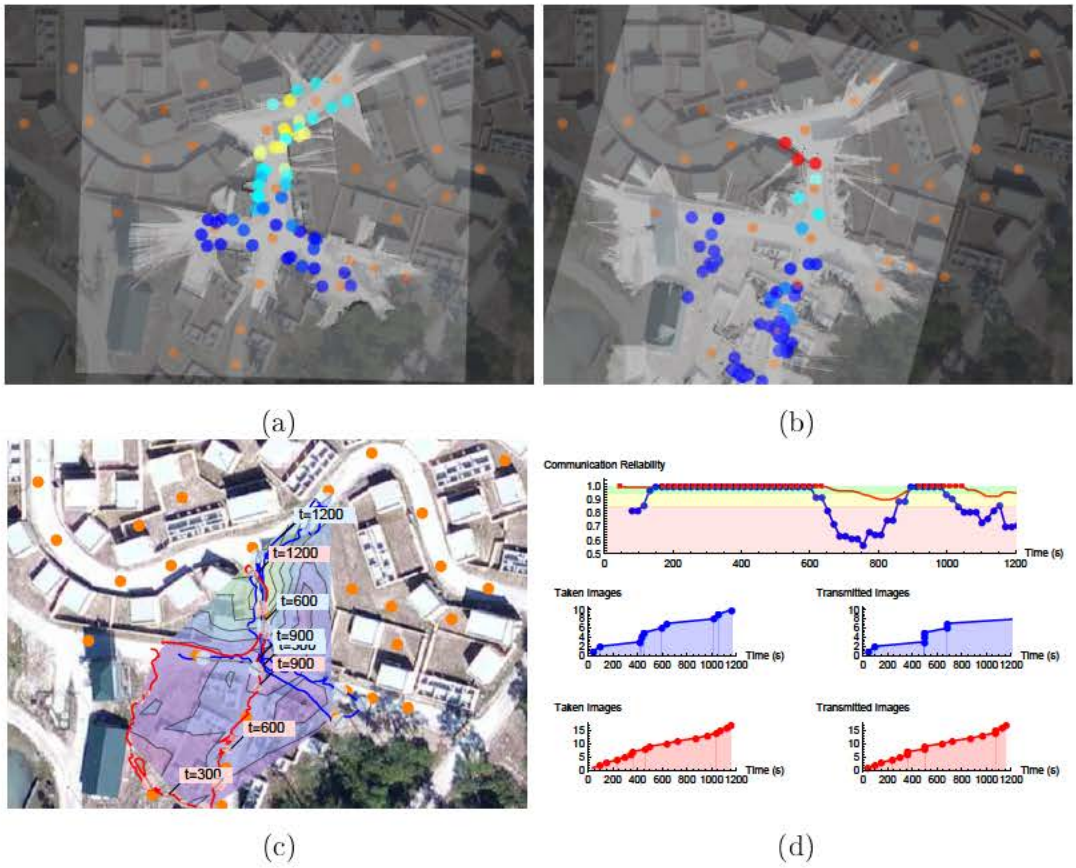


Fig. A-2 Experimental trial 4. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment

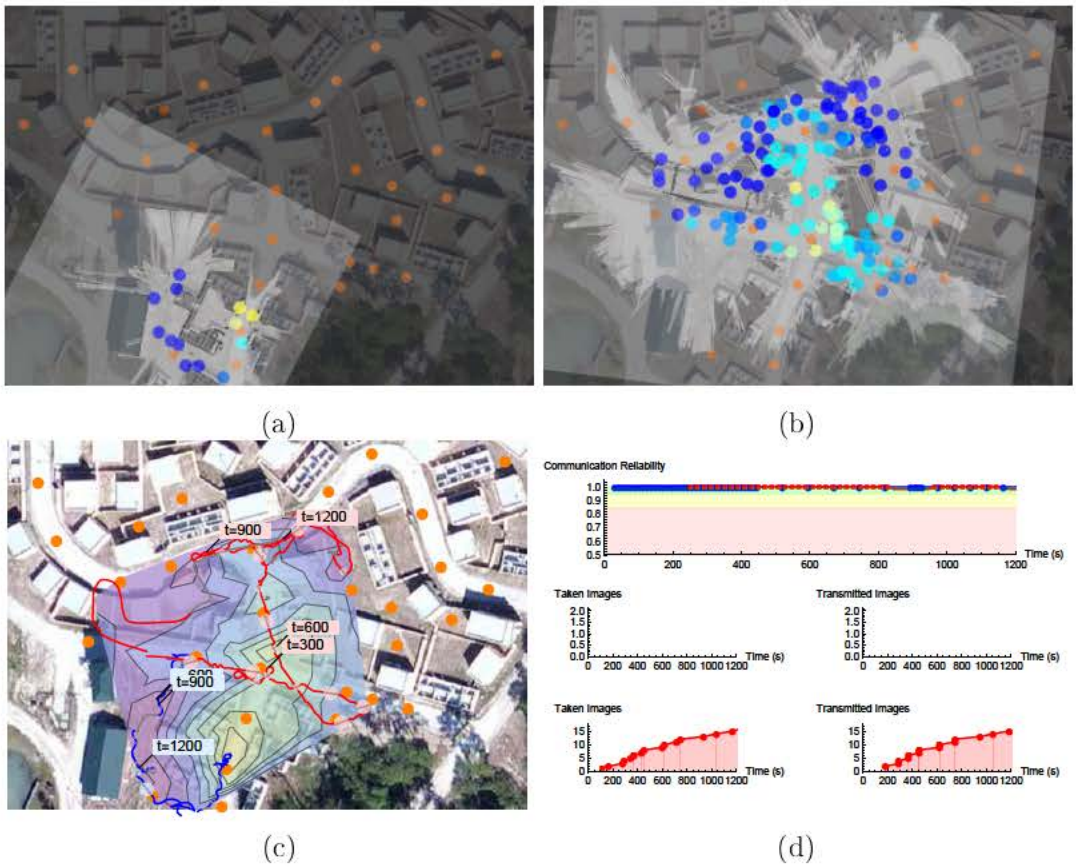


Fig. A-3 Experimental trial 5. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment. Note, in this experiment, the signal source from the mock VIP is accurately localized

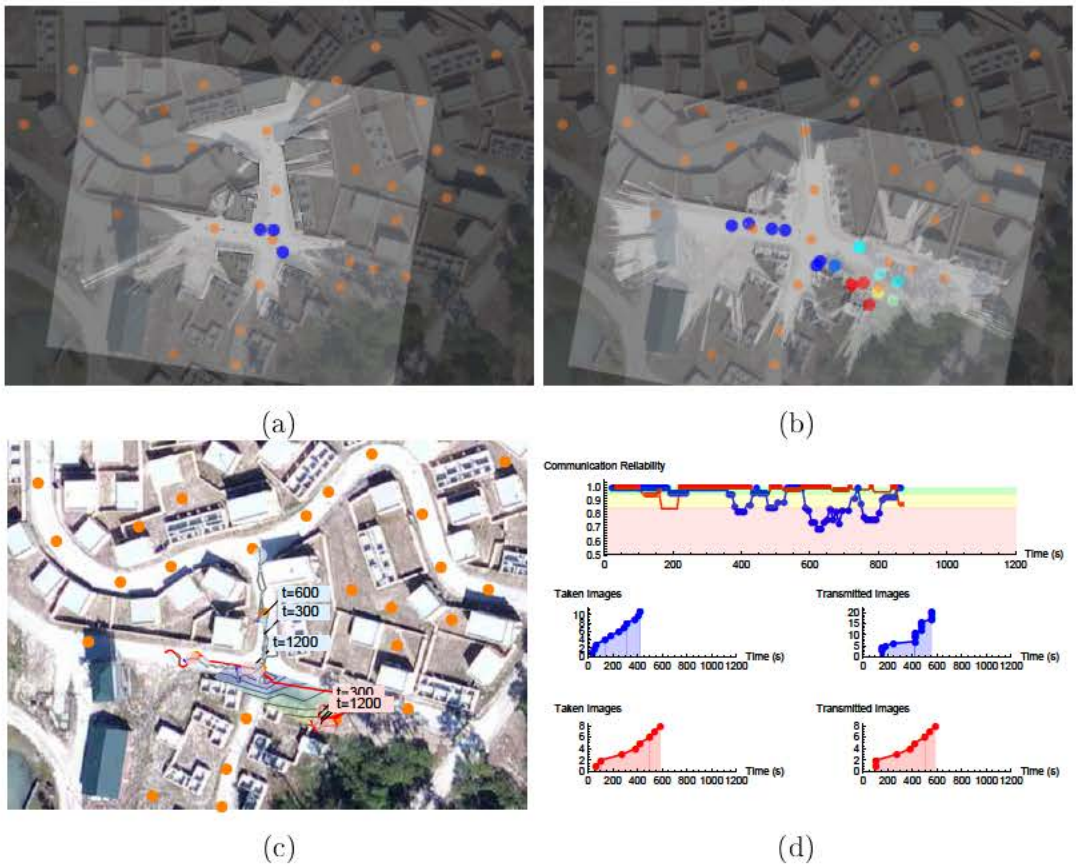


Fig. A-4 Experimental trial 7. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment. Note, in this experiment, the signal source from the mock VIP is accurately localized

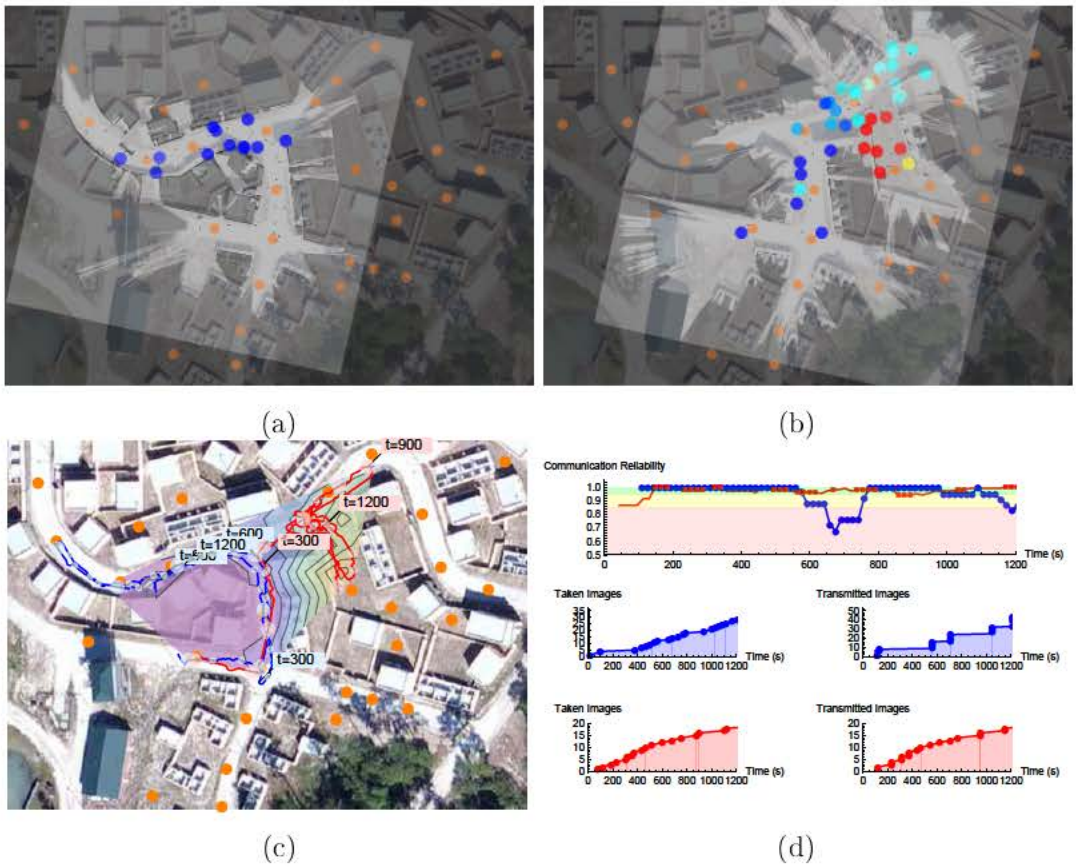


Fig. A-5 Experimental trial 8. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment. Note, in this experiment, the signal source from the mock VIP is accurately localized

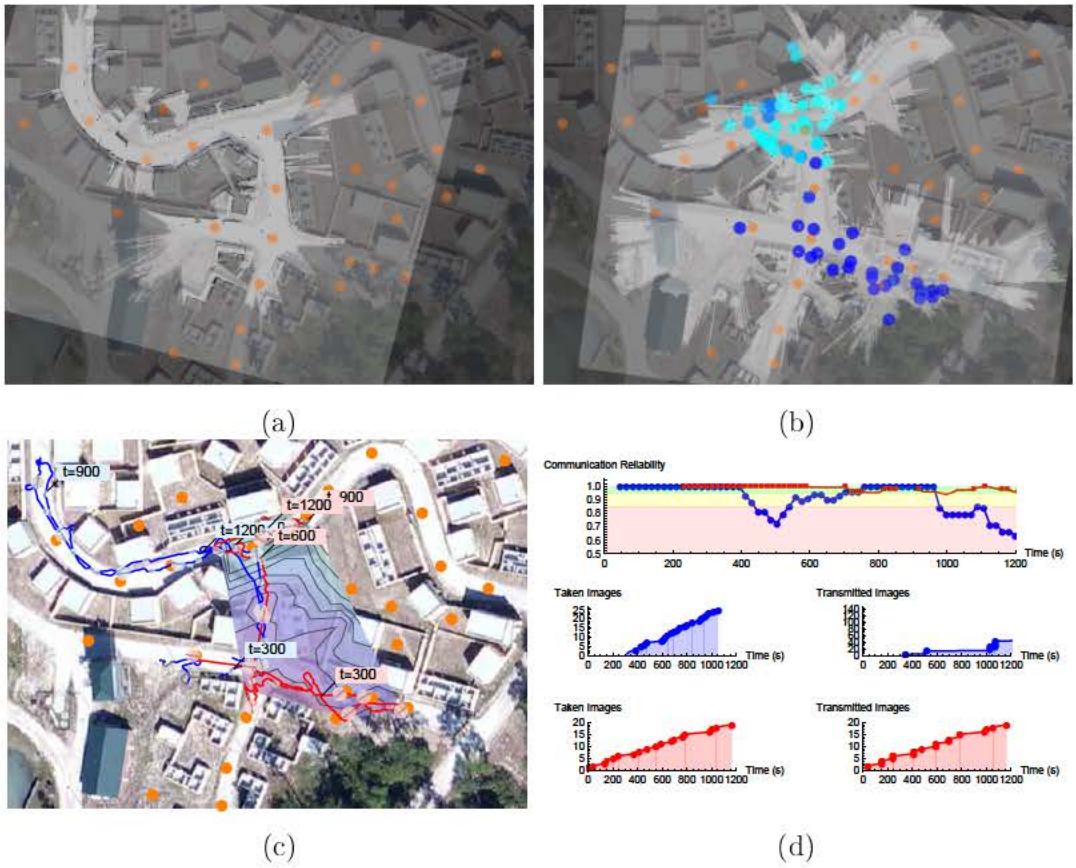


Fig. A-6 Experimental trial 9. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment

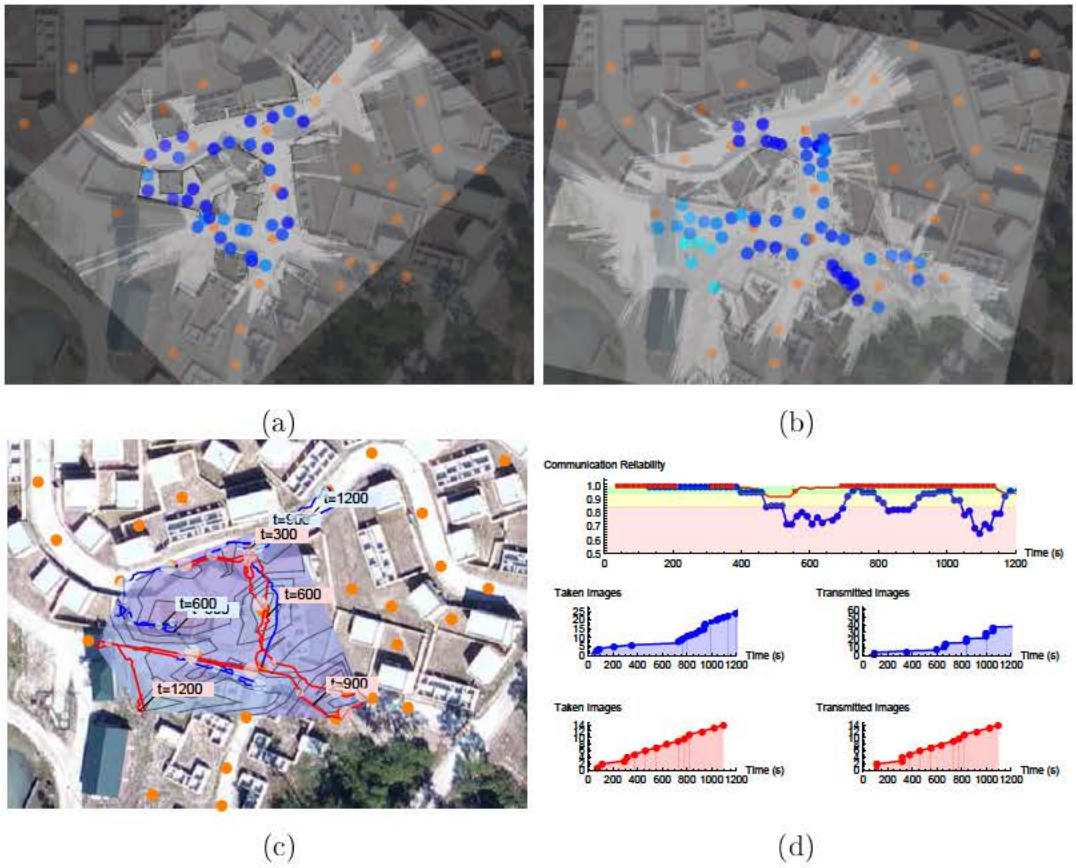


Fig. A-7 Experimental trial 10. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment

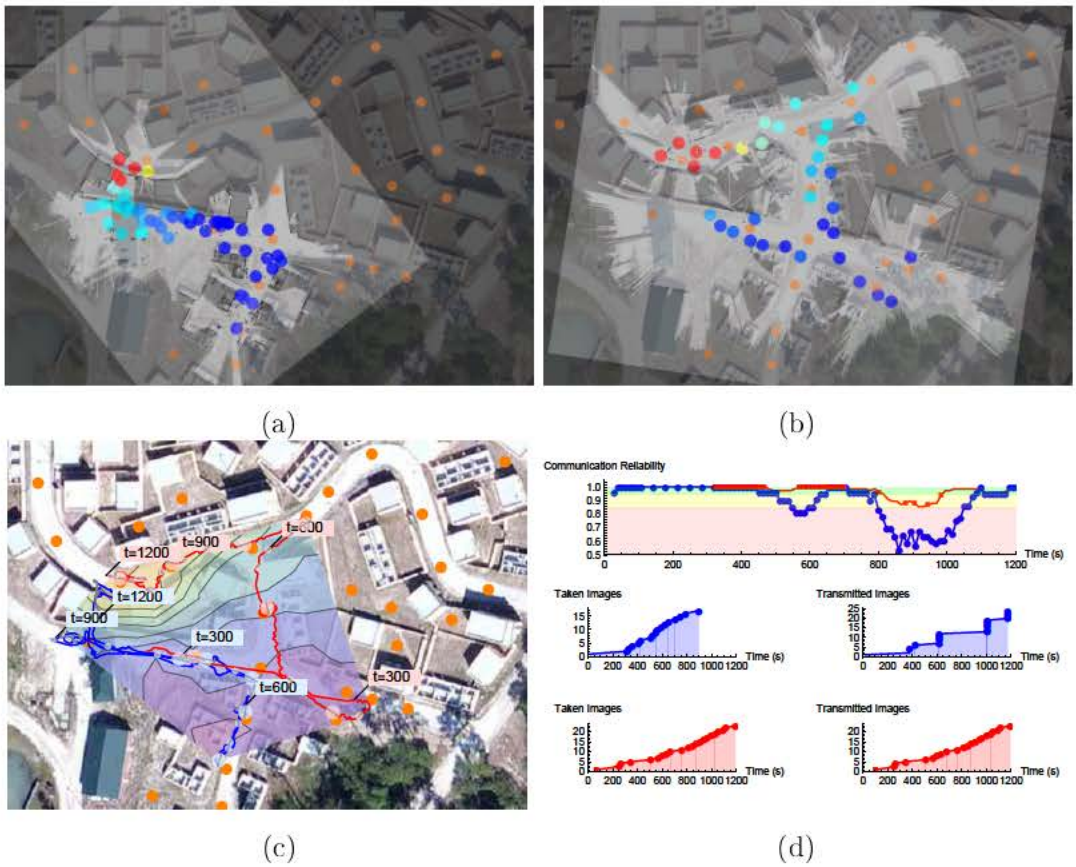


Fig. A-8 Experimental trial 11. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment. Note, in this experiment, the signal source from the mock VIP is accurately localized

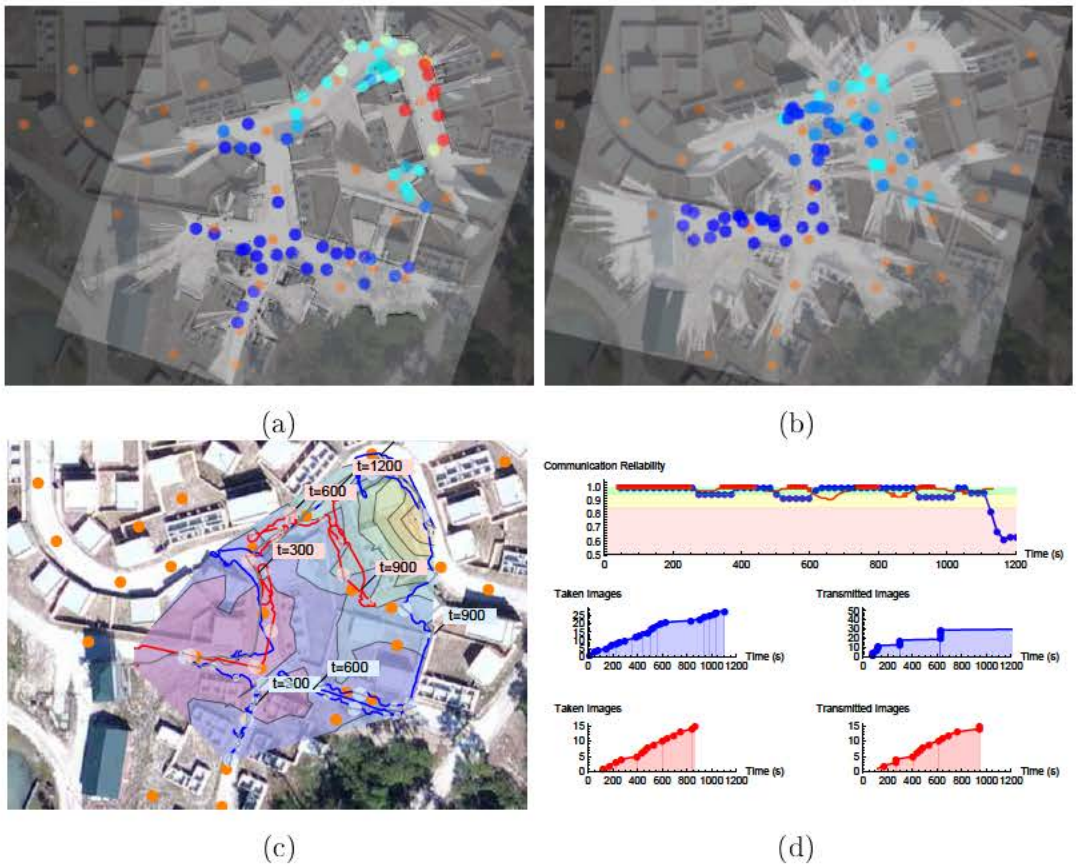


Fig. A-9 Experimental trial 12. (a) and (b) depict the operators' maps that were built by the PackBot and Husky, respectively. (c) depicts the trajectories driven by each robot with interpolated received signal strength data overlayed. In this colormap, red indicates stronger signal strength. (d) depicts the communications reliability and rate of collected images throughout the experiment. Note, in this experiment, the signal source from the mock VIP is accurately localized

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
9 (PDF)	US ARMY RESEARCH LAB RDRL CII A BARBARA BROOME STUART YOUNG JASON GREGORY JONATHAN FINK ETHAN STUMP JEFFREY TWIGG JOHN ROGERS DAVID BARAN NICHOLAS FUNG
3 (HC)	US ARMY RESEARCH LAB RDRL CII A BARBARA BROOME STUART YOUNG JASON GREGORY