Application of Multi-Robot Systems to Disaster-Relief Scenarios with Limited Communication

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

Abstract In this systems description paper, we present a multi-robot solution for intelligence-gathering tasks in disaster-relief scenarios where communication quality is uncertain. First, we propose a formal problem statement in the context of operations research. The hardware configuration of two heterogeneous robotic platforms capable of performing experiments in a relevant field environment and a suite of autonomy-enabled behaviors that support operation in a communication-limited setting are described. We also highlight a custom user interface designed specifically for task allocation amongst a group of robots towards completing a central mission. Finally, we provide an experimental design and extensive, preliminary results for studying the effectiveness of our system.

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 [24]. For example, the use of robots in the aftermath of the Fukushima Daiichi nuclear disaster has been well documented [19, 25], and analysis of the response suggests that action at one of several "inflection points" of the crisis would have probably averted further catastrophe [31] if those actions had not been deemed too dangerous at the time. Partly inspired by these implications, the 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

Jason Gregory
U.S. Army Research Laboratory, Adelphi, MD,e-mail: jason.m.gregory1.civ@mail.mil

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 U.S. Department of Defense [34] 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 [14] and the MAGIC 2010 competition [26, 15]. From these activities, we learn that, although physical platform capabilities 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 establishes a preliminary formal problem description that places an HA/DR-inspired, information-gathering mission in an operations research context (Sec. 2). The primary contribution of this work is to provide documentation and analysis of a multi-robot system capable of performing intelligence-gathering tasks in communications-limited, disaster-relief scenarios. We present the design of such a system (Sec. 3), a set of autonomy-enabled behaviors that can be used to address the HA/DR mission in a relevant environment (Sec. 4), and a user interface that allows a human operator to task the system (Sec. 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 (Sec. 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 two specific problems: the evaluation 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 two 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 with details discussed below.

If we considered only the problem of efficiently visiting 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 [35]. The environment limitations suggest one 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 Capacitated Team Orienteering Problem (CTOP): as a TOP but with a constraint on the total reward that any individual vehicle may gather on a single trip [13].

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 [30]. If we assign a distribution to these rewards initially or as the mission progresses, there is prior work on solving TOPs with stochastic rewards [36] that could apply.

The last challenge is to incorporate the effects of unreliable autonomy, which we model 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 [16] that could apply.

Our preliminary formal problem formulation is thus as a Capacitated Team Orienteering Problem 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 solving it in a receding-horizon fashion as information is uncovered. Developing these upper-bounds for this experiment remains future work.

3 Experimental Multi-Robot System

We present a heterogeneous, multi-robot system with a rich sensor suite, composed of hardware and software components for autonomous operations in relevant environments. 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 robustly operate in real-world scenarios.

3.1 Hardware

Two robotic platforms are used in this work: an iRobot PackBot [8] and a Clearpath Robotics Husky [3]. The PackBot, seen in Figure 1(a), 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 3D point cloud data by nodding a Hokuyo UTM-30LX-EW LiDAR [5] 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) [6] mounted on a custom-made vibration isolator. Additionally, a Garmin 18x PC GPS sensor [4] is elevated on a mast in an effort to receive better GPS measurements. Finally, an ASUS Xtion Pro Live provided RGB data [1].

The second robot used in this work, the Clearpath Husky seen in Figure 1(b), 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 two Quad-Core Intel i7 Mini-ITX processing payloads, each with a 256 GB SSD. The Husky has a Velodyne HDL-32E LiDAR [12], 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 CCD color camera [9].

Both robots use Ubuntu 14.04 (Trusty) and leverage the open-source *Robotics Operating System* (ROS) Indigo [27] to support higher-level algorithms for mapping, navigation, and autonomous capabilities.

To provide the necessary wireless connectivity, we utilize off-the-shelf *IEEE 802.11.g* radios operating in the 2.4 GHz frequency band and capable of 28 dBm transmit power. The PackBot and Husky are equipped with Ubiquiti RouterStation Pro and PicoStation2HP respectively [11]. Each wireless radio operates in *AdHoc* mode and runs of the open-source embedded Linux distribution *OpenWRT* [7] with end-to-end connectivity supported by the *B.A.T.M.A.N.* mesh routing protocol[2]. 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 and placed the "base station" in an advantaged location, i.e., a tower approximately

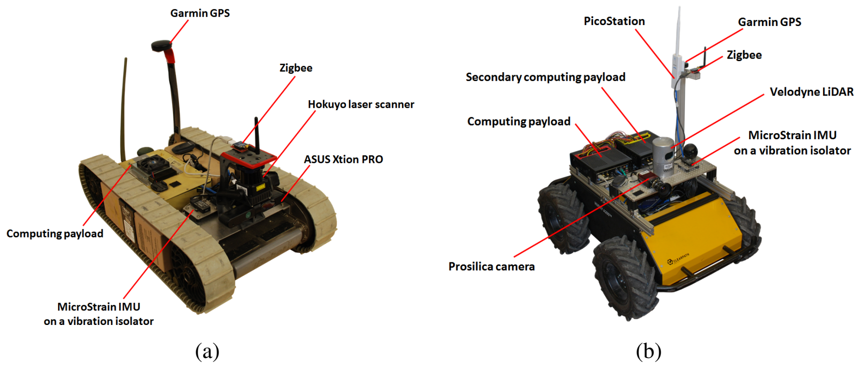


Fig. 1 The hardware configurations of (a) the iRobot PackBot and (b) the Clearpath Husky.

20 m above the ground [10]. 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 experimental facility that 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.

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 one or more beacons. We use a low-power IEEE 802.15.4 XBee radio, shown in Fig. 2, to broadcast a beacon once per second at 2.4 GHz. Each robot also carries a XBee radio and records radio signal-strength when it successfully receives packets from the beacon while traversing the environment in pursuit of the other data-collection tasks.

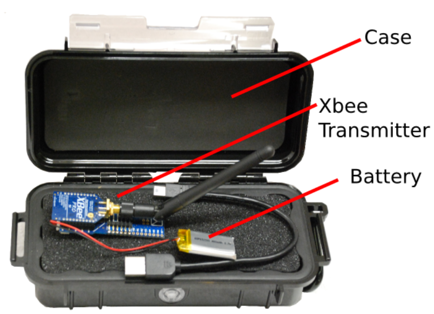


Fig. 2 XBee "beacon signal" transmitter with protective case.

3.2 Mapping

The simultaneous localization and mapping (SLAM) problem focuses on the requirement for precise, consistent knowledge of the robot's trajectory as it gathers sensor measurements and has been studied for some time in the robotics literature [22, 32]. We adopt a modern *graph-based* solution to the SLAM problem based on the square-root smoothing and mapping ($\sqrt{\text{SAM}}$) technique [17] and the *GT-SAM* software library developed at the Georgia Institute of Technology[18]. Our technique leverages the Generalized Iterative Closet Point (ICP) algorithm [29] 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 [28].

We refer to our SLAM system as *OmniMapper* due to its ability to integrate sensor data from a variety of sensor sources including laser scanners and 3D cameras. 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. The *OmniGraph* solves for the robot's optimal trajectory using the *GTSAM* library; the frontend tasks of data association and generating relevant measurements is handled by the *OmniCache*. 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 two primary purposes. First, it can respond to queries about the relative pose of two 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 Figure 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.

3.3 Navigation

We use a three-stage architecture, consisting of a global motion planner, a local planner, and a local controller, to drive our software design within the ROS framework. Each stage of the navigation system depicted in Fig. 3 is implemented as a node,

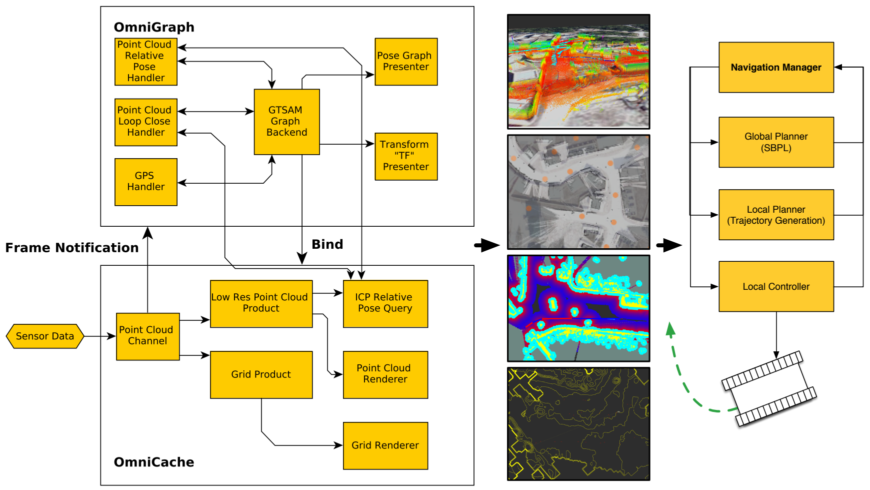


Fig. 3 Architecture for autonomous mapping and navigation.

or independent software process, which provides an *ActionServer* interface that responds to an abstraction of the navigation problem. *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. 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. In this formulation, 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. 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.

For this experiment, we rely on the Search-Based Planning Library (SBPL) [23] to perform global planning actions. We generate a custom set of motion primitives based on our platform's kinematics and use of 0.2 m 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 its path 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. Based on prior work in trajectory generation [20], 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 [21]. 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.

4 Behaviors Supporting Autonomy

In this section, we describe how we build automata to sequence basic capabilities of our multi-robot system in order to provide higher-level autonomous actions and begin to address the data-collection mission described in Sec. 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. 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 returning to the *safe* region.

With the addition of a simple *Collect* action that causes the robot to capture and store an image, the operator can immediately begin to address the data-collection mission from Sec. 2. By specifying a sequence of *goal* regions with accompanying *Collect* actions, 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 all of 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 graphical user interface (GUI) that enables a human operator to task one 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.

For this work, we used RViz to display a top-down orthographic view of satellite imagery of our experimental facility, predefined GPS locations throughout the site, the occupancy grid produced by the 3D mapping techniques described in Sec. 3.2, and the current positions of all the robots during a mission. 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 Sec. 4. After defining a graph in RViz, the system runs a verification to ensure that there are one or more *goal* regions and only one *safe* region for each task. The mission definition is then communicated to each robot where the resulting state machine is executed.

As each robot drives near the radio beacon marking the location of an injured VIP, it will successfully receive transmissions and be able to record the signal strength. 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

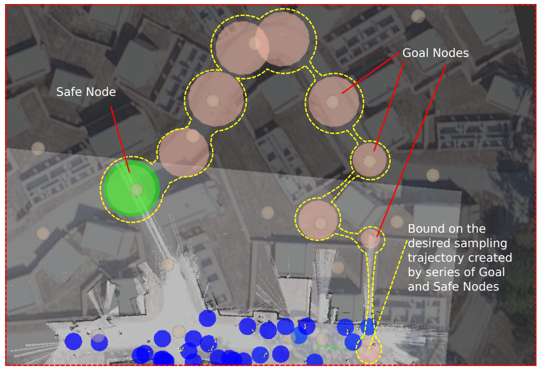


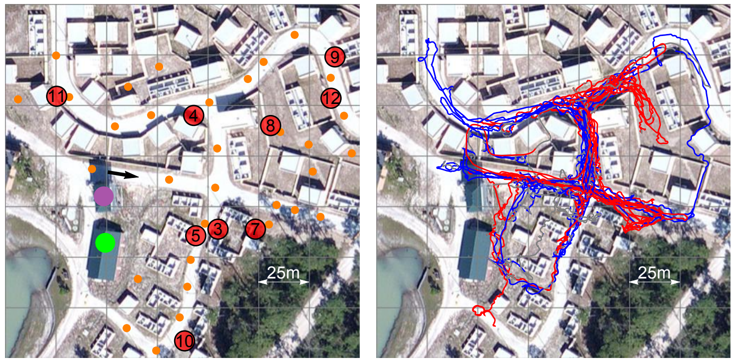
Fig. 4 An example of the user interface for a single data-collection task in a trial. The map is overlaid on top of a satellite image with small pink disks representing the predefined GPS mission nodes. The blue disks indicate that the robot has measured poor received signal strength data thus far. The 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.

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 [33]. 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.

6 Experimental Results

We conducted a series of experimental trials using the 175 x 175 m environment pictured in Fig. 5 to evaluate the capability of our system to address missions defined according to the problem statement in Sec. 2. Each experiment consisted of one or two 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 minutes. 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 minute 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 au-



(a) (b)

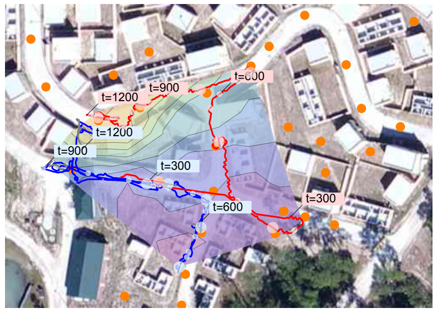
Fig. 5 A satellite overview of the experimental facility overlaid with (a) experiment annotations (green: operating center, purple: elevated base station antenna, orange: mission-specified sites, red: VIP location for each trial) and (b) the aggregated paths driven by robots over all trials.

Trial	*Interventions*		*Intervention Distance (m)*		*Autonomous Distance (m)*		*Percent of Mission Autonomous*		Sites	*VIP Localization Error (m)*
	PackBot	Husky	PackBot	Husky	PackBot	Husky	PackBot	Husky		
3	17	4	14.5	2.6	101.2	167.1	87.5%	98.5%	4	60
4	20	7	51.6	21.9	386.4	336.4	84.1%	93.9%	13	15
5	22	9	34.2	77.9	175.5	494.4	83.7%	86.4%	9	3
7	5	1	9.9	0.5	162	169.9	94.2%	99.7%	7	0
8	10	6	25.6	1.7	334.3	378.4	92.9%	99.6%	15	2
9	8	16	26.1	15.2	403.1	371.4	93.9%	96.1%	17	45
10	13	5	61.5	0.1	454.4	426.2	88.1%	99.9%	15	53
11	24	0	48.1	0.0	446.3	342.7	90.3%	100.0%	12	0
12	13	11	107.0	125.7	605.9	326.0	85.0%	72.2%	17	8

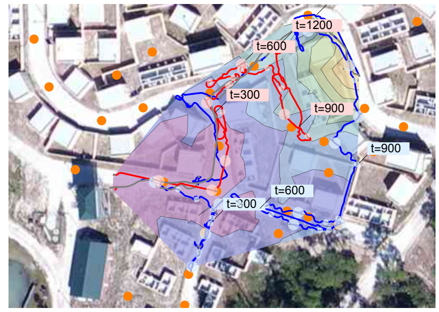
Table 1 Results from each experimental trial.

tonomous 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.

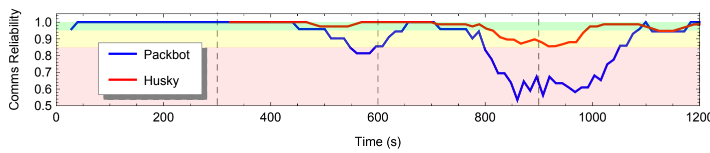
We report on the results of 9 experimental trials with respect to the number of sites visited and mock VIP localization accuracy in Table 1. The trajectories traversed by both robots across all experiments are overlaid in Fig. 5(b) to depict the breadth of experiments conducted. In most experimental trials, the robots drove more than 90% of their total distance while autonomously executing *Guarded-Navigation*-based sub-missions designed by the human operators to gather high-resolution images and VIP signal strength data.



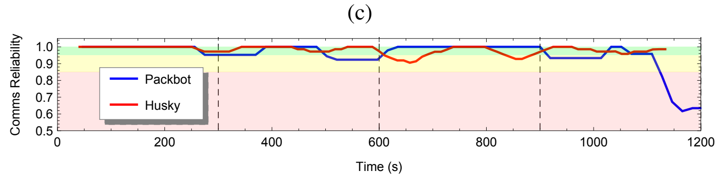
(a)



(b)



(c)



(d)

Fig. 6 Experimental trials 11 (a) and 12 (b). Robot trajectories are shown for the PackBot (blue) and Husky (red). The colormap indicates interpolated signal strength from the VIP beacon (red indicates high signal strength). The communication reliability for trials 11 and 12 are depicted in (c) and (d), respectively, where background colors indicate teleoperation (green), command (yellow), and position-only (red) communication thresholds.

Figure 6 depicts the trajectories of both robots, sites visited, and measured VIP signal strength for two specific examples of experimental trials. Note that in both of these trials, in addition to visiting a number of sites and collecting images, signal-strength data were collected that provide good estimates of the VIP beacon location. Indeed, in trial 11 an image of the VIP was captured, providing the system operator with direct evidence as to the VIP's location and well-being.

Figures 6(c) and 6(d) depict the reliability of operator communication with each robot during experiments as measured by analysis of the reception of periodic diagnostic packets sent by each robot to the operating center. For the purposes of these experiments, we define three levels of communication – reliability exceeding 95% allows for teleoperation, within 85% - 95% robot sub-missions can be commanded and map data are updated after some delay, and below 85% provides no guarantee on useful communication but robot position data may occasionally be available. In all experimental trials, the use of sub-mission specifications using the *Guarded-Navigation* capability allowed operators to task robots routinely into regions of the environment with 85% - 95% reliable communication and, in several cases, enabled collection of data in the 0% - 85% reliability regime.

7 Conclusion

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.

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. 4. Encoding the intelligence that goes into incorporating this a priori knowledge will be key to the application of autonomous planners that schedule the collection mission specifications for multiple robots operating in challenging environments. 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.

References

[1] ASUS Xtion Pro Live. `http://www.asus.com/us/Multimedia/Xtion_PRO_LIVE/`.

[2] Better Approach to Mobile Ad-Hoc Networking. `http://www.open-mesh.org`.

[3] Clearpath Robotics Husky. `http://www.clearpathrobotics.com/husky/`.

[4] Garmin GPS. `https://buy.garmin.com/en-US/US/oem/sensors-and-boards/gps-18x-oem/prod27594.html`.

[5] Hokuyo LiDAR. `http://www.hokuyo-aut.jp/02sensor/07scanner/download/products/utm-30lx-ew/`.

[6] MicroSrain IMU. `http://www.microstrain.com/inertial/3DM-GX3-25`.

[7] OpenWRT. `https://openwrt.org/`.

[8] PackBot. `http://www.irobot.com/For-Defense-and-Security/Robots/510-PackBot.aspx`.

[9] Prosilica Camera. `http://www.alliedvisiontec.com/us/products/cameras/gigabit-ethernet/prosilica-gt/gt2750.html`.

[10] RouterStation. `http://wiki.ubnt.com/RouterStation`.

[11] Ubiquiti Networks. `http://www.ubnt.com`.

[12] Velodyne LiDAR. `http://velodynelidar.com/lidar/hdlproducts/hdl32e.aspx`.

[13] C Archetti, D Feillet, a Hertz, and M G Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6):831–842, May 2008.

[14] Stephen Balakirsky, Stefano Carpin, Alexander Kleiner, Michael Lewis, Arnoud Visser, Jijun Wang, and Vittorio Amos Ziparo. Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11-12):943–967, 2007.

[15] Jonathan Butzke, Kostas Daniilidis, Aleksandr Kushleyev, Daniel D Lee, Maxim Likhachev, Cody Phillips, and Mike Phillips. The university of pennsylvania magic 2010 multi-robot unmanned vehicle system. *Journal of Field Robotics*, 29(5):745–761, 2012.

[16] Ann M. Campbell, Michel Gendreau, and Barrett W. Thomas. The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186(1):61–81, May 2011.

[17] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, December 2006.

[18] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report September, GT RIM, 2012.

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

[20] T M Howard and A Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, 26(2):141–166, 2007.

[21] Steven G. Johnson. *The NLopt nonlinear-optimization package*. http://ab-initio.mit.edu/nlopt.

[22] JJ Leonard and HF Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1991.

[23] Maxim Likhachev. Search-Based Planning Library. `https://github.com/sbpl/sbpl`.

[24] Robin R Murphy. *Disaster Robotics*. MIT Press, 2014.

[25] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.

[26] Edwin Olson, Johannes Strom, Ryan Morton, Andrew Richardson, Pradeep Ranganathan, Robert Goeddel, Mihai Bulic, Jacob Crossman, and Bob Marinier. Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792, 2012.

[27] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation*, Open-Source Software workshop, 2009.

[28] JG Rogers, JR Fink, and EA Stump. Mapping with a ground robot in GPS denied and degraded environments. In *American Control Conference*, 2014.

[29] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: Science and Systems*, 2009.

[30] Cyrill Stachniss and Wolfram Burgard. Exploring unknown environments with mobile robots using coverage maps. In *IJCAI*, pages 1127–1134, 2003.

[31] Eliza Strickland. 24 hours at fukushima. *IEEE Spectrum*, 48(11):35–42, 2011.

[32] S. Thrun. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429, May 2006.

[33] J. N. Twigg, J. Fink, P. L. Yu, and B. M. Sadler. Efficient Base Station Connectivity Area Discovery. *The International Journal of Robotics Research*, July 2013.

[34] U.S. Department of Defense. Foreign humanitarian assistance. Joint Publication 3-29, January 2014.

[35] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, February 2011.

[36] Taylan lhan, Seyed M. R. Iravani, and Mark S. Daskin. The orienteering problem with stochastic profits. *IIE Transactions*, 40(4):406–421, February 2008.