

# Mapping Planetary Caves with an Autonomous, Heterogeneous Robot Team

**Ammar Husain**  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
ammahr@andrew.cmu.edu

**Heather Jones**  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
hlj@cs.cmu.edu

**Balajee Kannan**  
GE Global Research  
Schenectady, NY 12309  
Balajee.Kannan@ge.com

**Uland Wong**  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
uland.wong@gmail.com

**Tiago Pimentel Martins da Silva**  
Mechatronics Engineering Dept.  
University of Brasilia  
Brasilia, DF, Brazil  
tiagopms@gmail.com

**Sarah Tang**  
Mechanical and Aerospace Eng. Dept.  
Princeton University  
Princeton, NJ 08544  
sytang@princeton.edu

**Shreyansh Daftry**  
Electronics and Communication Dept.  
Manipal Institute of Technology  
Manipal, Karnataka, India  
shreyanshdaftry219@gmail.com

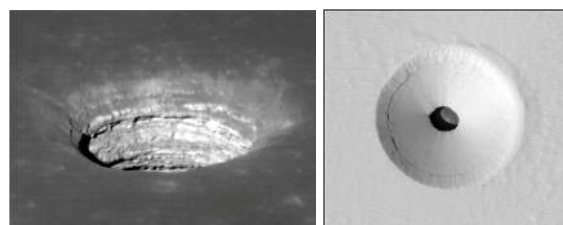
**Steven Huber**  
Astrobotic Technology, Inc.  
Pittsburgh, PA 15213  
steve.huber@astrobotictech.com

**William L. “Red” Whittaker**  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
red@cmu.edu

**Abstract**—d Caves on other planetary bodies offer sheltered habitat for future human explorers and numerous clues to a planet’s past for scientists. While recent orbital imagery provides exciting new details about cave entrances on the Moon and Mars, the interiors of these caves are still unknown and not observable from orbit. Multi-robot teams offer unique solutions for exploration and modeling subsurface voids during precursor missions. Robot teams that are diverse in terms of size, mobility, sensing, and capability can provide great advantages, but this diversity, coupled with inherently distinct low-level behavior architectures, makes coordination a challenge. This paper presents a framework that consists of an autonomous frontier and capability-based task generator, a distributed market-based strategy for coordinating and allocating tasks to the different team members, and a communication paradigm for seamless interaction between the different robots in the system. Robots have different sensors, (in the representative robot team used for testing: 2D mapping sensors, 3D modeling sensors, or no exteroceptive sensors), and varying levels of mobility. Tasks are generated to explore, model, and take science samples. Based on an individual robot’s capability and associated cost for executing a generated task, a robot is autonomously selected for task execution. The robots create coarse online maps and store collected data for high resolution offline modeling. The coordination approach has been field tested at a mock cave site with highly-unstructured natural terrain, as well as an outdoor patio area. Initial results are promising for applicability of the proposed multi-robot framework to exploration and modeling of planetary caves.

## TABLE OF CONTENTS

1	INTRODUCTION.....	1
2	MULTIROBOT COORDINATION FRAMEWORK ..	3
3	OPERATOR INTERFACE.....	4
4	TASK ALLOCATION.....	5
5	MODEL BUILDING.....	7
6	EXPERIMENTS & RESULTS.....	7
7	CONCLUSIONS & FUTURE WORK.....	10
	ACKNOWLEDGMENTS .....	10
	REFERENCES .....	10
	BIOGRAPHY .....	11



(a) Lunar skylight (b) Skylight in a funnel-shaped pit on Mars

**Figure 1.** Skylights, collapse features that provide entrance into sub-surface caves, are known to exist on the Moon and Mars, and evidence for them has been discovered on other planetary bodies.

## 1. INTRODUCTION

### *Mission Context*

Subsurface caverns may be the best place on Mars for life seeking. They may be the best hope for safe havens and habitation on the Moon. They can provide a window into a planet’s geology, climate, and even biology. Skylights, formed by partial cave ceiling collapse, provide access to subsurface voids. Tunnel entrances have been conclusively shown to exist on Mars [1] and the Moon [2]. There is also evidence supporting their existence on other planetary bodies throughout the solar system [3] (See Figures 1(a) and 1(b)). Despite astonishing discoveries of skylights and cave entrances, and their inevitable exploration, they do not yet appear in NASA’s Decadal Survey. Skylights and the voids below are so unknown that it is too risky to send astronauts to explore them without prior robotic reconnaissance and modeling.

Unlike surface exploration, robotic exploration of skylights and caves is a daunting venture. Exploration of these voids is well matched to the ability of multi-robots to extend data gathering and mapping operations into large spatial areas that otherwise defy access. Multi-robots increase system redundancy and enable distributed capability deployment, where vehicles can have varying sensor/manipulator capabilities to

better achieve a broad range of objectives. Autonomy is imperative due to lack of direct-to-Earth communication in subterranean environments.

The innovation is a heterogeneous multi-robot team for optimized, fault-tolerant subsurface planetary exploration and mapping. Multi-robot configurations offer unique solutions to the challenges of exploration and mapping in subsurface voids. A ‘parent’ robot could provide access to larger section of the void for mobile ‘children’ and serve as a docking station for power and communication through a tethered connection to a surface station. Coverage of complex, challenging terrain is optimal by coupling small and large robots to explore a variety of potential features. Heterogeneous robots could provide efficient mobility in rocky terrain through employing a combination of locomotive methods, like legs and wheels. Additionally, since subterranean features provide protection from radiation hazards, robots that only operate underground need not incur the expense of radiation hardening. This facilitates the use of many low-cost robots in place of a single higher-cost unit. Multiple low-cost robots are more expendable enabling higher-risk exploration in challenging terrain. The robot team requires software to enable autonomous collaboration among large multi-robot groups.

State-of-art for planetary exploration includes line-of-sight, direct-to-overhead communication on single mobile robots on the moon and Mars. These missions succeed with human controllers augmented by some automation. Subterranean exploration precludes these solutions, motivating autonomy and multi-robot systems for efficient execution of mission tasks, such as exploration and mapping. The robotic team can re-configure in novel ways to extend range, increase mapping fidelity, or maintain a communication link. Multi-robot considerations require ground-up design of planetary robots to address unique challenges of robot interactions, such as deployment and docking, communication, and specialization of robots. Existing multi-robot systems in research are the realm of academic development and/or are highly tuned to specific applications on Earth. The majority of these systems operate in controlled or moderate environments, compared to rock-strewn subsurface voids. This work advances beyond the state-of-art to develop a multi-robot team to enable operations in rugged space environments.

Increased exploration efficiency relative to single robots is realized through task parallelization and specialization with multiple robots. A heterogeneous team could include a highly mobile ‘parent’ to enter a subsurface void, a few highly capable rovers to perform science objectives, and numerous low-capability mapping robots.

### ***Related Work***

Study by Werker et al. [4], researched the scientific value of exploring caves on other planets. This research speculated on planetary cave value by comparing to scientific knowledge gained by investigation of terrestrial caves. This study listed devices and infrastructure that are required to execute subsurface planetary exploration. Important aspects include communication networks, biological sensing, and drilling capabilities.

Dubowsky, Iagnemma, & Boston [5], [6] proposed exploration of subsurface voids with a large team of expendable robots. These robots were self-contained spherical hopping robots weighing approximately a 100 g with a 100 mm diameter. The rationale behind this development is that wheeled rovers such as Sojourner or Curiosity are not well suited to

navigate through extremely rough terrain or access highly sloped surfaces anticipated to be present in subsurface environments. Additionally, Dubowsky, Iagnemma, and Boston opted for a large team of small-scale, low-cost robots, as large rovers were deemed too valuable to risk entrapment. This concept is further detailed by Kesner in [7].

In terrestrial applications, multi-robot teams are increasingly being used in functionally-distributed missions. This requires complex coordination among multiple robots performing numerous tasks such as planning, coordination, and information sharing in highly dynamic and potentially hazardous operating environments. To ensure optimum coverage and exploration in the shortest duration possible the robots need to coordinate with one another. Additionally, a well established communication protocol is necessary for coordinating a team of heterogenous robots with differing capabilities. Furthermore, coordination can help compensate for sensor uncertainty and state errors ensuring an overall high quality of solution. How to effectively coordinate heterogeneous teams towards maximizing exploration in unknown environments has been an ongoing challenge in multi-robot research and has been addressed using a variety of techniques [8],[9], [10], [11].

A popular way of handling conflicting interests and desires is via the use of distributed constraint satisfaction or optimization [12], [13]. When optimality is a key concern or uncertainty is very important, researchers have often turned to decision theoretic techniques, with Markov Decision Processes able to find very high-quality plans for big teams. In dynamic, unstructured, non-cooperative settings, like the one considered here, where the goals of the team members are different, market-based approaches have been very popular. Such approaches use a simulated economy where robots buy and sell tasks according to their estimated cost for completion. In fact, Dias et al.[14] demonstrate that path cost estimates that better reflect environmental and mission characteristics, as well as online learning, results in increased system efficiency. This becomes particularly important in heterogenous robotic teams, where capabilities across robots may differ drastically. Dias et al. [15] present the concept of such pick-up teams and emphasize the challenges in efficient task allocation for such teams. The key to many algorithms is for the robots to reason locally about what information is most important and prioritize it on the network. Many algorithms have been developed that minimize communication while maximizing information, but previous work has not taken into account that different robots might have different priorities and capabilities for which information gets on the network.

In summary, most existing multi-robot systems are highly tuned to specific applications on Earth. They operate in controlled or moderate environments, compared to rock-strewn subsurface voids. In the work where planetary subsurface operation is envisioned, details of coordination architecture, task generation, task allocation, and data fusion have not been explored.

### ***Outline***

The paper is organized as follows. Section 2, describes the system architecture for coordinating multi-robot teams and detail its individual components: communication, mapping and planning. Section 3 details capabilities and functions of the Operator Interface, such as data fusion and frontier generation. Section 4 elaborates approach for autonomous task assignment. Data collected during experiments are post-

processed to create high accuracy models using techniques described in Section 5. Finally, Section 6 presents details on the experiments conducted and their findings. Conclusions are presented in Section 7.

## 2. MULTIROBOT COORDINATION FRAMEWORK

This research addresses questions pertaining to appropriate methodology for autonomously generating exploration plans and behaviors for information gathering and monitoring. Towards achieving this mission objective, a team of heterogeneous robots is used to explore and map the subterranean domain. The diversity of the robots varies in terms of their size, mobility, sensing and capabilities towards executing the mission goal. Further, the robots have inherently different low-level behavior architectures, making coordination a challenge. A generalized capability-based framework for dynamic allocation of autonomously generated tasks has been developed.

To ensure successful task completion, a well-defined high-level software architecture that allows ease of coordination and communication across different robotic platforms is needed. Towards that, the open-source Robot Operating System (ROS), the emerging standard software platform for robotics research, is adopted and enhanced as the core software platform for this architecture. ROS is an open-source, meta-operating system that provides a variety of services, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. Further, the ROS software is released under an Open Source license, and the great majority of it is licensed under a BSD-style license that allows users and companies to build applications on top without licensing constraints. The use of ROS enables a platform-agnostic system. This facilitates contributions of current research to coordination, exploration, and mapping. Each robot is a separate ROS master. This research adds a custom communication layer for inter-robot and robot-to-ground communication. It is not required for the on-board intelligence to be ROS-based. A task execution ROS node simply issues motion primitives to the robot. This node is robot specific and designed to issue CGI, IPC or ROS messages for task execution. Additionally the robots possess mapping, planning and communication modules (ROS node) that are robot agnostic. This section further describes the individual modules, their design rationale and capabilities. A custom communication architecture that supports inter-robot and robot-ground relays was developed in this work. A limitation within the ROS framework is the capability for directed messaging (unicast) between different ROS masters. For a lunar relevant project, communication bandwidth could be a major constraint. Therefore a differential messaging pattern dependent on data capability was developed. This is described below.

### Communication Module

The communication module within the multi-robot framework supports two primary functions: information sharing and task assignment.

- *Information Sharing:* The objective of coordinating multiple robots is to develop a framework for shared intelligence. Each robot acts as an intelligent agent capable of executing tasks independently. However, the robots must aid other robots as well as humans in the loop with their perception

of the environment. A design decision in such a scenario is to choose the kind and amount of information that the robots should provide. This decision is mission specific and depends heavily on the sensing capabilities, communication restrictions etc. For the purpose of exploration, information includes low resolution (1 meter) occupancy grid maps, 3-Dimensional local pose  $(x, y, \theta)$ , and initial global transforms between agents. Since the operator should be up to date with robot state, a publish/subscribe routine updates the ground station at 10 Hz. The inter-robot coordination occurs on a request/response basis. An agent lacking perception can poll other robots or the operator for a map of the environment. The reliability in transport of individual data streams of information is not mission critical and is transported via UDP. This is a preferred protocol, since robots may go in and out of the network and a loss of packets does not stall the mission.

- *Task Assignment:* This module conveys task instructions to the robot. The generation and assignment of these tasks (frontiers) are discussed later in Sections 3 and 4 respectively. A reliable TCP data transfer, implemented with the ZeroMQ library in Python, is used for this purpose. If the task assignment is not acknowledged by the robot, it can be reassigned. A directed publish/subscribe messaging pattern is used, proactively assigning tasks, given  $n > m$ , where  $n$  is # of tasks and  $m$  is the # of agents in the system, elaborated further in Section 4.

### Online Mapping Module

Due to limited power and computing resources, only coarse maps are generated on the robots. A resolution finer than 1m grid sized required higher computation time for path planning described later in this section. Additionally, higher resolution maps cannot be transmitted real time to an operator interface, given limited communication bandwidth. Data collected from the experiments was used to create models with higher accuracy as described in Section 5.

The key element in any mapping problem is to calculate the posterior over maps given some control inputs  $(x)$  and sensor measurements  $(z)$ .

$$p(m|z_{1:t}, x_{1:t}) \quad (1)$$

Occupancy grids are an effective method for mapping large amounts of data in 2 dimensions. An occupancy grid is a uniform discretization of space into cells, where each cell value represents the degree of belief that the cell is occupied. Occupancy models partition the problem of estimating the map into a collection of separate problems as described in [16], giving the following equation for the posterior probability.

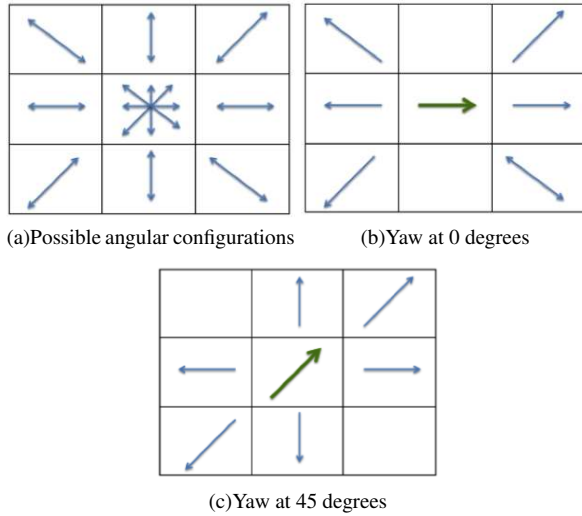
$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t}) \quad (2)$$

One drawback of this representation is that it fails to encode dependencies between neighboring cells. Sensor measurements can then be integrated using the technique introduced by Moravec and Elfes [17].

$$p(m|z_{1:t}) = \left[ 1 + \frac{1 - p(m_i|z_t)}{p(m_i|z_t)} \frac{1 - p(m_i|z_{1:t-1})}{p(m_i|z_{1:t-1})} \frac{p(m_i)}{1 - p(m_i)} \right]^{-1} \quad (3)$$

This estimation can be converted to the *logOdds* form, making computation easier and updates faster. Here a uniform prior  $(P(m_i) = 0.5)$  is assumed.

$$L(m_i|z_{1:t}) = L(m_i|z_{1:t-1}) + L(m_i|z_t) \quad (4)$$



**Figure 2.** Non-Holonomic Configurations

Lastly, in order to overcome any overconfidence in the map, Yguel et al. [18] proposed a *clamping update* policy.

$$L(m_i|z_{1:t}) = \max(\min(L(m_i|z_{1:t-1}) + L(m_i|z_t), l_{max}), l_{min}) \quad (5)$$

In this framework, the belief  $L(m_i|z_{1:t}) \in [1, 255]$  with 1 being vacant. The map was initialized with a numeric value of 0, corresponding to unexplored grids. This belief metric performed well for dynamic obstacles appearing and disappearing from the map. While the environment is expected to remain static for planetary cave exploration, robots can appear as dynamic obstacles to their teammates.

### Planning Module

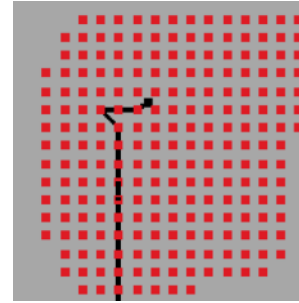
Non-holonomic A\* was used for global planning. The responsibility for local planning is more robot specific and was therefore carried out by the intelligence on the agents. The goals were assumed to have a 360 degree angular tolerance. In other words, a goal was achieved when the robot reached a goal  $(x, y)$  on the map irrespective of its final heading. Hence, the heuristic was simply point to point euclidean distance. In a grid based model of the world, a third discretized dimension of heading was added. Therefore the planner was searching in a 2.5D space, rather than 2D for any holonomic discrete search algorithm.

Figure 2(a) illustrates this concept. The block in the center is the current node and can have 8 possible angular configurations. Yaw angle is snapped to 45 degree increments. Of the surrounding 8 possible nodes, the non-holonomic planner accepts 6 of them as acceptable configurations. The neighboring nodes have fixed angular configurations. This essentially means that the top right node will always accommodate a heading of 45 degrees or 135 degrees. Similarly row 2, column 3 node will always have a yaw angle of 0 degree or 180 degrees. Therefore given a current robot configuration, the robot may either go forward or turn 45 degrees left or right. The same motion is possible in the reverse direction as well. Figures 2(b) and 2(c) highlight the potential nodes that the robot can take, given a heading of 0 degree and 45 degrees respectively. The planner operated on a 1-meter squared grid size.

A perfect reverse motion is not very achievable even for a

very controllable robot. Since most robots have front-facing sensors, going in reverse implies traversing blind. Therefore, a penalty is added to the cost function for pursuing reverse motions. This causes the robot to reverse some amount and change its heading to face the target. This can be seen in Figure 3, where the robot initially reverses out of its position and then turns and moves forward towards the goal. The black line indicates the path to goal and the red points are the nodes explored. The angular and reverse penalties were not incorporated in the heuristic leaving that function optimistic.

Lastly, the goal of the project is to maximize exploration. Though the frontiers generated for exploration are on the periphery of the explored area, the planner must still handle unexplored areas. Therefore a higher penalty is assigned to occupy an unexplored grid, thereby planning a straight path to goal in unexplored space.



**Figure 3.** Planner reverses the robot briefly and turns to face goal

## 3. OPERATOR INTERFACE

In addition to the core coordination framework, an operator interface was developed to enhance operational flexibility and situation awareness. This interface was developed using the FLTK library integrated with a ROS-based robot communication module. The key capabilities of the interface included:

- Getting information updates from the robot.
- Emergency stop all agents and cancel task schedules.
- Provide manual control of an agent to the operator.
- Send task assignments to agents
- Generate frontiers of interest given an area to explore

Figure 4 illustrates the operator interface. The following sections elaborate on map merging and generating frontiers of interest, specifically as they pertain to the user interface.

### Map Merging

Exponentially weighted moving averages fuse new and old information from the robots. A set of robots  $\Theta = \{\theta_1, \dots, \theta_m\}$  produces local occupancy grid maps  $M_l = L(m|z_{1:t}) \in [1, 255]$  as described previously under online mapping in Section 2. The interface merges the local information to produce  $M_G$ . The first update  $u = 1$  is initialized as:

$$m_{1G} = m_l(H_G^{\theta_i} * \theta_i) \quad (6)$$

where  $\forall m \in M$  is a grid in the map and  $H_G^{\theta_i}$  is the transformation matrix to a global frame for robot  $\theta_i$ .

For  $u > 1$ :

$$m_{uG} = \alpha(m_l(H_G^{\theta_i} * \theta_i)) + (1 - \alpha)m_{(u-1)G} \quad (7)$$

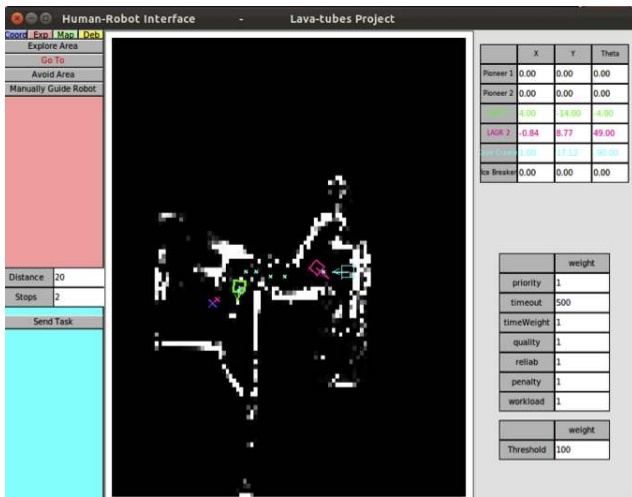


Figure 4. User Interface

where  $\alpha \in [0, 1]$  is the weight associated with the new information.  $\alpha$  is set to 0.333 to avoid skewing the map with outliers such as dynamic obstacles in the  $m_i(\theta_i)$ . Figure 5 shows three robots positioned on a fused map during experimentation at the patio site in Figure 12.

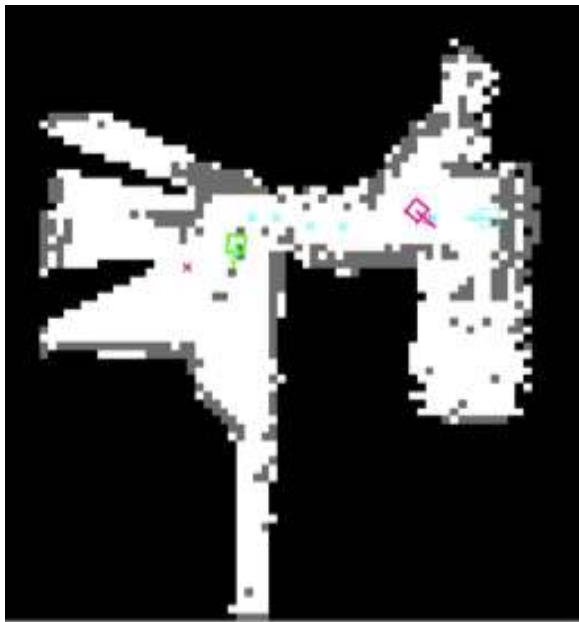


Figure 5. Online Map Merging

### Frontier Based Exploration

Frontiers are robot goal points located on the contour of known and unknown space. The interface is capable of generating frontiers given an operator specified area of interest. In order to maximize coverage in a given time, no robot should be sitting idle while there are areas left to explore. To this end, the interface generates more frontiers  $T = \{t_1, \dots, t_n\}$  than there are robots  $\Theta = \{\theta_1, \dots, \theta_m\}$ ,  $n > m$ . Robots bid on tasks as they are generated and put up for assignment, so they may accept a task before embarking on a prior task. This means that the robot state (i.e. pose) when it accepts a task may be different than its state when it goes to execute that task, creating an  $\mathcal{NP}$ -complete scheduling problem (Travel-

ing Salesman Problem) for each robot  $\theta_i$ . This problem is discussed further in Section 4. The exploration strategy aids in providing a greedy approximation to this problem.

Upon generation of  $n$  frontiers, the  $k$ -means clustering algorithm is applied, where  $k = m$ . As discussed in [19], the algorithm attempts to find sets  $S = \{S_1, \dots, S_k\}$  such that:

$$\operatorname{argmin} \sum_{i=1}^k \sum_{t_j \in S_i} \mathbf{d}(t_j, \mu_i) \quad (8)$$

where  $\mu_i$  is the mean of set  $S_i$  and  $\mathbf{d}(\cdot, \cdot)$  is the euclidean distance between two points. Given the costing dependent approach of both task allocation strategies, described later in Section 4, a robot  $\theta$  will be assigned  $S_i$  set of tasks. Therefore within each  $S_i$  a greedy solution to the travelling salesman problem is found.

If a set  $S_i$  contains  $j$  tasks  $t_j$ ,  $j$  greedy solutions to visit all points in  $S_i$  are found. This is calculated by assigning every possible  $t_j \in S_i$  as the initial task. Finally the lowest cost solution is chosen as the order of tasks in  $S_i$  and pushed for task allocation. Rosenkrantz et al. in [20], proved that this approximation factor is not worse than  $\log(|V|)$  where  $V = j$  is the number of frontiers to explore in the cluster. The cross marks visible on the map in Figure 4 & 5 were the frontiers generated for exploration.

## 4. TASK ALLOCATION

Once a task has been generated by the interface, the next step in the process is to assign it to a robot within the team for execution. The heterogeneous nature of the team dictates that not all robots are capable of executing every task generated by the system. Consequently, task allocation strategies are developed to build on the notion of robot capabilities for executing tasks. A two-fold approach is taken for task execution.

### Basic Allocation Problem

Frontier allocation to maximize exploration can be formalized in terms of a ST-SR-TA Optimal Assignment Problem (OAP) as discussed by Gerkey and Mataric in [11]. Acronyms are elaborated below:

- *Single Task Robots (ST)*: These robots can execute only one task at a given instance of time.
- *Single Robot Tasks (SR)*: Each task (frontier) generated for exploration requires at most one robot for successful completion. In other words, if a robot is sent to probe a feature of interest, it would not require physical aid from another agent. However the robot could request support from other robots in terms of information. The coordination framework developed supports this feature.
- *Time Extended Assignment (TA)*: In order to maximize coverage, the system is always flooded with frontiers to explore. Therefore, an agent cannot guarantee immediate task completion after winning (centralized approach) or accepting (distributed approach) a task. If there were more agents than tasks, or the tasks were extended in temporal assignment, an Instantaneous Assignment (IA) would be possible.

Given the dynamic nature of such a multi-robot system, it is difficult to optimize on a single objective function. An intuitive explanation for this stems from the fact that each robot maintains its own backed up set of tasks that it needs to complete. The costing for these tasks was performed at a different time instance (& most likely different robot state) from the actual execution. Therefore the cost is bound to change depending on the nature of the tasks. For true optimality, each robot must solve the Travelling Salesman Problem which is  $\mathcal{NP}$ -complete [20]. As described in Section 3 under Frontier Based Exploration, this scheduling problem is approximated during frontier generation. Therefore, to simplify computational complexity each robot merely follows a FIFO scheduling system. Given, this simplification the OAP for this multi-robot exploration framework is an instance of the ST-SR-IA class.

The formulation of the multi-robot exploration problem presented here is derived from Scerri et al. in [21]. The coordination framework consists of a set of heterogeneous and homogeneous robots  $\Theta = \{\theta_1, \dots, \theta_m\}$  and a set of tasks with varying capabilities  $T = \{t_1, \dots, t_n\}$ . The capability of a given robot  $\theta_i$  to perform a task  $t_j$  is given by:  $Cap(\theta_i, t_j) \in [0, 1]$ . The  $Cap(\theta_i, t_j)$  could be treated as a continuous function and incorporated as a parameter for optimization, if a quantification of task quality can be performed. Since the task structure in exploration is simple, a binary assignment is used. The robot is either capable or incapable of performing a task. The goal of OAP is to find an allocation matrix  $A$  that minimizes the total team cost  $C$ :

$$A = \underset{t_i \in T, \theta_j \in \Theta}{\operatorname{argmin}} \sum \sum Cap(t_i, \theta_j) * c_{ij} \quad (9)$$

subject to:

$$\sum_{i=1}^m a_{ij} = 1, 1 \leq j \leq n \quad (10)$$

$$\sum_{j=1}^n a_{ij} = 1, 1 \leq i \leq m \quad (11)$$

where  $a_{ij}$  is an element in matrix  $A$ . The cost  $c_{ij}$  used during experimentation was the path length to goal computed using the onboard A\* planner described in Section 2. Towards ensuring an efficient task-allocation process, two types of task allocation strategies are implemented: centralized and distributed.

### Centralized Allocation

A market based approach is used for centralized task allocation. Market-based approaches use a simulated economy where robots buy and sell tasks according to their estimated cost for completion. Each robot in the team is modelled as a self-interested agent, and the team of robots as an economy. Each robot aims to minimize individual cost; however, since all revenue is derived from satisfying team objectives, the robots self-interest equates to doing global good. The estimated costs of tasks are based on a set pre-defined cost functions shared amongst the agents in the market-based system. This allows for a system where robots attempt to minimize their individual costs and consequently the cost of the group as a whole, which in turn maximizes the work performed by the team of robots. Thus, the robots run task auctions, and bid on tasks in other robots' task auctions, with the task being allocated to the robot with the lowest cost.

TraderBots[22] is one such market-based coordination strategy that is particularly well suited to the exploration problem. TraderBots' default cost paradigm involves pairwise cost functions; cost functions that take two tasks, A and B, as arguments and calculate the cost of completing B given the completion of A. Subsequently, the cost of a schedule of tasks can be calculated by running this pairwise cost function through each consecutive pair of tasks and computing the cumulative cost. Schedules are ordered lists of tasks that the robot must perform in sequence based on a defined prioritization criteria. For this system, cost is computed using a distance metric, i.e., the total distance (in meters) that the robot must travel to complete the tasks.

### Distributed Probabilistic Allocation

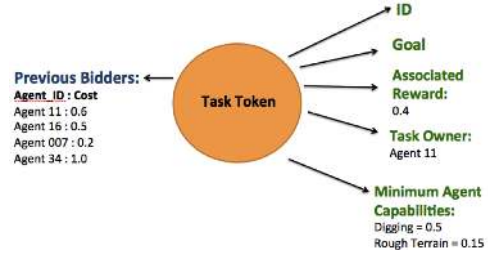


Figure 6. Sample Task Token

Distributed Probabilistic Allocation (DPA) attempts to find an approximate solution to the OAP described earlier in this section. This methodology is inspired by the token based allocation proposed for LA-DCOP in [21]. Scerri et al. proved that incorporating a token system can reduce message transfer requirements by up to six orders of magnitude. This reduction in communication has not yet been quantified using DPA, though similar performance is expected. Additionally DPA utilizes a market based system where the individual costing of a robot is respected, to deal with homogeneity within the team of robots. Each task is assigned a reward value based on the importance of its execution. This is embodied into a token along with information regarding owner, goal, required capabilities, and history of bidders as illustrated in Figure 6.

A particular token can be held by exactly one agent at any time instance after its generation. The agent makes a decision to either retain this particular token or pass it on. Figure 7 is a flowchart of the algorithm.

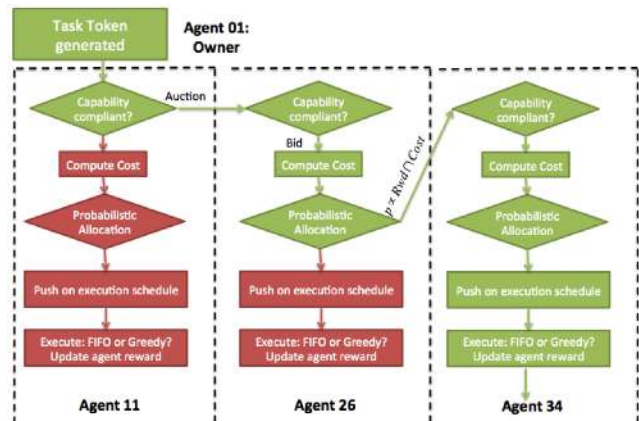


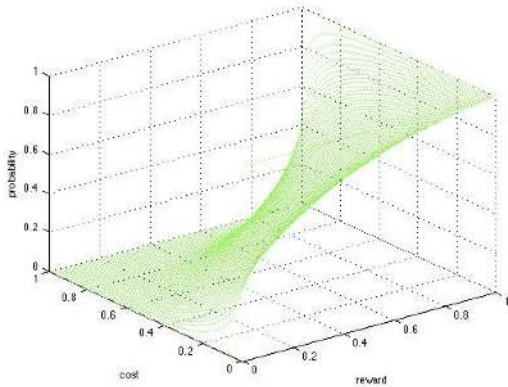
Figure 7. DPA Flowchart

The decision to retain a given task token is a probability function of the associated reward and the individual predicted

cost to execute.

$$p(\text{retain}) = e^{-\frac{\ln(R)}{\ln(c_{ij})}} \quad (12)$$

where  $\forall R, c_{ij} \in [0, 1]$  are normalized rewards and costs. Figure 8 is a plot of the 3 dimensional probability distribution function. Following such a probabilistic retainment structure increases task completion likelihood exponentially given a rise in reward (or reduction in cost).



**Figure 8.** Probability Distribution Function for retaining tasks

## 5. MODEL BUILDING

The planetary exploration domain presents a contradiction: the slow computation and low communication bandwidth of space systems limit the resolution of maps and models built online, but the highest quality maps and models are desired, because robots are there to observe what humans cannot. For the mission scenario discussed here, the solution is for robots to store data as they explore and to slowly transmit data back to operators after they have completed their exploration tasks. Better resolution maps and higher-dimensional models can then be generated in post-processing. For a mission, this might mean that coarse 3D maps are created online; post-processing builds higher resolution 3D models and 4 or 5D models that augment geometry with multi-spectral reflectance, soil cohesion, or gas concentration. In order to demonstrate this concept with readily available robot hardware, experiments presented here generate coarse 2D maps online, produce refined 2D maps in post-processing, and augment 2D maps with 3D data where available.

### 2-Dimensional

Two dimensional maps for the mapping robots are constructed using the GMapping SLAM algorithm [23], [24]<sup>2</sup>. This produces an occupancy grid with cells marked as either free, occupied, or unknown. To combine maps from multiple mapping robots, each map is converted to a 2D point cloud, with points for every occupied cell. The point clouds are transformed into roughly the same frame based on the recorded starting pose. The point clouds are trimmed to include only points in the overlapping region. The alignment is then refined using 2D iterative closest point (ICP) [25]. The transformation that results from ICP is then applied to the maps. Figure 14 shows example merged maps. To display

<sup>2</sup>The gmapping ROS package was used. For more information, see <http://www.ros.org/wiki/gmapping>.

2D maps augmented with 3D data, the maps were converted to mesh models and trimmed to include explored areas only.

### 3-Dimensional

The scans from a 3D modeling robot (described in more detail in Section 6) have been obtained in a stop-and-scan fashion from a rotating laser scanner to obtain 3D point clouds. Pre-processing of the point clouds is done using a median filter to remove Gaussian noise, and the point cloud is further down-sampled to reduce the computational requirements.

Multiple 3D scans are necessary to preclude occlusions when digitizing environments. This requires registering several 3D scans and merging them into one coordinate system. If a 3D modeling robot has precise localization, the registration can be done directly using the robot pose. However, robot motion on natural terrain has to cope with 3D rotations and translations. Pose estimation becomes a problem with six degrees of freedom. This combined with noisy sensors, makes the self-localization erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration.

Automated registration of the scans is thus done at two levels. An initial alignment is done based on the noisy odometry data followed by the fine alignment using scan matching. For scan matching, point clouds are trimmed to include only points that, based on coarse alignment, lie in overlapping regions. Simultaneous matching [26], a modified version of the 3D iterative closest point algorithm (ICP) has been used to find the transformation matrices between the two scans. kD-trees have been used to speed up the data access [27], which ensures that a data point can be selected in  $O(\log n)$ . See Figure 15 for an example of a generated 3D model.

## 6. EXPERIMENTS & RESULTS

Experiments were conducted at two outdoor test sites and included teams of two to four robots. High-resolution laser scans were collected with a tripod-mounted Faro Photon80 at each test site. These models serve as ground truth against which experiment data is compared. A 2D experiment evaluated exploration performance, and two 3D experiments investigated 2D/3D model building.

### Robots

The robot team in the experiments consisted of 2D mapping robots, a 3D modeling robot, and a science sampling robot. Not all robots were used in all tests. The 2D mapping robots have SICK scanning LIDARs that scan parallel to the ground plane (see Figure 9). One covers 180 degrees at 1 degree resolution, the other covers 270 degrees at 0.5 degree resolution. The 3D modeling robot has a SICK scanning LIDAR mounted on a rotating base, (See Figure 10). By spinning the LIDAR while scanning, it builds a 3D point cloud. In these experiments, this robot was operated in a stop-and-scan mode: the robot would stay still and collect data for approximately 15 seconds before moving to its next position. The science sampling robot has a cone penetrometer to measure soil properties, and no exteroceptive navigation sensors (See Figure 11). The sampling robot was given tasks to proceed to a sampling location by the task allocation system, and to operate the penetrometer.

### Test Sites

One test site was a patio area surrounded by buildings and retaining walls (See Figure 12). This created defined bound-



**Figure 9.** 2D Mapping Robots



**Figure 10.** 3D Modeling Robot



(a) Sampling robot at mock-cave test site



(b) Detail of cone penetrometer

**Figure 11.** Science Sampling Robot

aries to the 2D explorable area, as would be observed inside a cave. The buildings, trees, and one tunnel under a bridge also provide significant 3D structure to be modeled. The other test site was a mock cave. It had terrain more realistic for planetary exploration, with dirt, gravel, and rocks. Caves, and especially cave entrances, often have areas of very blocky and rough terrain that pose a significant challenge to robot mobility. Because robot mobility and cave access were not studied in these experiments, the terrain was designed to be mostly drivable, but still challenging enough that less capable robots (in this case, the 2D mappers) occasionally get stuck. The test site contained an outdoor “surface” area and an

indoor “cave” area.



**Figure 12.** Patio test site. Orange lines outline drivable area. Red star indicates approximate starting location of robots.



(a) Outside View



(b) Inside View

**Figure 13.** Mock cave test site. Tunnel extends approximately 300m. Cave floor is covered in rocky material to emulate planetary terrain. Site contains surface terrain and tunnel (inside building).

### *Exploration Performance Experiments*

A comparison of exploration performance between distributed, centralized, and uncoordinated task allocation was conducted using a 2-robot team of 2D mapping robots. For the uncoordinated runs, tasks were randomly assigned to a robot, and the robot randomly decided whether to keep the task, not taking into account any costs associated with that



robot's performance of the task. Maps at a resolution of 0.05 meters per pixel were built from 5 runs of each type. Each run lasted 15 minutes. The operator indicated tasks that should be performed, and the system assigned these tasks to a robot. In 3 out of 5 runs for each set, the first selected task was in the direction of the bridge, in the other 2 it was in the direction of the dead-end to the right of the starting position in Figure 12. The same operator selected tasks for all runs.

Tables 1, 2, and 3 show results for each run. Percent explored is the percentage of the explorable area (as determined from the ground truth model) that the robot team explored. Unique to total is the ratio of the area explored by a single robot to the total area explored by the team. This metric gives a sense of how much overlapping work the robots are doing. The average percent explored for runs with a first task in the direction of the bridge was 68%, and for runs with the first task in the direction of the dead-end, 67%. The average ratio of unique to total explored for these cases was 0.45 and 0.44, respectively. The average over all runs was 67% of explorable area covered and a ratio of 0.45 unique to total explored area. Figure 14 shows merged maps for the runs with the largest and smallest explored areas in this experiment.

**Table 1.** 2D Mapping Results, Distributed

Run	Bridge 1st?	% Explored	Unique:Total
1	1	80	0.57
2	1	52	0.43
3	1	94	0.45
4	0	97	0.76
5	0	60	0.37
Mean		77	0.52
Std Dev		20	0.15

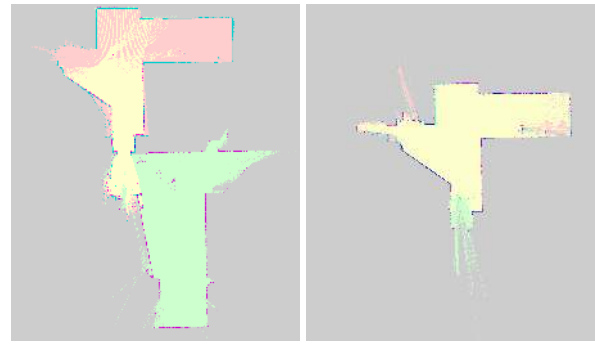
**Table 2.** 2D Mapping Results, Centralized

Run	Bridge 1st?	% Explored	Unique:Total
1	1	75	0.58
2	0	52	0.46
3	1	55	0.29
4	0	49	0.15
5	1	62	0.59
Mean		59	0.41
Std Dev		10	0.19

**Table 3.** 2D Mapping Results, Uncoordinated

Run	Bridge 1st?	% Explored	Unique:Total
1	1	51	0.20
2	0	54	0.26
3	1	73	0.49
4	0	87	0.65
5	1	68	0.42
Mean		67	0.41
Std Dev		15	0.18

These results show high performance variation within run sets and do not show significant difference between sets. The direction of the first assigned task did not significantly affect results. Limited navigation and path planning capabilities on individual robots, (common for all runs), likely introduces significant randomness. If the robots could more reliably complete their assigned tasks, differences between task allocation strategies would likely become more evident. Explo-



(a) Largest explored area (b) Smallest explored area

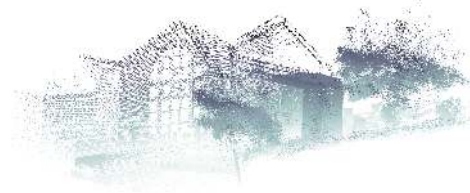
**Figure 14.** Maps built by a pair of 2D mapping robots. Yellow indicates area seen by both robots. Magenta indicates area seen by one robot, and Cyan represents area seen by the other.

ration of larger areas could also make differences clearer, as more tasks would need to be assigned.

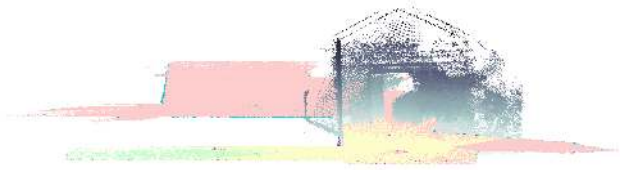
The lack of significant differences between allocation methods is somewhat encouraging, however. It indicates that distributed task allocation, the method believed to be most promising for planetary missions, does not perform any worse than other methods in early tests. The uncoordinated method, while by far the simplest, would fail once robots with different capabilities are introduced. Failure would occur, for example, if a 3D modeling task were assigned to a 2D mapping robot.

### Mapping and Modeling Experiments

An experiment including both 2D mapping and 3D modeling was conducted at the patio test site. In this experiment, the two 2D mapping robots were operated as described in section 6. A mapped area was then selected by the operator for 3D modeling, and the 3D modeling robot was sent to complete that task. There was no time limit on the run. Figure 15 shows a 3D point cloud built of the patio environment. Figure 16 shows a model built combining 2D map data with 3D model data.



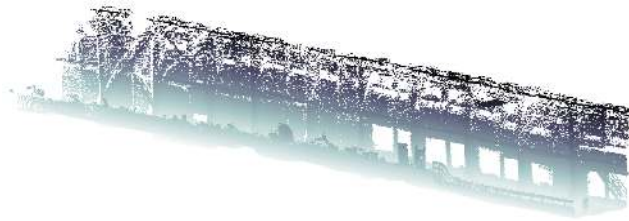
**Figure 15.** 3D model of the patio test site



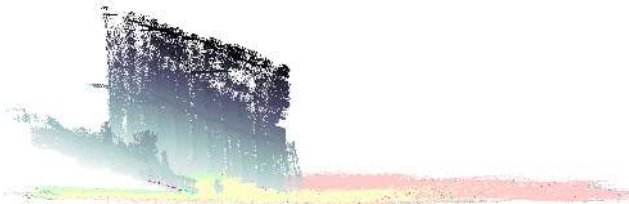
**Figure 16.** Model of the patio test site combining 2D map data with 3D model data.

A four-robot mission scenario experiment was conducted at the mock-cave test site. This included two 2D mapping

robots, a 3D modeling robot, and a science sampling robot. There was no time limit on the run. Figure 17 shows a 3D model of the tunnel in the mock cave. Figure 18 shows a model built combining 2D map data with 3D model data.



**Figure 17.** 3D model of the tunnel in the mock cave test site



**Figure 18.** Model of the mock cave test site combining 2D map data with 3D model data.

## 7. CONCLUSIONS & FUTURE WORK

The multi-robot coordination framework presented in this paper has been demonstrated to work for planetary cave mission scenarios where robots must explore, model and take science samples. Toward that end, two coordination strategies have been implemented, centralized and distributed. Further, a core communication framework has been outlined to enable a distributed heterogeneous team of robots to actively communicate with each other and the base station, and provide an online map of the explored region. An operator interface has been designed to give the scientist enhanced situational awareness, collating and merging information from all the different robots. Finally, techniques have been developed for post processing data to build 2 & 3-D models of the world that give a more accurate description of the explored space. Fifteen 2D mapping runs with 2 robots were conducted. The average coverage over all runs was 67% of total explorable area. Maps from multiple robots have been merged and combined with 3D models for two test sites.

Despite these encouraging results, several aspects have been identified that can be enhanced. Given the short mission durations and small team of robots in the experiments conducted, a simple path-to-goal costing metric was sufficient. To use this system for more complex exploration and sampling missions, there is a need for learning-based costing metrics. Additional costing parameters have already been identified and analyzed for future implementation over the course of this study. One of the allocation mechanisms in this study was a distributed system, however, task generation remained centralized through the operator interface. In an ideal system robots would have the capability to generate and auction tasks, based on interesting features they encounter. Lastly, the  $\mathcal{NP}$ -complete scheduling problem was approximated during task generation. However, better results could potentially be obtained by releasing this responsibility to the individual robots.

## ACKNOWLEDGMENTS

The authors thank the NASA STTR program for funding this project. They would also like to thank Paul Scerri and the rCommerceLab at Carnegie Mellon University, for lending hardware and robots for this research.

## REFERENCES

- [1] G. Cushing, T. Titus, and E. Maclennan, "Orbital observations of Martian cave-entrance candidates," in *First Intl. Planetary Caves Workshop*, Carlsbad, NM, 2011.
- [2] J. W. Ashley, M. S. Robinson, B. R. Hawke, A. K. Boyd, R. V. Wagner, E. J. Speyerer, H. Hiesinger, and C. H. van der Bogert, "Lunar caves in mare deposits imaged by the LROC narrow angle camera," in *First Intl. Planetary Caves Workshop*, Carlsbad, NM, 2011.
- [3] J. W. Ashley, A. K. Boyd, H. Hiesinger, M. S. Robinson, T. Tran, C. H. van der Bogert, and LROC Science Team, "Lunar pits: Sublunarean voids and the nature of mare emplacement," in *LPSC*, The Woodlands, TX, 2011.
- [4] J. C. Werker, S. M. Welch, S. L. Thompson, B. Sprungman, V. Hildreth-Werker, and R. D. Frederick, "Extraterrestrial caves: Science, habitat, and resources (a niac phase i study)," NASA Innovative Advanced Concepts (NIAC), Tech. Rep., 2003.
- [5] S. Dubowsky, K. Iagnemma, and P. J. Boston, "Microbots for large-scale planetary surface and subsurface exploration, niac phase i." NASA Innovative Advanced Concepts (NIAC), Tech. Rep., 2006.
- [6] S. Dubowsky, J. Plante, and P. Boston, "Low cost micro exploration robots for search and rescue in rough terrain," in *IEEE International Workshop on Safety, Security and Rescue Robotics*. Gaithersburg, MD, 2006.
- [7] S. B. Kesner, "Mobility feasibility study of fuel cell powered hopping robots for space exploration." Master's thesis, Massachusetts Institute of Technology, 2007.
- [8] M. Tambe, D. Pynadath, and N. Chauvat, "Building dynamic agent organizations in cyberspace," *IEEE Internet Computing*, vol. 4, no. 2, pp. 65–73, March 2000.
- [9] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robot. Auton. Syst.*, vol. 54, no. 12, pp. 945–955, Dec. 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.robot.2006.06.003>
- [10] B. Browning, J. Bruce, M. Bowling, and M. M. Veloso, "Stp: Skills, tactics and plays for multi-robot control in adversarial environments," *IEEE Journal of Control and Systems Engineering*, 2004.
- [11] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, September 2004.
- [12] M. Koes, I. Nourbakhsh, and K. Sycara, "Heterogeneous multirobot coordination with spatial and temporal constraints," in *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*. AAAI Press, June 2005, pp. 1292–1297.
- [13] M. Koes, K. Sycara, and I. Nourbakhsh, "A constraint optimization framework for fractured robot teams," in *AAMAS '06: Proceedings of the fifth international joint*

conference on Autonomous agents and multiagent systems. New York, NY, USA: ACM, 2006, pp. 491–493.

- [14] M. B. Dias, B. Ghanem, and A. Stentz, “Improving cost estimation in market-based coordination of a distributed sensing task.” in *IROS*. IEEE, 2005, pp. 3972–3977.
- [15] M. B. Dias, B. Browning, M. M. Veloso, and A. Stentz, “Dynamic heterogeneous robot teams engaged in adversarial tasks,” Tech. Rep. CMU-RI-TR-05-14, 2005, technical report CMU-RI-05-14.
- [16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005, ch. 9, pp. 222–236.
- [17] H. Moravec and A. E. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, March 1985.
- [18] M. Yguel, O. Aycard, and C. Laugier, “Update policy of dense maps: Efficient algorithms and sparse representation,” in *Intl Conf on Field and Service Robotics*, 2007.
- [19] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “An efficient k-means clustering algorithm: analysis and implementation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002.
- [20] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, “An analysis of several heuristics for the traveling salesman problem,” *SIAM Journal on Computing*, Sept 1977.
- [21] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe, “Token approach for role allocation in extreme teams: analysis and experimental evaluation,” in *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2004.
- [22] M. B. Dias, D. Goldberg, and A. T. Stentz, “Market-based multirobot coordination for complex space applications,” in *The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, May 2003.
- [23] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- [24] —, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, 2006.
- [25] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, June 2012.
- [26] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, “6d slam with an application to autonomous mine mapping,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 1998–2003.
- [27] D. Simon, M. Hebert, and T. Kanade, “Real-time 3-d pose estimation using a high-speed range sensor,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994, pp. 2235–2241.

## BIOGRAPHY



**Ammar Husain** received his B.S. in Mechanical Engineering (Robotics) from the University of Illinois at Urbana-Champaign. He is pursuing an M.S. in Robotic Systems Development at Carnegie Mellon University. He has previously worked on the guidance and control of autonomous aerial vehicles. His research interests lie in the field of perception-based planning.



**Heather Jones** received her B.S. in Engineering and B.A. in Computer Science from Swarthmore College in 2006. She analyzed operations for the Canadian robotic arm on the International Space Station while working at the NASA Johnson Space Center. She is pursuing a PhD in Robotics at Carnegie Mellon University, where she researches reconnaissance, exploration and modeling of planetary caves.



**Balajee Kannan** received a B.E. in Computer Science from the University of Madras and a B.E. in Computer Engineering from the Sathyabama Institute of Science and technology. He earned his PhD from the University of Tennessee-Knoxville. He served as a Project Scientist at Carnegie Mellon University, and is currently working at GE as a Senior Cyber Physical Systems Architect.



**Uland Wong** received a B.S. and M.S. in Electrical and Computer Engineering and an M.S. and PhD in Robotics, all from Carnegie Mellon University. He currently works at Carnegie Mellon as a Project Scientist. His research lies at the intersection of physics-based vision and field robotics.



**Tiago Pimentel** Tiago Pimentel is pursuing a B.E. in Mechatronics at Universidade de Brasilia, Brazil. As a summer scholar at Carnegie Mellon University's Robotics Institute, he researched on multi-robots exploration. His research interests lie in decision making and mobile robots.



**Sarah Tang** is currently a senior pursuing a B.S. degree in Mechanical and Aerospace Engineering at Princeton University. As a summer scholar at Carnegie Mellon University's Robotics Institute, she researched multi-robot coordination. Her research interests are in control and coordination for robot teams.



**Shreyansh Daftry** is pursuing a B.E. in Electronics and Communication from Manipal Institute of Technology, India. As a summer scholar at Robotics Institute, Carnegie Mellon University, he researched on sensor fusion and 3D modeling of sub-surface planetary caves. His research interests lie at the intersection of Field Robotics and Computer Vision.



**Steven Huber** received a B.S. in Mechanical Engineering and an M.S. in Robotics from Carnegie Mellon University. He is currently Director of Structures and Mechanisms and Director of Business Development at Astrobotic Technology, where he leads several NASA contracts.



**William "Red" L. Whittaker** received his B.S. from Princeton University, and his M.S. and PhD from Carnegie Mellon University. He is a University Professor and Director of the Field Robotics Center at Carnegie Mellon. Red is a member of the National Academy of Engineering, and a Fellow of the American Association for Artificial Intelligence.