

# Multi-UAV Exploration with Limited Communication and Battery

Kyle Cesare, Ryan Skeelee, Soo-Hyun Yoo, Yawei Zhang and Geoffrey Hollinger

**Abstract**—We propose a multi-robot exploration algorithm that uses adaptive coordination to provide heterogeneous behavior. The key idea is to maximize the efficiency of exploring and mapping an unknown environment when a team is faced with unreliable communication and limited battery life (e.g., with aerial rotorcraft). The proposed algorithm utilizes four states: explore, meet, sacrifice, and relay. The explore state uses a frontier-based exploration algorithm, the meet state returns to the last known location of communication to share data, the sacrifice state sends the robot out to explore without consideration of remaining battery, and the relay state lands the robot until a meeting occurs. This approach allows robots to take on the role of a relay to improve communication between team members. In addition, the robots can “sacrifice” themselves by continuing to explore even when they do not have sufficient battery to return to the base station. We compare the performance of the proposed approach to state-of-the-art frontier-based exploration, and results show gains in explored area. The feasibility of components of the proposed approach is also demonstrated on a team of two custom-built quadcopters exploring an office environment.

## I. INTRODUCTION

The reduced cost and increased reliability of autonomous vehicles and sensing technology has made it possible to field multi-robot teams capable of mapping unknown environments. High-impact applications include urban search and rescue, military reconnaissance, and underground mine rescue operations. The algorithm presented here is applicable across application domains, but is primarily motivated by the use of autonomous rotorcraft to explore indoor environments (e.g., buildings, caves, and mines). In such scenarios, clutter and rubble often make the use of ground vehicles impractical, and fixed-wing aircraft are not appropriate because of tight spaces and obstacles. Small autonomous rotorcraft (e.g., quadcopters) are capable of highly-agile operation within confined spaces and are not impeded by ground obstacles [1]. As such, they are uniquely suited for such exploration tasks.

A key limitation of small autonomous rotorcraft is that they typically have limited battery life (often on the order of tens of minutes) [2]. This limitation restricts their operating lifespan and makes it challenging to complete complex exploration missions in a limited amount of time. The use of

K. Cesare and R. Skeelee contributed equally to this article and share first authorship.

The authors are with the Robotic Decision Making Laboratory in the School of Mechanical, Industrial & Manufacturing Engineering, Oregon State University, Corvallis, OR 97331 USA Email: cesarek@onid.orst.edu, skeeleer@onid.orst.edu, yoos@onid.orst.edu, zhanyawe@onid.orst.edu, geoff.hollinger@oregonstate.edu

This work was supported in part by the following grants: NSF IIS-1317815 and ONR N00014-14-1-0509.

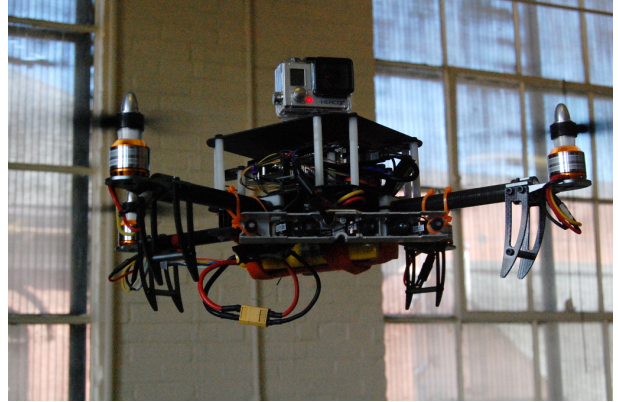


Fig. 1: Unmanned aerial vehicle (UAV) flying in an indoor environment. Our proposed algorithm allows UAVs to “sacrifice” themselves by continuing to explore even when they do not have sufficient battery for returning to the base station. The resulting information is then transmitted back to the base station through the use of relay UAVs.

multiple rotorcraft is one method of overcoming this limitation because their combined flight time may be sufficient to complete the mission. However, many indoor environments create restrictions on communication between vehicles (e.g., WiFi range and signal strength limitations), which makes it difficult to communicate between team members. Finally, the scenarios of interest do not allow for instrumenting the environment with wireless access points or cameras to improve communication or localization. The combination of these factors motivates the design of multi-robot coordination techniques that allow for adaptive, heterogeneous behavior to improve the exploration efficiency of the team.

The proposed algorithm utilizes four states – explore, meet, sacrifice, and relay – to improve exploration efficiency in such scenarios. When in the explore state, a robot uses an existing frontier-based exploration algorithm to generate goals for the robots. When in the meet state, a robot returns to a previous location in attempt to rendezvous with team members. Finally, in the sacrifice state, a robot continues exploring even when it no longer has sufficient battery life to return to the base station. Once the robot has nearly exhausted its battery, it may land to act as a communication relay.

The proposed algorithm is the first to provide heterogeneous exploration capabilities with limitations on communication and battery life. The key novelties of this paper are: (1) a multi-robot coordination algorithm that provides sophisticated relay and sacrifice behavior to adjust for limitations on communication and battery life, and (2) the design and implementation of a team of low-cost autonomous rotorcraft

capable of multi-robot exploration in indoor environments.

The remainder of this paper is organized as follows. We first discuss related work in multi-robot exploration and highlight the need for an adaptive, heterogeneous approach (Section II). Then, we formulate the multi-robot exploration problem (Section III) and present the proposed coordination algorithm (Section IV). Next, we present simulations demonstrating the benefit of our proposed approach versus a frontier-based exploration algorithm (Section V), and we demonstrate an implementation on a team of custom-built autonomous quadcopters (Section VI). Finally, we conclude and discuss avenues for future work (Section VII).

## II. RELATED WORK

This research in robot coordination focuses on the difficulties of real-world environment mapping with autonomous robots. There is often unreliable communication between robots, making coordination difficult. While previous research has focused on exploration algorithms [3], [4] and maintaining mesh networks [5], [6], we focus on coordinating with unreliable communication. Additionally, power limitations cause difficulty when implementing previous research on physical systems [2]. Finite power limits the usefulness of exploration robots in large environments. Our research addresses these limitations in developing an optimized control algorithm.

Prior work has examined a number of algorithms for coordinating robots to explore environments, including using stochastic differential equation solutions [7]. Path planning for aerial vehicles has also been considered in conjunction with state estimation [8], [9]. Coordination algorithms have been applied to autonomous ground vehicles in real-world environments [10]. However, these prior works do not consider heterogeneous behaviors that allow for improved efficiency with unreliable communication and limited battery life. Our work bridges this gap.

The proposed algorithm is compared to a frontier-based exploration baseline [3]. While there is extensive research in exploration algorithms, frontier-based exploration was chosen because it allows for fully distributed operation. Prior work has shown that frontier-based algorithms perform competitively with alternative approaches for indoor exploration tasks [11]. We also note that our proposed algorithm may use any exploratory algorithm, such as market-based approaches [12], as its core baseline (see Section IV). Previous research [3] requires each robot to return home for collection of its map of the environment. This constraint wastes precious energy while robots are returning through explored areas. The key contribution of our work is the ability to build on a core exploration algorithm by adding heterogeneous relay and sacrifice behaviors to improve the efficiency of operation.

## III. PROBLEM FORMULATION

We are given the task of coordinating a team of robots to navigate an unknown environment and maximize exploration for their given battery life. We assume the robots are able

to communicate intermittently, limited by obstacles or range. We also assume that sacrificing robots for a gain in exploration is acceptable. The robots only have communication with the base station when they are in range. To be useful, at least one robot must return to the base station at a pre-specified location.

More formally, we are given  $K$  robots with limited battery life. We denote the battery life for robot  $k$  as  $B_k$ . A robot may stop moving (land) to conserve battery. The goal is to explore the maximal area of a bounded unknown environment and then relay that information back to a base station. We assume that the environment is 2.5D, in that the area that must be mapped is 2D, but the robots may fly some distance above the ground.

Our assumptions on the robots and environment follow closely those described in [3]. We assume a bounded planar workspace  $\mathcal{W} \subseteq \mathbb{R}^2$ . The workspace is divided into free regions  $\mathcal{W}_{free}$  and obstacle regions  $\mathcal{W}_{obs}$ . The partition of the workspace into obstacle and free is initially unknown to the team. The areas with unknown and known partition status to a robot  $k$  at time  $t$  are denoted as  $\mathcal{W}_{unknown}^k(t)$  and  $\mathcal{W}_{known}^k(t)$  respectively. The goal is to reveal the maximal subset  $\mathcal{W}_{known}$  and relay that information back to a base station.

Each robot is modeled as a disk of radius  $p$ , whose configuration  $q_k$  is described by its Cartesian center. The particular kinematics of the robots are not considered in order to focus on the coordination algorithm. Each robot is path controllable (i.e., it can follow any path in its configuration space with arbitrary accuracy). The robots are equipped with an omnidirectional sensor which allows them to explore (and map) the environment around them.

The optimization problem can be stated as follows:

$$\mathcal{P} = \underset{\mathcal{P} \in \Psi}{\operatorname{argmax}} A_r(\mathcal{P}) \text{ s.t. } |\mathcal{P}_k| < B_K \quad \forall k, \quad (1)$$

where  $\Psi$  is the space of possible team paths,  $A_r(\mathcal{P})$  is the area explored and communicated back to the base station by a set of trajectories  $\mathcal{P}$ , and  $|\mathcal{P}_k|$  is the battery consumed by a trajectory  $\mathcal{P}_k$  for robot  $k$ .

## IV. COORDINATION ALGORITHM

First, we assign the robots sequential identification numbers. We define four possible states for an individual robot: explore, meet, sacrifice, and relay. The current state is dependent on the robot's ID number, remaining battery life  $\beta$ , battery required to return home  $B$ , time since the last meeting with any other robot  $t$ , and predicted distances from other robots.

All robots begin in the exploration state. This state uses a frontier-based exploration strategy, as described in [3], [4], though any algorithm could be substituted. The basic idea is for the robots to build their own local maps and extract "frontiers" between the explored space and the unknown space. The heuristic of moving to the closest frontier is used here, which has been shown to provide competitive performance with other heuristics [11]. The robots also share

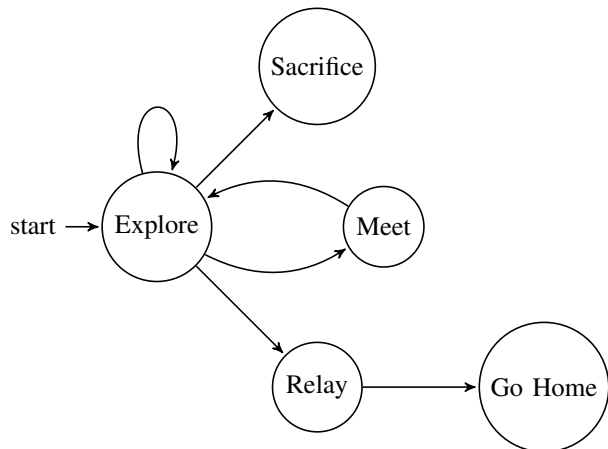


Fig. 2: Proposed state diagram for the adaptive exploration algorithm. The explore, meet, sacrifice, and relay states allow for heterogeneous behavior that adjusts for limitations on communication and battery life.

their maps and merge them using existing map merging techniques (see Section VI). If two robots move towards frontiers that are near each other, we impose a conflict resolution step where the robot with the higher priority ID number takes the frontier, and the other robot chooses the next-closest frontier. Collision avoidance is handled in a similar manner where the robot with the lower priority stops or lands and allows the higher priority robot to pass. This approach allows for fully distributed operation.

After a constant time  $T$  (set as a parameter), a robot will transition into the meet state. In this state, a robot will attempt to travel into communication range of another robot to transmit map information and to update its internal state of other robots’ locations. If it is able to meet with a robot, it will negotiate relay and sacrifice roles. The robot with the lower ID of the two marks itself as a relay and the higher ID robot marks itself as the sacrifice. These flags will be used when the robots determine to enter the relay or sacrifice state. Any meeting with another robot occurring at a later time will overwrite these flags.

If a robot is not a relay, it will transition into the sacrifice state when it determines it has just enough battery to make it to the last position it sighted its relay. It will then travel to that location in an attempt to meet with the relay. Should it not find its relay at the predicted position, it will revert to an exploration mode in a final attempt to locate the robot. If a robot is a relay, it will eventually reach a point at which it has only enough battery to travel to the base station. At this point, the robot will land (or go into a hibernation state) to conserve battery while it waits for its sacrificial robot to find it. Once it is found by the other robot, it will travel back to the base station.

These state changes are made independently of other robots’ states, which leads to a fully distributed architecture. Algorithm 1 gives a summary of the proposed algorithm, and Figure 2 shows a state diagram for the explore, meet, sacrifice, and relay states.

---

### Algorithm 1 Exploration coordination algorithm

---

```

1: Inputs: current battery  $\beta$ , time between meetings  $T$ 
2:  $t \leftarrow 0$ 
3: while  $\beta > 0$  do
4:    $B \leftarrow$  battery required to go home
5:   Transmit with nearby robots
6:   if  $\beta \leq B$  then
7:     if is relay then
8:       Land until sacrifice robot meets
9:     else
10:      Travel to last meeting location
11:    end if
12:  else if  $t \geq T$  then
13:    Travel to last meeting location  $p$ 
14:    if robot found then
15:      Update relay and sacrifice flags
16:    else if at  $p$  then
17:      Revert to exploration
18:    end if
19:     $t \leftarrow 0$ 
20:  else
21:    Explore local frontiers
22:  end if
23:   $t \leftarrow t + 1$ 
24: end while

```

---

## V. SIMULATED RESULTS

Simulations were used to test the proposed coordination algorithm. Simulations were run over two maps (shown in Figure 3) with varying battery levels and starting positions. The quadcopters are capable of moving at a speed of 1 m/s in simulation, and the parameter  $T$  was empirically set to one minute.

We compared the proposed algorithm to a baseline frontier-based exploration approach [3]. The frontier-based exploration approach is equivalent to remaining in the “explore” state and then returning to the base station once battery is low. Thus, the simulations demonstrate the improvement from utilizing the meet, sacrifice, and relay behaviors. Figure 4 shows the results of these simulations demonstrating that the proposed algorithm is able to explore a greater percentage of the map than the baseline algorithm under varying conditions. The improvement of the proposed algorithm ranges from 5% to 18%. We note that in all cases the proposed algorithm provides some improvement over the frontier-based exploration baseline.

In the complex environment, the improvement from the sacrifice, meet, and relay behavior increases as the number of robots increases. This result is expected since increasing team sizes means that more robots can be sacrificed for additional exploration. We show the results for a large team (8 robots) in the simple environment, which provides substantial benefit over the frontier-based exploration baseline. These improvement demonstrate that the proposed algorithm provides improvement in both cluttered environments (e.g.,

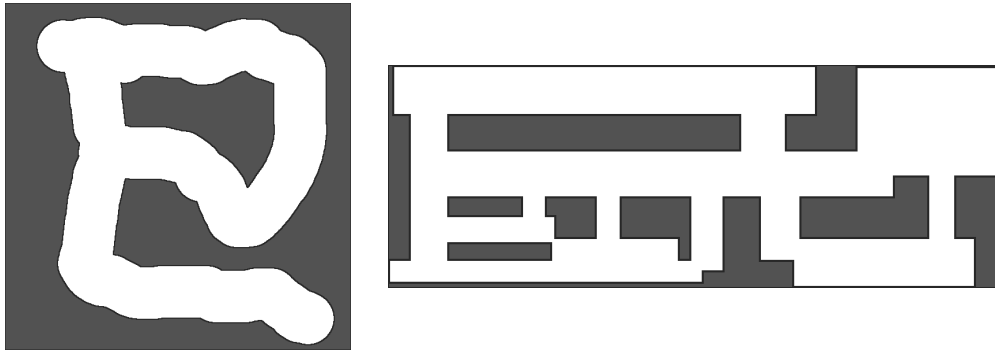
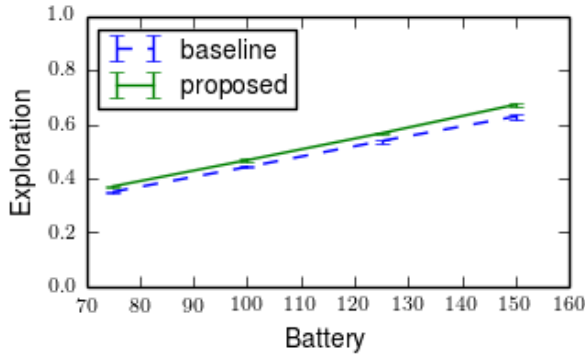
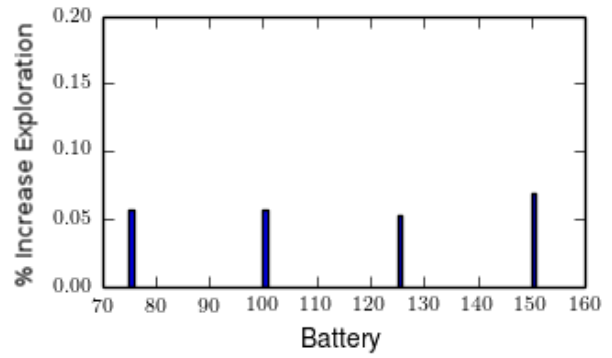


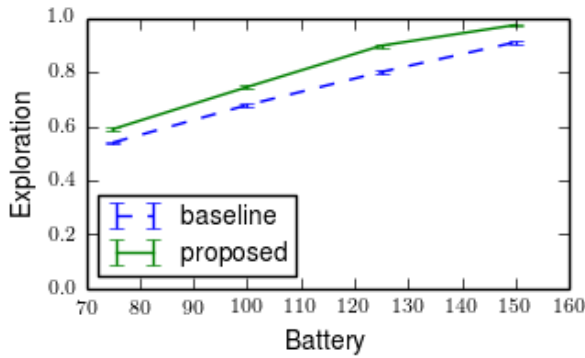
Fig. 3: **Left:** Simple environment map used for multi-robot exploration trials. The environment is approximately 50 m × 50 m. **Right:** More complex environment map used for multi-robot exploration trials (120 m × 40 m).



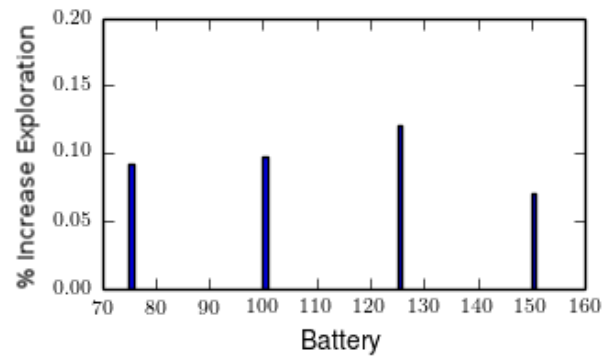
(a) Complex Environment, 2 robots



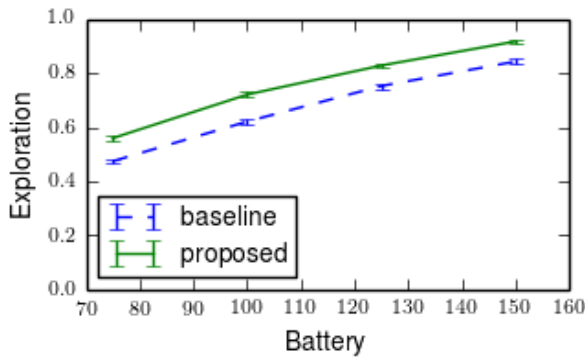
(b) Complex Environment, 2 robots



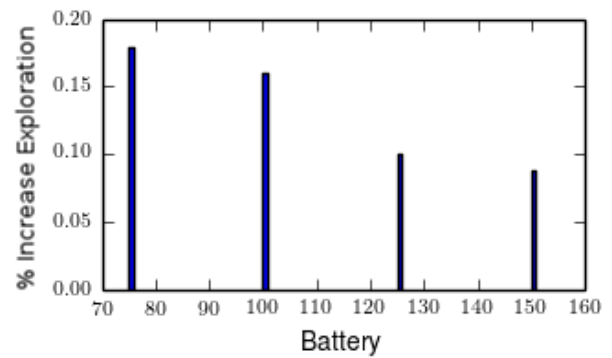
(c) Complex Environment, 4 robots



(d) Complex Environment, 4 robots



(e) Simple Environment, 8 robots



(f) Simple Environment, 8 robots

Fig. 4: Simulated algorithm performance. Each data point is an average of 200 simulation runs with random starting points. Error bars are one standard error of the mean.

office buildings) and more open environments (e.g., caves).

## VI. EXPERIMENTS ON AUTONOMOUS QUADCOPTERS

The proposed algorithm was demonstrated in an office environment using two autonomous quadcopters. The system used to test our robot coordination algorithm is described below. Design requirements are identified as follows. All processing must be performed onboard the system, and the system must be capable of full autonomy once in the air. Each robot must be able to communicate and coordinate with additional robots. We also design for low cost (less than \$1,500 per robot). Existing robots are either substantially more expensive or require an instrumented environment [2].

The system must have sufficient computing power to run vision and planning algorithms in real-time. A Gigabyte Brix i7-4500 was chosen to meet our power, weight, and computational performance requirements. A PX4FMU is used for flight stabilization, and a PX4Flow camera provides optical flow measurements. An Asus Xtion acts as the main onboard camera, and is capable of providing RGB and depth images.

### A. Software Architecture

The software architecture is split into two systems: the high-level processing running on the onboard computer and the low-level flight stabilization on the flight control board. The onboard computer runs Robot Operating System (ROS) [13] on top of Ubuntu 13.10. The OpenNI ROS package interfaces directly with the Xtion camera to publish RGB and depth information, which is processed by the RGBDSLAMv2 package. The latter package localizes the vehicle in its environment and produces a point cloud map of the area. To improve the usability of this map, the point cloud is passed to the OctoMap package to produce a 3D probabilistic occupancy tree. We then use the ROS navigation stack for 2D path planning and frontier exploration.

The flight control board accepts local velocity commands from the onboard computer over the MAVLink protocol. It then interprets these commands based on its internal velocity estimate provided by the flow camera and inertial measurement unit. Beyond this point, we treat the flight control software as a black box.

### B. Experiments

We successfully demonstrated the key components of our proposed algorithm on a team of two quadcopters. A sacrifice-relay handoff was implemented using one sacrificial quadcopter (denoted as SQ) to perform frontier based exploration in a new room while a relay quadcopter (denoted as RQ) traveled to a meeting location and then returned with data of the explored room. A colored OctoMap of the passage that RQ mapped is shown on the left in Figure 6, and the additional explored area mapped by SQ is shown on the right.

The experiment proceeded as follows: (1) both RQ and SQ started at the same home base location; (2) SQ and RQ were both moved manually (due to safety considerations in

the experiment space) to the entrance of the unexplored area where RQ landed as a relay; (3) SQ was flown to a stable altitude and switched to autonomous frontier-based exploration until it registered that it was nearly out of battery; (4) SQ landed and transmitted its map data to RQ, (5) RQ was returned to the home base with the gained map information from both SQ and RQ. This proof-of-concept demonstration shows successful exploration and mapping capabilities on the quadcopter hardware as well as a successful handoff of information between UAVs acting in the relay and sacrifice roles. Additional hardware experiments are also shown in this paper's supplemental video attachment.

## VII. CONCLUSIONS

This paper has demonstrated coordinated exploration using a multi-UAV team with unreliable communication and limited battery life. Our results show that the proposed algorithm, which leverages meeting, sacrificing, and relaying behavior, increases the percentage of the environment explored. By capitalizing on relay behavior, the mapped environment returned to the base station was shown to be more complete than a baseline frontier-based exploration strategy. We have also demonstrated our approach on a team of two autonomous quadcopters, which were designed to be low-cost, fully autonomous, and capable of operating without pre-installed infrastructure.

It is important to note that the mapping capabilities of the quadcopters use off-the-shelf algorithms available in the ROS package. It may be beneficial to improve the mapping capabilities using more sophisticated distributed smoothing and mapping approaches [14]. Additional future work lies in examining the effect of parameter selection in various types of environments (e.g., tunnels, caves, mines, etc.). Finally, we plan to scale up this approach to larger teams to achieve exploration in larger-scale environments.

## REFERENCES

- [1] S. Shen, N. Michael, and V. Kumar, "Obtaining liftoff indoors: Autonomous navigation in confined indoor environments," *IEEE Robotics and Automation Mag.*, vol. 20, no. 4, pp. 40–48, 2013.
- [2] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, E. T. K. Yoshida, K. Ohno, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *J. Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.
- [3] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatronics*, vol. 14, pp. 163–175, 2009.
- [4] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. Int. Conf. Autonomous Agents*, 1998, pp. 47–53.
- [5] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams," *Proc. IEEE*, vol. 100, no. 1, pp. 164–178, 2012.
- [6] B. J. Thibodeau, A. H. Fagg, and B. N. Levine, "Signal strength coordination for cooperative mapping," DTIC Document, Tech. Rep., 2005.
- [7] S. Shen, N. Michael, and V. Kumar, "A stochastic differential equation-based exploration algorithm for autonomous indoor 3D exploration with a micro-aerial vehicle," *Int. J. Robotics Research*, vol. 31, no. 12, pp. 1431–1444, 2012.
- [8] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous path planning and estimation using UAVs," *IEEE Robotics and Automation Mag.*, June 2009.

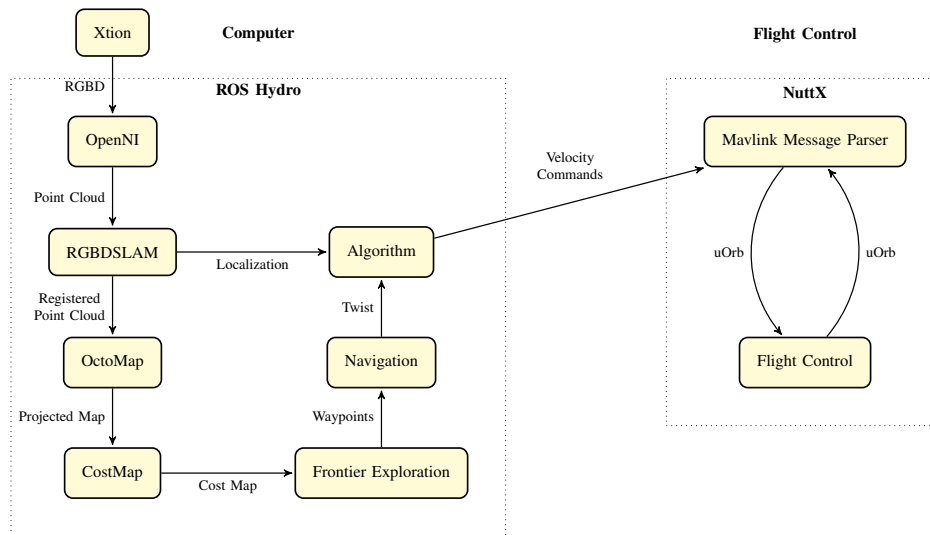


Fig. 5: Software architecture for the custom-built autonomous quadcopter system.

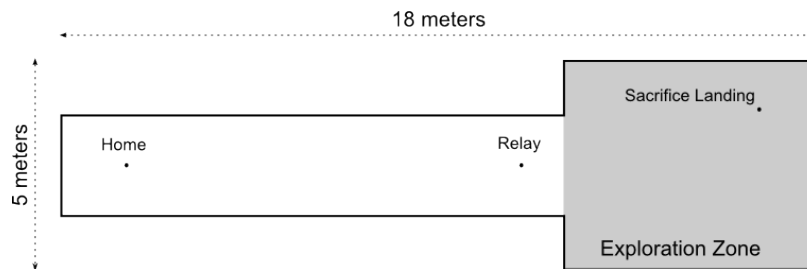
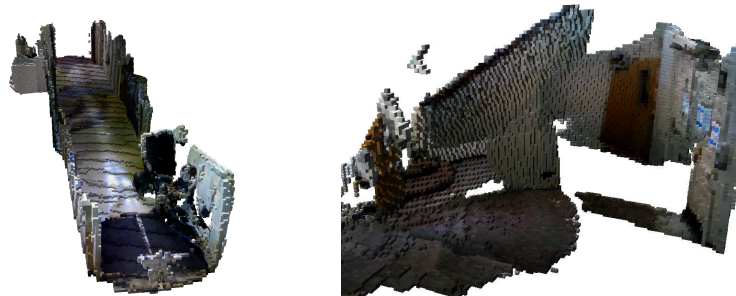


Fig. 6: **Left:** Colored OctoMap from relay quadcopter, leading to unexplored area. **Right:** Colored OctoMap from sacrificial quadcopter showing additional explored area. **Bottom:** An overview of the testing area.

- [9] S. Shen and N. Michael, "State estimation for indoor and outdoor operation with a micro-aerial vehicle," in *Proc. Int. Symp. Experimental Robotics*, Quebec City, Canada, 2012.
- [10] C. Stachniss, *Coordinated multi-robot exploration*. Springer, 2009, vol. 55, pp. 43–71.
- [11] M. Julia, A. Gil, and O. Reinoso, "A comparison of path planning strategies for exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.
- [12] N. Kalra, "A market-based framework for tightly-coupled planned coordination in multirobot teams," Ph.D. dissertation, Robotics Institute, Carnegie Mellon Univ., 2006.
- [13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.
- [14] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2013, pp. 5200–5207.