Noname manuscript No.
(will be inserted by the editor)

Distributed Inference-Based Multi-Robot Exploration

Andrew J. Smith · **Geoffrey A. Hollinger**

Received: date / Accepted: date

Abstract This work proposes a technique for distributed multi-robot exploration that leverages a novel method of map inference to increase the team's cumulative exploration efficiency. The proposed distributed inference technique uses observed map structure to infer unobserved map features. The multi-robot team then uses a decentralized algorithm to coordinate the exploration using both the inferred and observed portions of the map. Individual robots select exploration poses by accounting for expected information gain and travel costs. Robots resolve conflicts between exploration goals with local auctions of expected travel costs. The benefits of inference-informed exploration are demonstrated in both simulated explorations and hardware trials. The proposed technique is compared against frontier-based and information -based exploration approaches. These comparisons evaluate the performance of the three exploration methods with decaying communication and a varied number of agents. Additionally, the accuracy of the map inference technique is evaluated using publicly available sensor datasets. The proposed inference technique improves the correctly-estimated subset of the environment by an average of 34.47% (maximum 108.28%) with a mean accuracy of 95.1%. This leads to a 13.15% reduction in the cumulative exploration path length in the trials conducted. The developed system was then verified on three Pioneer P3-DX robots with laser scanners in experimental trials.

A. Smith
Robotics Group
School of Mechanical Industrial and Manufacturing Engineering
Oregon State University
Corvallis Oregon 97330
E-mail: smithan7@oregonstate.edu

G. Hollinger
Robotics Group
School of Mechanical Industrial and Manufacturing Engineering
Oregon State University
Corvallis Oregon 97330
E-mail: geoff.hollinger@oregonstate.edu

1 INTRODUCTION

The applications of distributed robotic systems continues to grow, and consequently there is an increasing demand for algorithms that efficiently coordinate teams of robots to explore and map environments. This paper presents a method of coordinated exploration for decentralized multi-robot teams that leverages novel distributed map-inference techniques. This coordination method is applicable to multi-robot teams exploring environments for civilian, search and rescue, military, and research applications. Results are presented to demonstrate the value and accuracy of the inference as well as the benefit inference provides to multi-robot exploration. Experiments were conducted with teams of robots exploring an indoor space, but the techniques are generalizable to other structured environments such as tunnels, caves, and mines.

The primary objective of multi-robot exploration can be stated in two ways. First, fully explore a fixed space in the minimum time or travel distance. Or alternatively, explore the maximum space within allotted time and energy constraints. Meeting this objective is done by solving the problem of determining the next location each robot should observe from. In a distributed system the coordination problem is complicated by each agent attempting to identify their own exploration goal with limited information sharing and planning between agents. In a decentralized system, this process is further complicated by the lack of a hierarchical structure with which to assign each member of the team exploration tasks. Despite the complications of distributed and decentralized systems, they are still frequently used because of their benefits. As each robot requires minimal input from other members of the team, they are robust to failures of individual team members and to unreliable and limited communications.

While distributed systems are robust to single point failures, they still require some type of coordination to properly allocate system resources. Without a method to resolve conflicting robot goals, if each agent acts greedily they will likely converge onto a single goal, significantly reducing system efficiency [1]. Market based approaches have been used to settle disputes to increase system performance in exploration tasks [2], [3]. One complication in exploration based tasks is determining the value of exploring different areas of the environment. Some recent work has attempted to infer the structure of the unobserved portions of the map to aid in the exploration using visual features [4] [5]. Our work develops a new distributed map inference technique that uses simulated laser scans and not visual features to infer map structure. Then the inferences are used to inform goal selection for a market-based distributed exploration. The technique is demonstrated in simulation and hardware experiments.

The contributions of this paper are:

1. A distributed inference technique that uses environment geometry and a library of previously collected map structures to infer over unexplored regions of the environment and merge the inferences into the agent's exploration map.
2. A decentralized coordinated exploration algorithm to exploit the information provided by the distributed map inference.

The developed distributed map-inference technique uses intuitive methods to infer unexplored map structure based upon the partially observed map, as shown in Fig. 1. Then, using the information gained by the inference, each member selects a goal pose to observe from and broadcasts their selected goal pose to other team members. The inference process consists of two steps. First, we estimate the boundaries of the explorable space using a heuristic method. Second, a library of map structures uses the inferred boundary and observed map structure to infer unobserved map portions of the map. The inference method is tested in simulation, on real-world datasets, and in hardware trials to demonstrate both the accuracy of the inference and the benefits inference provides during exploration. The developed inference method provides a mean increase of 34.47% (maximum of 108.28%) of map recall and maintains a mean precision of 95.1%.

The decentralized exploration algorithm leverages the map-inference by searching over the both the observed and inferred portions of the map for poses that fully observe the inferred free space (traversable space that is currently unexplored). Then, each robot selects a pose using an information theoretic approach from the set of sampled poses as its exploratory goal; the robot broadcasts the selected pose

Fig. 1 The developed inference technique uses the observed portions of the map (white) to infer the structure of the unobserved portions (red), then uses the inferred structure to guide the exploration.

to local robots. Disputes are settled using a distributed single bid local auction of travel cost. This approach is compared against frontier-based exploration [1] and (inference-free) information pose coordination strategies. Our strategy reduces the average time required to explore an environment by 13.15% and 12.34% for varied communication strength and numbers of agents, respectively.

2 BACKGROUND AND RELATED WORK

The exploration and planning research community has made great strides in increasing the efficiency of robotic teams exploring unknown environments. Recently there has been growing interest in using the observed portions of the map to infer unobserved map structure with the purpose of improving exploration efficiency and mapping accuracy [4] [5]. Simultaneously, there have been advances in exploration techniques to increase exploration efficiency for individual and teams of robots. This section will discuss related work in autonomous exploration, distributed coordination, and map inference.

2.1 Autonomous Exploration

Robotic exploration of an unknown environment is primarily the task of determining where a robot should observe from next. Common approaches guide robots towards frontiers; i.e. the areas that separate the explored portion of a map from the unexplored. A greedy frontier approach, for example, iteratively assigns each robot their closest frontier to explore. They then explore the receding frontier until the area is completely explored or another frontier becomes closer. This process is then repeated until the exploration is complete [1]. These greedy approaches, while effective, are often inefficient as the nearest frontier is not necessarily the frontier with the highest exploration value.

One approach that has been used to increase the efficiency of exploration is a market or value-based approach to selecting exploration goals [6]. This approach uses some metric to determine the value of an exploration pose by accounting for the information gained and travel cost of observing from the selected pose. To select the pose, robots evaluate the set of poses in their market and then select the pose that will provide them with the largest value [6]. Robots then continue the exploration by continuously selecting the pose that will result in the largest cumulative value. Unfortunately, while the travel cost can be easily estimated it has been an ongoing problem to estimate the reward of exploring each pose.

There have been efforts to use potential information gain to predict the reward of exploration poses or series of linked poses forming a path. For example, information-theory based approaches were successfully used to calculate potential information gain from different robot poses and plan optimal multi-step actions [7] [8] [9]. These methods estimate the information gained over a series of actions over the next few time steps and then select the actions that will lead to the largest information gain. Another method uses estimated information gain to determine a minimal set of poses that observe the frontiers and plans the exploration using this set of poses [10]. These approaches simulate sensor readings from a prospective pose to approximate the information gained by observing from that pose.

While such approaches have been successful, there are two ways the are limited by the lack of knowledge on the areas they have not yet been observed. First, by not inferring where the explorable space ends, information-based approaches can over-estimate the value of exploration poses. Second, not inferring the boundary of the explorable space limits the locations where simulated poses can be sampled. Incorporating map inference resolves both of these problems. By inferring the outer limit of the environment we can simultaneously increase the accuracy of estimated pose rewards and allow for poses to be sampled (estimate pose reward) in unobserved, but inferred to be explorable, space.

2.2 Coordinating Multi-Robot Teams

The problem of automated exploration becomes further complicated when it is expanded to include teams of multiple robots that must be coordinated. A common approach is a centralized system that views the team as a single robotic system governed by a central controller with an action space consisting of the cumulative actions of all the robotic hardware it controls. There are multiple issues with this approach. For example, a centralized system requires every robot it controls to maintain communication with the central controller. If a robot travels to a remote area, which is often the goal in an exploration, then it may be incapable of planning a path to continue the exploration or even regain communication with the central controller. The next issue is that, as the number of robots grows, the action space increases exponentially. Consequently, optimal coordination is frequently computationally intractable for teams of more than a few robots [11], significantly diminishing the benefits of a centralized controller. Finally, such a system is highly vulnerable to failure of the central controller. If the central controller fails then it is likely the entire team will be unable to continue the exploration [6]. A common solution to some of these issues is to use a distributed system to coordinate teams of multiple robots.

A distributed system is one in which each robotic member acts largely independently from the remainder of the team. There is no hierarchy of command; no team member supervises the actions of any other team member. Distributed robots primarily rely on locally available information and interact with agents in their immediate vicinity, limiting the computational and communication requirements for planning [12]. The structure of a distributed system makes it robust against failure, loss of communication, and unexpected changes in the environments. Thus distributed systems are especially well suited for exploration tasks, as each agent performs solitary observation tasks based off of local information.

The information based approach to exploration provides a straightforward method to account for other agents' exploration action in a distributed system. The coordination method implemented in this work uses a single-bid local auction to settle disputes between agents [13]. Robots bid over potential exploration goals using their expected travel cost. As outlined above, each robot uses an internal market to select their exploration goal poses. Each robot selects an exploration pose and then broadcasts both their exploration pose and its travel cost to robots in the local vicinity. As the exploration continues, the auction process proceeds and newly found exploration poses are locally auctioned and awarded to the wining robot. Robots keep a record of recent bids of local robots allowing coordination to continue when robots break communication. Auction based coordination requires only limited local communication (conflicts are more likely locally and limited information is exchanged for the auction) and computation (robots individually compute their bids) [14].

2.3 Map Inference

While there has been some prior work in map inference for exploratory robotics, this is a largely unexplored area of research. Previous work in map inference has focused on using visual features to identify similar map segments in the area being explored and a library of map segments [4] [5]. Then the visual features are used to align and merge the library entry into the map of the area being explored. These two approaches will be discussed individually below.

Early work in map inference, named P-SLAM (Predictive-SLAM), used a Bayesian model to predict hidden structures of partially observed maps by comparing the partially observed map areas against fully observed structures [4]. This was done to speed up the process of simultaneous localization and mapping (SLAM) by providing additional information into a traditional SLAM algorithm, allowing it to converge faster. Also, if a partially observed portion of the map had a high similarity with a previously explored area, then it could be identified and skipped to further speed up the exploration. This portion of their algorithm, named look ahead mapping, uses visual features to identify corners and walls and then uses spatial alignment to identify matches and place them on the map. While P-SLAM provides accurate estimates of the map structure, it is limited to areas that are partially observed. Additionally, it can only identify map segments that are similar to segments that been explored in the current exploration and relies on visual features to identify matches.

An alternative method of map inference was developed to identify potential loop closures in the unobserved portions of the map [5]. This method uses a visual bag of words technique used in appearance based place recognition, specifically FABMAP 2 [15], to expand the potential matches that could by added to the map. Potential matches are selected from a library of map structure by comparing visual feature descriptors and then the best matches are aligned using RANSAC [16]. Then the matching map structure is merged into the exploring robot's map. The addition of the library of map structure allowed for a larger set of map structures to be searched over for potential matches.

Our proposed work similarly uses a library of map structure to infer the unobserved portions of the map. Our method differs in two fundamental ways. First, visual features are not used (such as SURF [17] or SIFT [18] features) to identify map structure because visual features in maps are often highly similar and lead to ambiguity in matching. This requires the use of RANSAC [16], ICP [19], or other computationally expensive techniques for match alignment. To compensate, our inference method uses simulated sensor readings (in the form of sparse 360^o laser scans) to identify potential matches. Second, our method uses a two step inference approach. The first step is to infer the outer limits of the explorable space using the observed outer map boundaries to infer the unobserved boundaries of the map. Then the observed and inferred boundaries are used to infer the now enclosed unobserved space using a library of map structure described by sensor readings.

3 METHODS

This work uses map inference to guide a distributed team of robotic agents to efficiently explore a space. Exploration is accomplished by having individual robots visit a series of poses and observe the environment. Our method uses a map-inference informed information-theoretic approach to identify a sequence of poses for each robot to explore. In this approach, robots:

1. Calculate pose reward using the observations and inferences of the environment to predict the expected information gain of observing from sampled poses.
2. Select individual exploration goal poses using an internal market to maximize their individual collected reward.
3. Settle conflicting exploration poses with a single bid auction between local robots.

This process is repeated as the exploration proceeds, allowing for continuous refinement of the map inference, exploration goals, and goal distribution.

3.1 Inference

The inference developed in this work is comprised of two components: a heuristic-based perimeter inference used to estimate the outer boundary of the area being explored, and structural inference used to infer internal map structure. We use the word 'inference' in this paper to refer to prediction based on probabilistic matching from a database. We note that our technique does not perform formal inference over a graphical model. The two inference components can be combined or applied independently depending upon the environment and user's desires. In an environment that is sparsely filled with occupied space (such as buildings where the space occupied by walls is significantly less than the traversable space bounded by the walls), perimeter inference is intended to estimate the outer boundary of the explorable space. Then the structural inference can be used to identify both structure inside the inferred perimeter and potential breaches of the inferred perimeter. However, if the area is mostly filled with non-traversable space (e.g. a cave or mine environment where there are few tunnels of traversable space among the mostly non-traversable space of the walls) then the structural inference can be used to estimate potential connections between partially explored tunnels and mine shafts. The inference process is performed by each robot during each planning step on the robot's complete observed costmap regardless of the current robot location.

Before proceeding to the description of the two inference algorithms we provide definitions for terms introduced in this work. The explorable space of the environment, denoted by X, is the set of poses, p, a robot can observe by traveling from its start location in a continuous path. The explorable space is discretized into cells, $c_{i,j}$ to form a 2D costmap representation of X. While the discretization of the environment into a 2D costmap results in a loss of information, by taking care when selecting the discretization step size or using structures such as an Oct-Tree [20] the loss of information can be reduced. During the exploration the robot can observe both free, $X_f \subset X$, and occupied, $X_o \subset X$, space. A costmap cell is considered free, $c_{i,j} \subset X_f$ if for all $p \in c_{i,j}; p \notin X_o$. Alternatively, costmap cells that contain any poses in collision with an obstacle are occupied. A cell is considered observed, $c_{i,j} \in O$, if the probability of occupancy surpasses a specified threshold to be either free, $O_f \subset X_f \cap O$, or occupied, $O_o \subset X_o \cap O$. A cell that is not observed can be either unknown, U, or inferred, I. Inferred cells can be either free, $I_f \subset I$, or inferred occupied, $I_o \subset I$. A cell is inferred to be either free or occupied when it is unknown but the nearby environment provides insight into what the occupancy of the cell is. It is possible for an unknown cell to become inferred and for an inferred cell to become unknown. It is not possible for an observed cell to become either unknown or inferred.

3.1.1 Perimeter Inference

The primary objective of the perimeter inference is to estimate the outermost boundary of the explorable space. The process, as outlined in Algorithm 1, is a heuristic method that begins by identifying the convex hull of the observed map. Observed wall cells in the costmap on the convex hull are used to infer likely unobserved portions of the building perimeter. These unobserved portions of the perimeter are found using a probabilistic Hough line transform to identify unobserved connections between observed wall segments; the probabilistic Hough line transform uses a maximum likelihood estimation of a line through sparsely connected points [21]. Identified wall-lines are then extended into the unobserved portions of the map. Where each of these wall-line projections intersects, an inferred corner (as a single cell, $c_{i,j} \in I_o$) is added to the exploration map at the point of the intersection (Figure 2a). In practice, it is beneficial to restrict the distance of inferred corners from the nearest observations to prevent nearly parallel lines from inferring corner locations impractically far from the observed space. Finally, a new convex hull is found, this time of the observed portions of the map and the identified corner points (Figure 2b). This new convex hull is an initial estimate of the bounds of the exploration space and the cells along the hull are inferred as occupied in the costmap while the cells inside the hull are initially inferred to be free.

Since the inferred perimeter is convex, it likely overestimates the boundaries of the explorable space. However, this optimistic approach increases the reward in uncertain areas providing increased incentive for their exploration. While the technique of extending lines tends to infer that most buildings are rectangular in shape, this is consistent with many man-made spaces. In non-rectangular environments, the inference is able to adapt to the shape of the environment because, while external boundaries are assumed to be straight, the inferred boundaries are not restricted to any orientation or class of polygon.

Algorithm 1 : Perimeter Inference

H = **convex hull**($O_f \cup O_o$) ◁ convex hull of observed walls and space
$H_o = O_O \in H$ ◁ identify occupied cells on convex hull contour
L_o = **Probabilistic Hough Lines**(H_o) ◁ search for lines on convex hull
for l in L_o **do**
  $c_{i,j} \in I_o$ = **find intersections**(l) ◁ identify intersecting hull lines
H = **convex hull**($O_f \cup O_o \cup I_o$) ◁ convex hull of observations and intersection points
return H

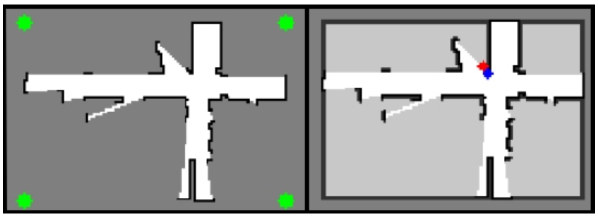

Fig. 2 (A) Partially observed space and the inferred intersection points shown in green. (B) The resulting inferred building hull and observed space with the exploring agent shown in blue and their current exploration goal shown in red.

Before proceeding to the structural inference and coordination, some maintenance is performed on the inferred map. Occupied cells, $O_o \cup I_o$, are inflated into I_f to account for unseen depth in obstacles; e.g. when an obstacle is only observed from one side it has no observable depth. Next, areas of the map that are inside of the inferred perimeter that are unreachable are marked as inferred obstacles.

3.1.2 Structural Inference

The structural inference uses the observed portions of the map and, if available, the inferred perimeter to infer unobserved internal map structure and potential breaches in the inferred perimeter. Structural inference is based on the assumption that most environment structure, even structure currently unexplored, is similar to other structure that exists. Following this logic, with the use of libraries of general map structure, it is possible to infer the unobserved portions of the environment by matching library structure with the observed portions of the environment.

In practice, the inference is performed by simulating a sparse 360^o laser range scan, s, at a randomly sampled pose in the observed environment, i.e. $p \in O_f$; demonstrated from the green pose in Figure 3 and in Algorithm 2 on line 1. s is simulated using ray-casting to simulate each individual beam, s_i, of the scan. If all of the cells passed through by any $s_i \in s$ are observed, then s is used to create a new library entry in the structural library of priors, L; as shown in Algorithm 2 lines 3-5 and the red inset in Figure 3. If any of the cells, c_j, passed through by any $s_i \in s$ are unobserved, $c_j \in I \cup U$, then the library is searched for the entry, $l^* \in L$, that maximizes the probability of having generated s.

Algorithm 2 : Structural Inference

1: p = **Sample Pose**(costmap) ◁ find sample pose
2: S = **Simulate Scan**(p)
3: **if** (!p_0.**need Inference**() **then**
4: $\quad L_c$.**add To Library**(S)
5: $\quad$ **return**
6: $l^* = \text{argmin}_{l \in L_c}$ **Compare Scans**(S, l) ◁ sample library for matching scan
7: **mergeScanIntoMap**(l^*, p) ◁ merge l^* into costmap at p
8: **return**

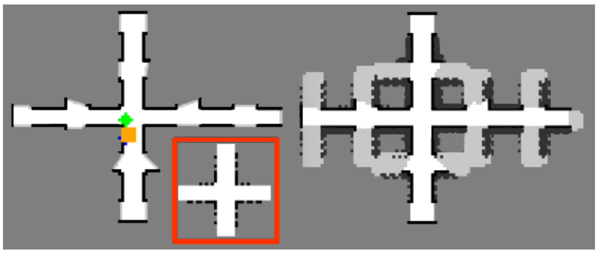

Fig. 3 The developed inference method uses the observed portions of the map (left) to create a library of map structure. The starting location (shaded square) and exploring robot (light-shaded circle) are shown. An example library entry is provided in the red inset from the exploring robot's pose. This library entry is then used to infer the unexplored portions of the map and is then merged with the observed map as shown in light gray for inferred free space and dark gray for inferred walls (right).

The probability of a match is calculated by considering each beam of the two scans; the library scan, l, and the simulated scan from the observed environment, s. Each scan is composed of a set beams with lengths, $z_l^i \in z_l$ and $z_s^i \in z_s$ and bearings. However, as both beams are simulated, the bearings have perfect correspondence and can be ignored. The mean and standard deviation of each set of scan lengths can be calculated giving μ_s, σ_s and μ_l, σ_l. The means are then used to normalize the library entry lengths, giving $\tilde{z}_l^i$, so that the shape of the library entry is prioritized over the size for the comparison. Then, assuming that the scan lengths are normally distributed, μ_s and σ_s define a normal distribution of z_s. The corresponding normal cumulative distribution function, $\Phi_s(z)$, can be approximated using the sigmoid [22]

$$\Phi_s(z) \approx \frac{1}{1 + e^{-\sqrt{\pi}(-0.0004406x^5 + 0.0418198x^3 + 0.9x)}}, \tag{1}$$

where

$$x = \frac{z - \mu_s}{\sigma_s}. \tag{2}$$

Then, the probability of a random beam in s being closer in length to z_i^s then $\tilde{z}_i^l$ is

$$P(s_i = l_i | \tilde{z}_l^i, z_s^i) = 1 - |\Phi_s(\lambda_1) - \Phi_s(\lambda_2)| \tag{3}$$

where

$$\lambda_1 = z_s^i + |\tilde{z}_l^i - z_s^i| \tag{4}$$

and

$$\lambda_2 = z_s^i - |\tilde{z}_l^i - z_s^i|. \tag{5}$$

Here, $|\Phi_s(l_1) - \Phi_s(l_2)|$ is the probability of another sample from the observed population being closer to the observed measurement then the library measurement. Resulting logic is that as $|\tilde{z}_l^i - z_s^i| \rightarrow 0$ that $P(s_i = l_i | \tilde{z}_l^i, z_s^i) \rightarrow 1$. From this, the probability of the library entry l generating the simulated observed scan s given all beams is

$$P(s = l | z_s, \tilde{z}_l) = \prod_{i=1}^{|z_s|} P(s_i = l_i | \tilde{z}_l^i, z_s^i). \tag{6}$$

Then, the library entry with the highest probability of generating the simulated observed scan is

$$l^* = \underset{l \in L}{\text{argmax}} \prod_{i=1}^{|z_s|} P(s_i = l_i | \tilde{z}_l^i, z_s^i). \tag{7}$$

Or as the product of probabilities is monotonically decreasing

$$l^* = \underset{l \in L}{\text{argmin}} \sum_{i=1}^{|z_s|} -log(P(s_i = l_i | \tilde{z}_l^i, z_s^i)) \tag{8}$$

is equivalent and simpler to compute.

To increase the library of structural priors beyond the observations of the current exploration, simulated scans from different environments are seeded into the structural inference library. For this work the structural library consisted of 86,246 entries from 124 environments [23]. While this provides a wide range of structural priors it also increases the computation time required to identify matching entries. To alleviate this concern, rather than searching through the complete library, the library is randomly sampled to identify the most probable corresponding library entry. To determine the number of samples to perform for each iteration of structural inference, an analysis of the match quality and computation time was performed. This analysis found that as the number of samples increases from 0 to 1000 the quality of the match significantly increases as shown in Figure 4. Increasing from 1,000 to 5,000 samples there is a small increase in match quality and there is little improvement beyond 5,000 sample comparisons. The time complexity scales linearly with the number of comparisons and each comparison takes $6.72 * 10^{-6} s$ on a Intel Xeon CPU with 8 cores at 3.7GHz and 32 GB or RAM.

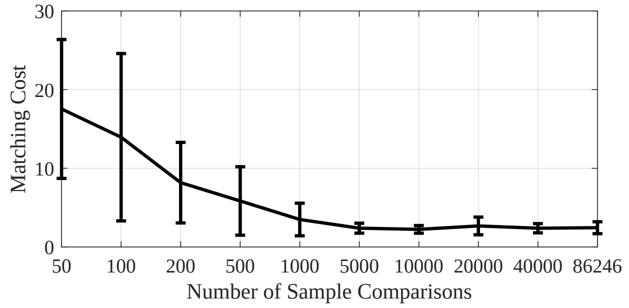

Fig. 4 The effect of the number of library comparisons performed in the structural inference on the minimum log-probability. The number of sample comparisons indicates how many library entries are sampled and compared against each query. A lower log-probability indicates a better quality match between the library and query scan. Each quantity of sample comparisons was tested 4,000 times and the mean and standard deviation are reported.

Matching library entries, l^*, are merged into the observed portions of the map if the log-probability of a match surpasses a user specified threshold. Bayesian inference is used to merge inferred cells into the costmap by assuming that the library of scans is a normally distributed collection that represents an exhaustive set of general map structure and the probability of a random entry in L having a higher probability of generating s than l^*, $P(l^*|L,s)$. Then, the occupancy of each affected cell, c_i, can be updated by

$$P(c_i \in I_f | l^*, L, s) = \frac{P(l^*|L,s) P(c_i \in I_f)}{P(l^*)} \tag{9}$$

if the corresponding cell in l^* if free. If the corresponding cell in l^* is occupied then $P(l^*|L,s)$ is replaced by $(1 - P(l^*|L,s))$. $P(c_i \in I_f)$ is the prior probability that cell c_i was inferred to be free before this iteration of inference. Initially all cells are given a prior probability of 0.5 representing complete uncertainty in the occupancy of c_i. When a cell is observed to be $c_i \in O_f \cup O_o$ its inferred occupancy is replaced with the observed occupancy of the cell. Using multiple iterations of inference over the same cell conflicting inferences can be resolved. When $P(c_i \in I_f) > 1 - \alpha$, where α is a user specified value, then c_i in the costmap is inferred to be free. Similarly, when $P(c_i \in I_f) \geq \alpha$, c_i is inferred to be occupied. To increase the usability of the resulting inferences, which as a reminder consists of simulated scans forming narrow and sparsely sampled inferences, the probability map is blurred using a Gaussian filter [24] to form a cohesive estimate of the inferred space before it is merged into agents occupancy map. An example output of the visual inference is provided in Figures 5.

Fig. 5 Example of a partially explored map, left, and sample visual inference, right. The starting location (shaded square) and exploring robot (light-shaded circle) are shown. The inferred free space in the right image is weighted by the number of predictions that infer the space to be free. The brighter the space the more likely it is believed to be free.

The inference process provides a two-step approach to infer the unexplored regions of the environment to assist in exploration. By inferring the unobserved free space, the exploring robots have additional information to plan how to explore the remaining space. Exploring robots use the inference in both their individual planner and in the multiagent coordination, Figure 6. In their individual planners, robots will search over both the inferred and observed free space for potential exploration poses. Then the robots will use the inferred free space to calculate the reward of exploring those poses. The inference informs the coordination by identifying conflicting robot goals.

3.2 Exploration Goal Selection

Each agent selects their exploration goal pose using an internal market-based approach. The market-based approach was chosen because of its high performance and ease of implementation in distributed exploration tasks [6]. In each robot's individual market they have a set of poses, P, with perceived rewards, $\hat{r}_p \in \hat{R}$, and costs, $\hat{c}_p \in \hat{C}$. The rewards are the predicted information gain from each pose, and costs are a weighted travel cost to reach each pose. The value of each pose, $\hat{v}_p$, is the reward minus the cost of each pose, $\hat{v}_p = \hat{r}_p - \hat{c}_p$. The robot explores by selecting

$$p^* = \operatorname*{argmax}_{p \in P} \hat{v}_p \tag{10}$$

and then executing the exploration of p^*.

least reward when the occupancy of the cell has a high probability of being occupied ($occ_i = 1.0$) or free ($occ_i = 0.0$). Alternatively, a reward mapping that prioritizes exploration is

$$\mathbf{reward}(c_i) = \begin{cases} 1 : c_i \in (I_f \cup U) \\ r^b : c_i \in I_b \\ 0 : c_i \in (I_o \cup O_f \cup O_o). \end{cases} \tag{13}$$

This approach prioritizes exploration by providing reward for observing cells that are inferred to be free or breach the inferred perimeter. An inferred breach, I_b, occurs when the structural inference contradicts with and extends beyond the inferred perimeter; i.e. a doorway to a new wing of a building. However, it is worth noting that multiple reward mappings can be added simultaneously to achieve different behaviors and accommodate multiple weighted objectives. An example of a combined occupancy and exploration reward mapping is shown in Figure 7. Our adaptation of using a generic reward map allows for flexibility in practice. Without modifying the coordination or exploration algorithm we can prioritize different agent behaviors by adding or modifying reward mappings.

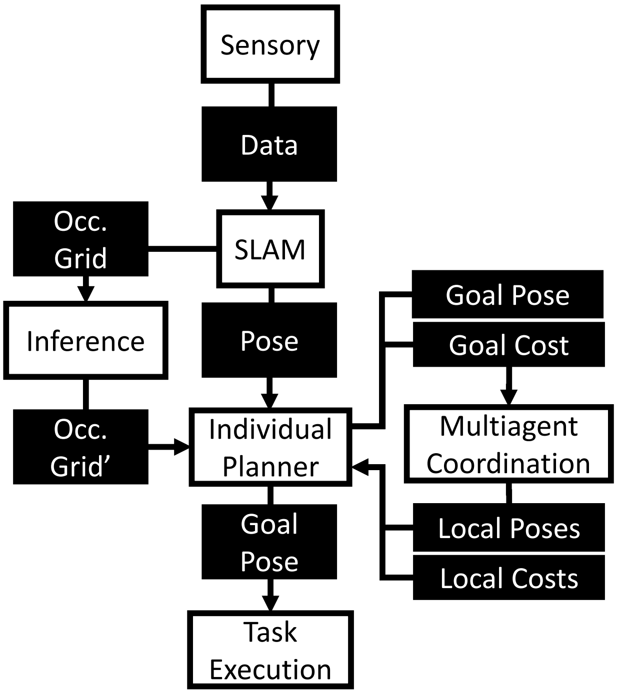

Fig. 6 Overall system architecture. Collected sensory data is used to perform SLAM to generate an occupancy grid or costmap. The costmap is then inferred over to provide additional information for both the individual planner and coordination.

Fig. 7 Example of a reward map. Green indicates areas with low reward, corresponding to places that have been observed, and red indicates area of high reward, corresponding with unobserved inferred areas.

We build upon previous methods of market-based goal selection [6] by improving how the reward of each pose is calculated. The reward of each pose is calculated using the additional information provided by the map inference. To estimate $\hat{r}_p$ a simulated measurement from p is taken to identify the set of observable cells from p, Γ. In this case a laser scanner is simulated, similar to [9], using ray-tracing and every cell, c_i, between the robot and first obstruction of the simulated laser is added to Γ. $\hat{r}_p$ then is the sum of all reward in Γ_p,

$$\hat{r}_p = \sum_{i}^{|\Gamma_p|} \mathbf{reward}(c_i). \tag{11}$$

To calculate the function **reward()**, a reward map is created, giving the reward the agent will receive for observing any given cell. A standard information based approach to reward mapping would use each cell's probability of occupancy to determine its reward, i.e.

$$\mathbf{reward}(c_i) = \text{argmin}(1 - occ_i, occ_i), \tag{12}$$

where occ_i is the Bayesian probability of occupancy of c_i. This reward mapping provides a maximum reward for cells with the highest uncertainty, that is when $occ_i = 0.5$ and the

Exploration goal poses are selected by each agent individually searching their reward map for the set of exploration poses that will form their market, P. This is an iterative process that attempts to find a set of minimal poses, P, that fully observe all available reward, R; Algorithm 3. As there are multiple poses that can observe the same portions of the environment, P is not a unique set. Calculating the optimal set P is infeasible as the number of potential poses an exploring robot can take is in the hundreds or thousands. To approach this problem, we simplify in two ways. First, the observed and inferred space is reduced to a high-level graph representation using the Zhang-Suen thinning algorithm [25], Algorithm 3 Line 7. This produces a skeletonized representation of the space that reduces the size of the set of poses to search. The benefit of the Zhang-Suen thinning algorithm is that it does not remove the poses near the center of each section of the map. This intuitively leaves likely favorable poses for observing the space while removing poses that likely have a restricted view. Second, the high-level graph is then repeatedly sampled to find the members of P. Poses are added to P using an evolutionary algorithm while attempting to satisfy two objectives. The first objective is to select P to maximize the cumulative observed reward, R, of the poses $p \in P$. The second objective, in priority, is to minimize $|P|$.

To search for P, an evolutionary algorithm is used for a predetermined number of iterations for each planning step. In subsequent planning steps the process is repeated to continuously refine and update P throughout the exploration. The algorithm operates by sampling the high-level travel graph, G on line 3, for a potential pose, p, to add to P, Algorithm 3 Line 8. If p observes reward that is not currently observed by a member of P, then p is added to P and the observed reward of p, r_p, is added to the cumulative reward of P, R, Algorithm 3 Lines 10,12-13. The second step is to sample the poses $p \in P$ to identify redundant poses and remove them from P, Algorithm 3 Lines 15-20. If the cumulative reward, R, is unaffected by the removal of p then p is removed from P.

Algorithm 3 : Pose Selection

1: $R :=$: *current reward of P, initially* 0
2: $R_T =$ **getTotalReward()** ◁ cumulative available reward
3: $G =$ **thin**$(I_f \cup O_f)$ ◁ reduce area to be searched
4: $i = 0$
5: $i_{max} :=$ *max iterations*
6: **while** $i < i_{max}$ and $R < R_T$ **do**
7:   **if rand**$(0,1) > \phi$ **then**
8:     $p =$ **samplePose**(G) ◁ sample pose to add to P
9:     $r_p =$ **getReward**(p)
10:     $R' = R \cup r_p$
11:     **if** $|R'| > |R|$ **then** ◁ does p add to P?
12:       $P = P + p$ ◁ yes, add p to P
13:       $R = R'$
14:   **else**◁ prune poses from P
15:     $p = P$.**random member()** ◁ select random $p \in P$
16:     $P' = P - p$
17:     $R' =$ **getReward**(P') ◁ reward of reduced set P
18:     **if** $|R'| == |R|$ **then** ◁ does p add to P?
19:       $P = P'$ ◁ no, remove p from P
20:       $R = R'$
21: **return** P

The result of Algorithm 3 is that set P contains a set of poses that maximizes R while, without a loss in R, attempts to minimize $|P|$. This optimization is considered every time step but only performed when R does not capture all available reward, Algorithm 3 line 6. To evaluate the performance of this optimization 1000 simulations were evaluated and it was found that P needs to be updated on average 14.6% of the planning steps of an exploration. On the planning steps that required P to be updated, on average 72.48 iterations (0.064 seconds) were required for P to fully observe all available reward. In the simulations and hardware results that follow i_{max} was set at 500 iterations.

This method of pruning the pose set allows robots to plan while accounting for all available reward and significantly reducing the search space.

With P identified, the next step in assembling each robot's market is to calculate the reward for $p \in P$

$$\hat{r}_p = \sum_{i}^{|\Gamma_p|} w^{x_0} r_i^{x_0} + \ldots w^{x_n} r_i^{x_n}, \tag{14}$$

where Γ_p is the set of observable cells from pose p, r_i^m and w^m are the reward of c_i and weighting of reward mapping m. Then agents select their next exploration goal, p^*, using

$$p^* = argmax_{p \in \mathbf{P}}(\hat{r}_p - \hat{c}_p), \tag{15}$$

where $\hat{c}_p$ is a weighted travel cost to reach p. Travel costs are calculated using A* on the costmap. As the reward of each pose is a weighted area, a useful weighting is $\hat{c}_p = (travel\ distance)^2$. While the travel cost and exploration reward have different units, the difference between the provides an estimate of the value of traveling to and exploring pose p. The exploration is completed when an agent's market contains no poses of value. In an exploration, this would only occur when there are no accessible inferred or unobserved cells.

3.3 Coordinated Exploration

After each agent, $a_i \in A$, has selected their individual exploration goal poses, $p_i^* \in P^*$, from their individual markets, the next step is to coordinate the exploration with local agents. Each agent broadcasts their goal pose, $p_i^* \in P^*$, and expected travel cost, $\hat{c}_i(p_i^*) \in \hat{C}$, to all agents in communication range. All local agents sharing P^* and $\hat{C}$ effectively form a single bid auction where agents bid with their expected travel costs to claim their goal poses. When each robot broadcasts their goal pose they are attempting to underbid other robots to retain their goal pose. Robots are not allowed to be deceptive in their bids and must broadcast their true expected travel cost. When a robot is underbid on a goal pose it will resolve the conflict by selecting an alternative goal pose from their individual market. This procedure is outlined in Algorithm 4.

Algorithm 4 : Pose Auction For a_0

1: $[bids_{\hat{c}}, bids_p, bids_{id}]$ = **get broadcast bids()**
2: **for** $b \in bids$ **do**
3: $\quad d_{euclid} = \sqrt{(a_0.x - p_b^*.x)^2 + (a_0.y - p_b^*.y)^2}$
4: $\quad$ **if** $d_{euclid} < \hat{c}_b(p_b^*) + \delta$ **then**
5: $\quad\quad \hat{c}_0(p_b^*)$ =**get travel cost**(p_b^*)
6: $\quad\quad$ **if** $\hat{c}_0(p_b^*) < \hat{c}_b(p_b^*) - \delta$ **then**
7: $\quad\quad\quad$ **continue**
8: $\quad\quad$ **if** $\hat{c}_0(p_b^*) <= \hat{c}_b(p_b^*) + \delta$ *and* $id < id_b$ **then**
9: $\quad\quad\quad$ **continue**
10: $\quad \triangleleft$ a_0 is not closer to p_b^* than a_b
11: $\quad$ **discount reward**(bid_p)
12: $P =$ **getPoseSet**(map_r)
13: $p_0^* = argmax_{p \in P}(\hat{v}(p)$
14: **return** p_0^*

By using broadcast goal poses and expected travel costs, robots settle conflicting goal poses using the method outlined in Algorithm 4. This auction-based approach allows each robot to explore their optimal exploration pose, p^* if is not in conflict with other robots. However, as the number of robots participating in the exploration increases the chances of conflicting goal poses increase. Two poses are considered conflicting when they have overlapping observations of the same available reward. Algorithm 4 resolves conflicts by having every agent compare their travel costs and discounting the reward in the predicted observable range of the less expensive pose of the losing robot's reward map, line 11. To reduce computational requirements, Euclidean distance, line 3, is used as an upper bound on travel cost before a more thorough search using A* [26], line 5. Once the bids of local robots have been accounted for, each robots identifies their pose set, P, and uses their internal market to select their next goal pose. The purpose of devaluing, instead of removing, conflicted reward from the reward map is that in the absence of another place to explore, robot a_0 will still move in the direction of the conflicted goal. Although it will be explored when the agent arrives, it is possible that it may branch into new areas that have not been explored, moving the robot closer to unexplored areas.

After the broadcast is made, the receiving robots account for the broadcast bid by calculating the reward remaining in the pose vicinity at the time that they could reach the broadcast pose. Receiving robots that would reach the broadcast pose after the broadcasting robot devalue the exploration reward observed from the the broadcast pose. Receiving robots with the potential to reach the pose before the broadcast robot will ignore the conflict when selecting their pose. Using the adjusted exploration reward, the robots individually plan their exploration while accounting for the anticipated actions of adjacent robots. Each robot uses the adjusted reward map to maximize their individual reward by selecting poses in their internal markets. In this way, the robots coordinate the exploration themselves to achieve the maximum team reward. This approach has the benefit of being fully distributed and lacking a central controller to resolve local disputes and assign goals.

To account for the potential observations of other robots, it is assumed in this work that each robot has identical sensor model; however, it is only necessary that each robot knows the sensor model of its other team members if they are different. While in this work it is a prior assumption, this could be adjusted to each robot broadcasting a description of their sensor model (sensing distance (m) and breadth (rad)) as part of their bid to allow agents to work with team members with initially unknown sensor models. Additionally, each robot maintains a list of recent broadcasts to reduce redundantly exploring a space in the event of communication failure between the acting and previously broadcasting robot.

As an example of how the coordination would work, if robot a_0 is beginning Algorithm 4 it will receive broadcasts from nearby agents a_1 and a_2. Their broadcasts contain their goal pose locations, p_1^* and p_2^*, and the travel costs for each of them to travel to their goal pose, $\hat{c}_1(p_1^*)$ and $\hat{c}_2(p_2^*)$; here travel costs are an estimate of travel distance as all robots are assumed to have the same travel speed. In the case of heterogeneous teams, time of travel can be used to settle disputes instead of travel distance to account for different travel speeds. To check for conflicting goal poses, a_0 checks if it can travel to either p_1^* or p_2^* before a_1 and a_2, respectively. This is done by checking if a_0's predicted travel cost to p_1^*, $\hat{c}_0(p_1^*)$ is less than $\hat{c}_1(p_1^*) - \delta$, where δ is used to account for movement of a_1 since the time of the broadcast, $\delta = \dot{x}\Delta t$. If a_0 can arrive at p_1^* before a_1 then a_0 disregards a_1's bid when selecting p_0^*. To continue the example, a_0 then repeats the process with a_2's bid. In this case, a_0 cannot arrive at p_2^* before a_2, i.e. $\hat{c}_2(p_2^*) < \hat{c}_0(p_2^*)$, and the reward observed from p_2^* is devalued for a_0 to decentivize a_0's exploration in this vicinity, e.g. for $r_i \in S_{p_2^*}$; $r_i' = 0.1 r_i$. If instead the a_0 and a_2 should arrive at nearly the same time, i.e.

$$\hat{c}_2(p_2^*) - \delta < \hat{c}_0(p_2^*) <= \hat{c}_2(p_2^*) + \delta, \tag{16}$$

then agent identification number is used to settle disputes. Ideally agent identification numbers are uniquely assigned before the exploration begins and remain fixed throughout the exploration. However, in practice assigning random fixed identification numbers also works as long as care is taken to reduce matching identification numbers.

Once communication between agents stops, all information exchange between them ceases. Consequently, both agents' stored location of the other agent will likely be incorrect shortly after the loss of communication. However, both of their goal poses (and observed reward in the vicinity) will likely continue to be valid until they reach their goal. This is sufficient for each robot to, initially upon loss of communication, separate and not explore the same goal area. Once either agent reaches their goal and/or selects a new goal, the other agent will no longer be aware of, it or capable of resolving conflicts, until communication between them is restored. While this seems problematic, the lack of coordination in the absence of communication does not cause many problems in practice as the agents that are most likely to be in conflict are also the most likely to be able to communicate.

The result of the inference based coordination is a goal pose for each robot to explore. Each robot uses the observed map structure to make their own inferences about the unobserved portions of the map. Then, robots sample poses from the observed and inferred portions of the map to create an internal market of poses that fully observe the inferred free space. Robots then select the exploration pose with the highest value from their internal market and set it as their exploration goal. Robots settle local conflicts by broadcasting their selected goal pose and travel cost. This approach provides a reasonable method of incorporating the benefits of map inferences into a distributed exploration by building upon many of the strengths of market-based coordination and information-theoretic approaches to exploration. This allows for a fully distributed approach to coordination and informed goal selection.

4 EXPERIMENTS AND RESULTS

As there are two fundamental contributions of this work, the results will be split into two separate sections. First, the inference is evaluated using both collected laser scan data and a series of simulations. The inference is evaluated for both the increase in information provided by the inference (recall) and the accuracy of the information provided (precision). Next, the inference based coordinated exploration is evaluated using simulation against two baselines for exploration efficiency. Finally, both are brought together in an experiment with three Pioneer robots demonstrating the feasibility of the system in practice.

4.1 Inference Quality

To evaluate the performance of the inference, two series of simulated explorations were conducted. The first simulation consisted of a single robot exploring a previously unexplored building using the described market-based frontier exploration method. It is assumed that the agent has a 360^o laser scanner and the ability to perform SLAM. While the robot explored the building it recorded both its observations and the output of the inference. The second simulated exploration uses publicly available laser scan and odometry data sets to perform inference using real world sensor data [27]. Each costmap is compared against a previously recorded map of the environment.

To evaluate the performance of the inference, the commonly used classification metrics of precision and recall are used. Precision describes how accurate the inferred information is:

$$Precision = \frac{|(I_f \cap X_f) \cup O_f|}{|I_f \cup O_f|} \tag{17}$$

where $|(I_f \cap X_f)|$ is the count of correctly inferred free cells and $|O_f|$ is the number of observed free cells. Recall describes how complete the inferred information is and is defined as

$$Recall = \frac{|(I_f \cap X_f) \cup O_f|}{|X_f|}. \tag{18}$$

These two metrics combine to allow for an accurate description of how much additional information is gained by the inference (recall) and how useful that information is (precision). The naive (without inference) exploration of the environment is used as a baseline for comparison. Increases in the inferred recall indicates that the inference provides information describing a larger portion of the map than is currently observed while the inference precisions indicates the accuracy of the inferred information. Large translation or rotation errors in the map inference will result in a significant number of cells being inferred free that are not actually free. As precision is a metric that quantifies the accuracy of the predicted inference, these misclassified cells will reduce the precision.

For the simulations, test environments were randomly selected from the set of maps and then starting locations were randomly selected from unoccupied cells in the chosen map. The robot then explores the map using a greedy frontier based approach without using map inference. At each time step, the exploring robot records the current map information gathered through direct observations, the naive costmap, and provided by the map inference for comparison. We note that prior methods for map inference using visual feature matching did not retrieve meaningful matches from our database of maps because they were unable to search a large enough portion of the database in a reasonable amount of time due to their computational requirements. In contrast, our methods using shape descriptors are able to find matches from the library in real time. Thus, we do not present comparisons to the visual feature based methods here.

Simultaneously recording the naive costmap and the inferred costmap allows the contributions of the inference to be identified. A total of 300 trials were completed. To account for the wide range of exploration times required to explore maps of different sizes, the results have been normalized with respect to mission time duration. Results for recall and precision are presented below in Figures 8 and 9, respectively.

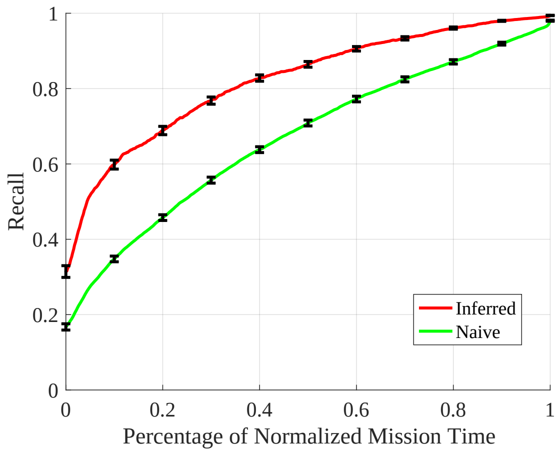

Fig. 8 The mean recall of the naive and inference costmaps for the exploration trials completed. Inferred refers to the method presented in this work while the naive costmap is the costmap without inference; i.e. only the area the robot has directly observed. The error bar indicates the standard error of the mean. Recall provides a measure of how much of the explored space is observed or inferred.

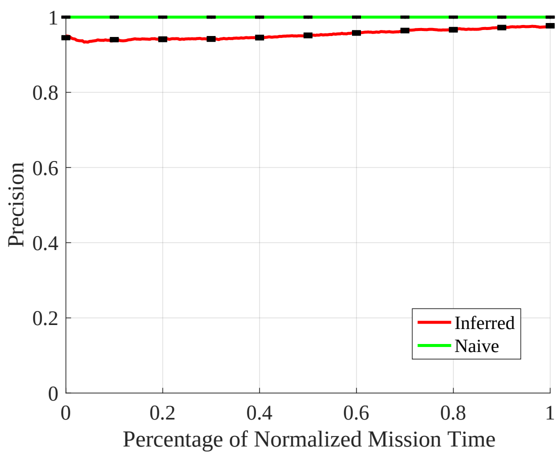

Fig. 9 The mean precision of naive and inference costmaps for the exploration trials completed. Inferred refers to the method presented in this work while the naive costmap is the costmap without inference; i.e. only the area the robot has directly observed. The error bar indicates the standard error of the mean. Precision provides a measure of how correct the observations or inference are. Notice that the naive observations have perfect precision; this is because no measurement errors are provided and all observations are assumed to be correct in the simulator.

As can be seen in Figure 8, a significant amount of information is gained by the inference, especially early in the exploration. The peak mean gain in recall occurs at 5.6% of the exploration duration with a gain in recall of 108.84% measured as a fraction relative to the naive baseline, Figure 10. The mean gain in recall throughout the exploration is 34.47%. This gain in recall provides the agent with an additional 34.47% information with which to plan the exploration. By looking at Figure 9 it can be seen that the information provided by the inference is also accurate and generally infers the map structure correctly, with a mean precision across the exploration of 0.9539 for the inference.

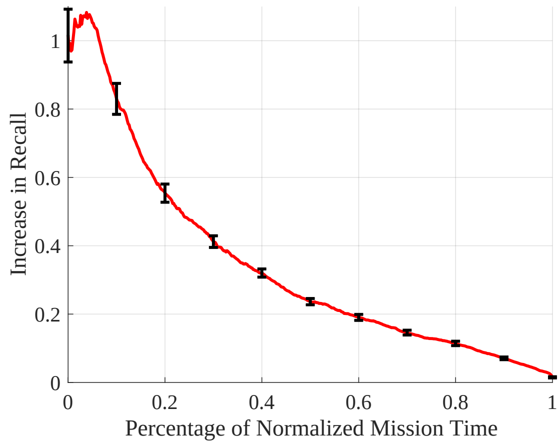

Fig. 10 Proportional benefit of recall by inference versus naive observation.

To evaluate the performance of the map inference using real data a publicly available data set of laser scans was used in new environments [27]. The laser used in the data set has 90^o FOV and returns 180 samples per reading. The inference was implemented using the Robot Operating System [28]. To prepare the laser scan and odometry for inference, the data set is provided to the ROS SLAM G-Mapping project [29] which returns a costmap. The costmap is then inferred over and, again, both the naive and inferred costmaps are recorded for comparisons. The inference used both the structural and perimeter inference with the same structural library as the previous trials.

As can be seen in Figure 12, the inference has a significant increase in the recall over the naive baseline. The precision is noticeably worse than the simulated trials for both the naive baseline and the inferred methods. Using real sensor data complicates the ability to determine the accuracy of the inference for two reasons. First, the "ground truth" map used to evaluate the observed costmap and the inference costmap is a recreated estimate of the true map; i.e. every time a map is created via SLAM it is slightly different. For this reason, the precision of the naive costmap drops below 100% and the inference has a corresponding decrease in precision. Second, the environment is not fully explored in the ”ground truth” costmap. As the inference attempts to predict the complete environment and the exploration is not completed, as shown in Figure 14, it is expected that the inference will suffer a slight decrease in performance in both precision and recall. This is because it is likely that the inferred but unobserved portions of the map are at least partially occupied with free space as inferred. Despite these testing complications, the inference still maintains high precision throughout the exploration as shown in Figure 11 with an appreciable difference in recall as shown in Figures 12 and 13.

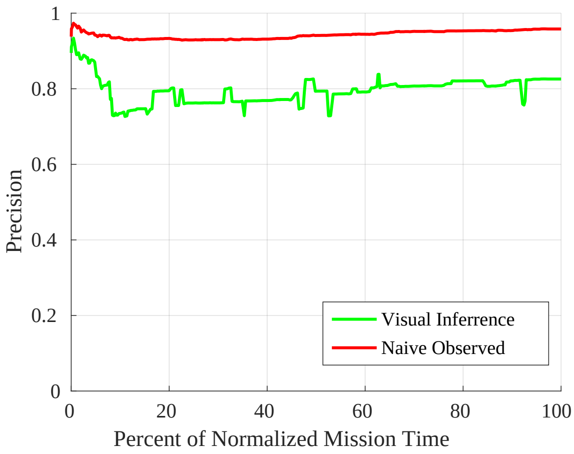

Fig. 11 The precision of the naive and inferred costmaps using the collected laser scans. Precision provides a measure of how correct the observations or inference are. Notice that the naive observations no longer have perfect precision, because of differences in the resulting SLAM maps between iterations.

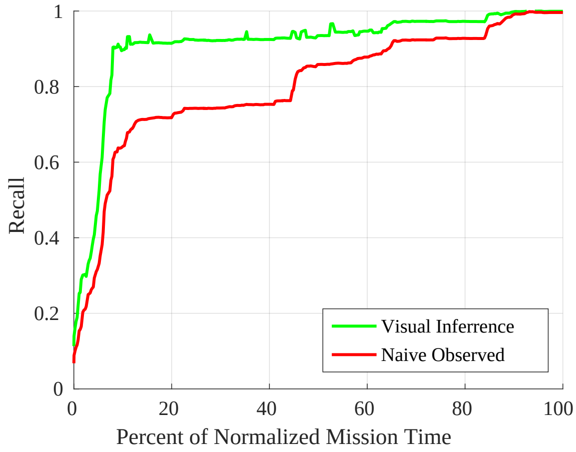

Fig. 12 The recall of the naive and inferred costmaps using the collected laser scans. Recall provides a measure of how much of the explored space is observed or inferred.

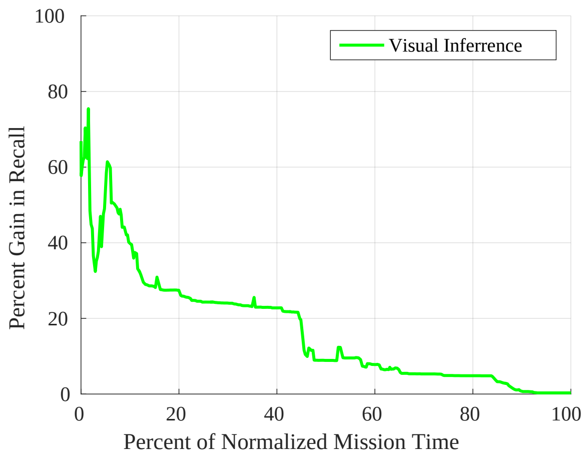

Fig. 13 Proportional benefit of recall by perimeter and structural inference versus naive observation.

4.2 Exploration Efficiency

To evaluate inference’s contribution to exploration efficiency multiple tests were conducted with different exploration methods, levels of communication, and number of agents. For the purpose of these tests, exploration efficiency is measured by the number of time steps required to complete an exploration, i.e. when $O_f = X_f$. The different exploration coordination methods all included a single-bid auction to settle agent disputes with the different exploration goal selection methods. The goal selection methods tested were the proposed inference-informed pose market, a naive (without inference) pose market, and a frontier market. The inference-informed pose market is the developed method introduced in Section 3.2 and is the method being presented in this work. A reward mapping that prioritizes exploration, see Equation 13, with $r^b = 10$ was used. The naive pose market is an identical implementation to the inference-informed pose market method except it does not have the additional information provided by the inference. Consequently, its poses are restricted to the observed space and the pose rewards are expanded to include unknown cells with equal weighting. In this way robots develop a pose set in the observed space that attempt to maximize their observational coverage of the unknown space. The frontier market is included as a well studied baseline. For the frontier market frontier cells were clustered based upon their 2D costmap 8-factor connectedness. The mean center of individual frontier cluster member locations is assigned as the exploration goal of each cluster. Each frontier centroid had a uniform reward, causing agents to operate to minimize their cumulative travel cost when select-

Fig. 14 The fully observed naive costmap (left) and the resulting inferred costmap (right). Notice that though the exploration is complete it is likely not fully explored resulting in a misrepresentation of the perceived ground truth and an underestimate of both the inference recall and precision. This is because, although it is not certain, it is likely that the inferred but unobserved portions of the map are at least partially occupied with free space

ing exploration goals. Robot markets consist of all known frontier clusters and robots. All three methods used an internal market for exploration goal selection and then broadcast their bids in the open local auctions. The primary difference between the three methods was the method of sampling goal poses and calculating each goal pose's reward.

The simulated explorations were performed in a simulation environment developed by the authors. The simulator used a discretized 2D costmap of the world. Environments were included by importing raw images of floor plans, Figure 15, which were discretized into cells that were either free or occupied. The exploration test map was not included in the inference library. All robots have no prior knowledge of the environment they are to explore. Robots have the available actions of moving to any adjacent costmap cell or remain stationary. Robot observations are simulated using ray-casting to detect obstacles and restrict the robot's field of view. Communication is similarly checked using ray-casting to detect obstacles between robots and restrict communication as necessary. There is no stochasticity in the robot's position or observations.

To test the robustness of the developed method, it was tested with varying communication restrictions and numbers of robots in the exploration team. The simulations used the same simulator as the inference verification. For each iteration a random starting location was chosen and all robots started at that location. The communication ability of each agent was varied from unrestricted global communication between all agents to range restricted line-of-sight communication. The range for line of sight communication was varied from 5 to 100 units. Robot sensor range was held fixed at 100 units for all trials. Tests were conducted to identify the performance of the system with two to eight agents. This range was chosen as the exploration gain of additional agents appeared to be negligible. For each test scenario, either number of agents or communication strength, a random starting location was chosen on the map for each test iteration for a total of 50 iterations.

Fig. 15 The test environment used for multi-robot simulated trials. The map has dimensions of 144x108 units. Note: the size of the test environment in meters was not available for every map in the dataset, so we normalize maps to dimensionless units for comparison.

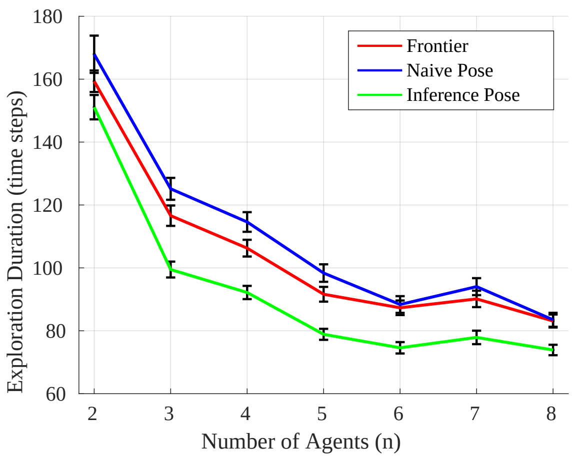

Fig. 16 Exploration efficiency for varying number of agents in the simulated trials. As can be seen across the range of agents tested the inference informed pose selection method results in more efficient exploration of the environment. The error bar indicates the standard error of the mean for the 50 iterations of each of the 7 testing scenarios.

The developed exploration method that leverages map inference to inform pose selection outperforms the naive pose and frontier based explorers by ($\mu = 17.70\%$, $\sigma = 5.37$) and ($\mu = 12.34\%$, $\sigma = 2.97$), respectively, across the range of team sizes tested as shown in Figure 16. This shows that the inference informed exploring team is able to identify better exploration goals for the members of the team resulting in reduced time to fully explore the same space. As the number of team-members is increased the inference informed exploring team continues to outperform the two baselines suggesting the inference leads to improved coordination. It is interesting to note that for the trials conducted with seven agents, all three methods had a reduction in exploration efficiency. This is likely due to there only being 50 simulations for each trial and increasing the number of simulations and varying the environment would remove this behavior.

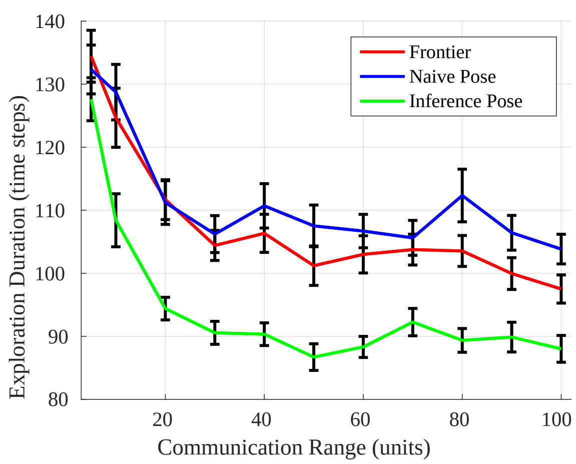

Fig. 17 Exploration efficiency for varying communication ability in the simulated trials with three robots. As can be seen across the range of communication tested the inference informed pose selection method results in more efficient exploration of the environment. The error bar indicates the standard error of the mean for the 50 iterations of each of the 11 testing scenarios.

The developed exploration method that leverages map inference to inform pose selection outperforms the naive pose and frontier based explorers by ($\mu = 16.89\%$, $\sigma = 4.92$) and ($\mu = 13.15\%$, $\sigma = 3.26$), respectively, across the communication ranges tested with three agents. This shows that as the communication between agents degrades the developed pose base inference degrades gracefully. This is because, similar to the two baselines, the inference informed robot uses the auction based approach to resolve local conflicts. When two robots come into contact they resolve exploration conflicts and retain the other robots goal locations. So, even after communication between them has been severed they continue to account for the other robot's broadcast exploration goal. This behavior results in the robots separating from one another and effectively exploring the space even with unreliable communication.

The inference informed pose market method significantly outperforms the other two methods, Figures 16-17. The inference assists the exploration in two ways. First, as discussed before the inference improves the estimate of pose rewards. Second, the inference allows for the marketed poses to be placed in additional locations. The inference informed pose market allows for poses to be placed in $O_f \cup I_f$ and I_f is the unobserved portions of the map. Previous methods that use information theory approaches restrain the poses of exploring robots to the explored portion of the map, O_f. Without an inferred boundary to limit pose placement, poses with the largest reward would be those well outside of the explored region. This is because a simulated sensor reading at this location would predict the maximum available reward in every direction; as there is unobserved space in every direction. In practice this would likely lead to poor performance as little information is known about the true value of exploring this area or if it is even reachable. Our method of map inference solves this problem by placing limits on where in the unobserved space the poses can be placed and inferring limits on the simulated sensor readings. This is done by limiting poses to the inferred free regions and limiting the sensor readings to rewarding only inferred free space.

4.3 Hardware trials

Hardware trials were conducted on Pioneer P3-DX robots, see Figure 18, to verify the functionality of the combined inference and coordination on hardware. For this trial global communication was assumed and a fixed team size of three agents was conducted with each of the coordination methods in one environment. The experiment environment is shown in Figures 19-20. Each Pioneer has an Acer netbook with a Pentium N3530 Processor ($<$ 2.58 GHz) and 4 GB DDR3 Memory and Ubuntu 14.04 LTS operating system. Each pioneer is differential drive and equipped with wheel encoders and an RPLidar 2D Laser Scanner ($\approx 200^o$ FOV) as its sensors for odometry and mapping. Each Pioneer uses the Robot Operating System (ROS) (Indigo) for all computation [28]. ROS packages used for sensor acquisition and SLAM include Rosaria[30], rplidar_ros[31], Rocon [32], and gmapping[29]. Navigation, control, and communication was performed using publicly available pioneer packages [33]. Map merging was performed using iterative-closest-point with initial estimates of team member positions and orientations.

During the hardware trials it was demonstrated that the pioneers were capable of using the inference based coordination to explore an indoor space with existing open source mapping/navigation packages and off-the-shelf laser scanners. Figure 21 shows a sequence of costmaps through out a

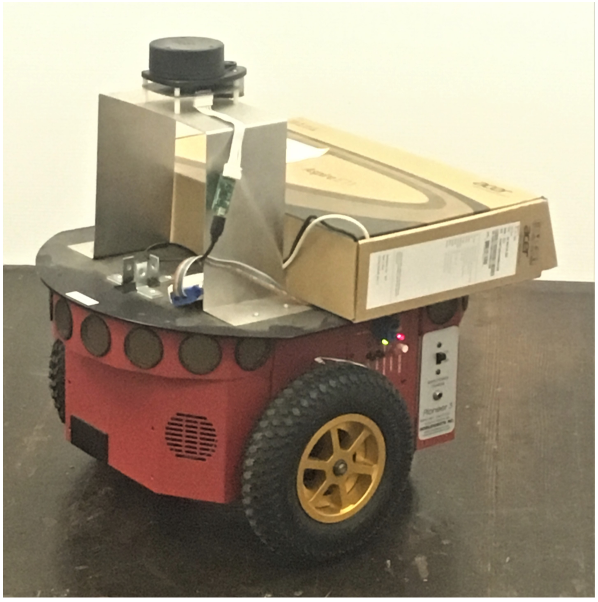

Fig. 18 Pioneer P3-DX used in the hardware experiments.

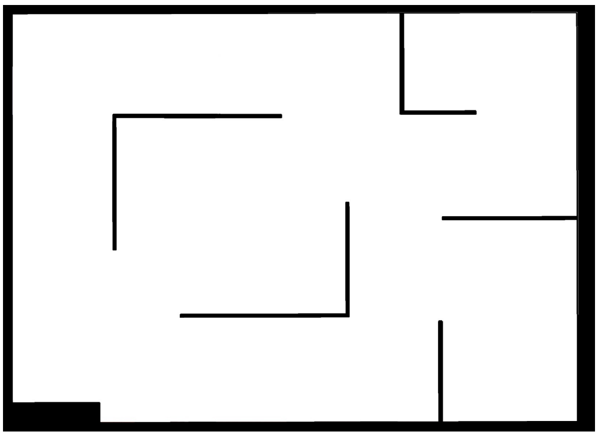

Fig. 19 The test environment used in the hardware trials. Environment is $8.4m$ x $6.1m$.

solo exploration by a pioneer. Each sequence identifies both the pioneer's location and the location of their current goal pose. Figure 22 shows the map constructed by the exploration team and their paths through the environment during the mapping. During the exploration the pioneers each attempt to collect the maximum amount of reward. Pioneer 0 initially infers a large reward forward and to the left. However, upon turning the corner and communicating with Pioneer 1, Pioneer 0 reverses direction to observe the area initially to its right. Pioneer 1 chooses its path to avoid the claimed areas of Pioneers 0 and 2, leaving mainly the room in the bottom right. Pioneer 2, upon observing the hallway in the top left, reverses direction to clear the corner room in the top right. This observed behavior qualitatively confirms the results from the simulations on a team of robots operating in an indoor environment with distributed computation.

Fig. 20 The test environment used in the hardware trials. Picture is taken from right side of map in Figure 19.

5 CONCLUSIONS AND DISCUSSION

This work presents a novel map inference based coordination method for distributed multi-robot exploration. Exploring robots sample potential poses from the observed and inferred portions of the map and use an internal market to select their goal pose for exploration. To resolve conflicting goal poses between agents, each agent broadcasts their goal pose and travel cost in an open auction. Robots use the developed map inference techniques to sample and evaluate potential goal poses. The map inference was tested across 126 different maps and provided an average gain in information of 34.47% with a mean precision of 95.1% in the simulated trials. The inference based exploration method increased the team's cumulative exploration efficiency against a naive (without inference) information pose and frontier based exploring teams in the simulated trials. The distributed coordination method was demonstrated to be robust to varying numbers of agents, outperforming the naive and frontier based exploration methods by 13.15% and 16.89%, and communication ability, outperforming the naive and frontier based exploration methods by 12.34% and 17.70%. The developed system was then demonstrated through hardware trials on a team of robots.

In this paper we addressed some of the current limitations in coordinated exploration. We demonstrated an effective method of coordinated exploration that leverages additional information provided by map inference to increase exploration efficiency. There are multiple extensions that may benefit from the developed techniques. First, individual path planning is non-optimized, which likely decreases the overall system efficiency. Information gathering path optimization methods [34, 35, 8] could be incorporated using simulated sensor readings to further improve system performance.

While exploring, the inference is able to identify *dominated* unexplored contours. These contours are dominated

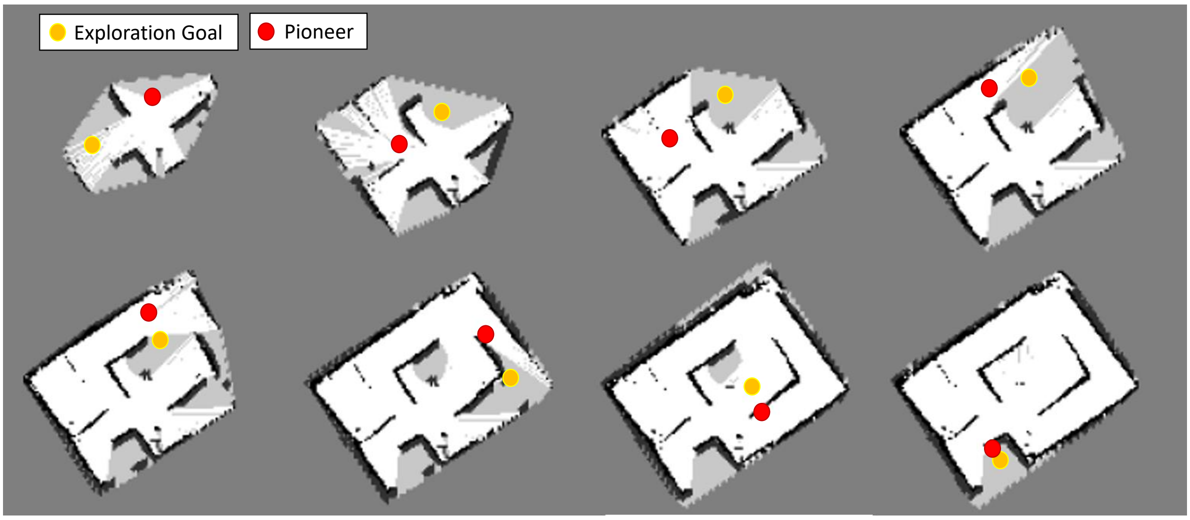

Fig. 21 An example of an inference informed exploration sequence. The pioneer is the dark-shaded circle and its current exploration is the lightly-shaded circle. The sequence shows eight inference updates throughout the exploration proceeding from left to right through the top and then bottom row.

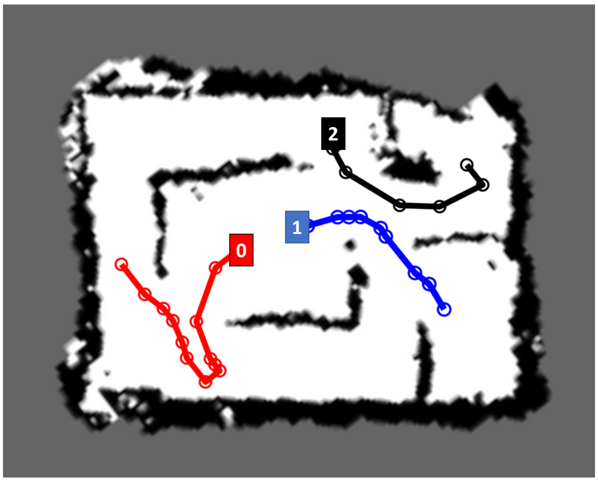

Fig. 22 The resulting merged map of the test environment and the paths of the three pioneers during the exploration. Numbered rectangles indicate starting positions of each pioneer.

in the sense that they are completely surrounded by explored space. Consequently these areas can be reduced in value to disincentive their exploration in favor of areas that are not dominated. This would likely extend the explorable range of the system at a potentially small information cost.

In an energy restricted exploration, common among quadrotors especially, exploring robots are required to select the last place they will explore before their battery expires. One method of extending battery life uses self-organizing pairs of relay and sacrificial robots. Near the end of the exploration the relay will rest, or land, with enough energy to return to the operator and await the sacrificial robot. The sacrificial robot will then explore until their battery decreases to point that is can only reach the relay. The sacrificial agent will then share its observations with the relay and land, unable to return to the operator. The relay will then return both their collected data and the data of the sacrificial robot. This method was demonstrated to increase the exploration range of a system [36]. However, as the potential cost of sacrificing an agent is large it would be beneficial to use inference to predict the value of information gain prior to selecting to sacrifice.

References

1. B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation*. IEEE, 1997, pp. 146–151.
2. N. Kalra, "A market-based framework for tightly-coupled planned coordination in multirobot teams," Ph.D. dissertation, Carnegie Mellon University, 2007.
3. M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proc. of the IEEE 94.7*, pp. 1257–1270, 2006.
4. H. J. Chang, C. G. Lee, Y.-H. Lu, and Y. C. Hu, "P-slam: Simultaneous localization and mapping with environmental-structure prediction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 281–293, 2007.
5. D. P. Strom, F. Nenci, and C. Stachniss, "Predictive exploration considering previously mapped environments," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2761–2766, 2015.
6. R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Robotics and Automation, 2002. Proc. IEEE International Conference on*, vol. 3, 2002, pp. 3016–3023.
7. B. Charrow, "Information-theoretic active perception for multi-robot teams," Ph.D. dissertation, University of Pennsylvania, 2015.

8. C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using rao-blackwellized particle filters.” in *Proc. Robotics: Science and Systems Conf.*, vol. 2, 2005, pp. 65–72.
9. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration,” *Proc. IEEE International Conference on Robotics and Automation*, pp. 476–481, 2000.
10. J. Faigl and M. Kulich, “On determination of goal candidates in frontier-based multi-robot exploration,” *Proc. European Conference on Mobile Robots*, pp. 210–215, 2013.
11. S. Parsons and M. Wooldridge, “Game theory and decision theory in multi-agent systems,” *Proc. Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 3, pp. 243–254, 2002.
12. G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed systems: concepts and design*. Pearson Education, 2005.
13. B. P. Gerkey and M. J. Mataric, “Sold!: Auction methods for multirobot coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
14. C. Tovey, M. G. Lagoudakis, S. Jain, and S. Koenig, “The generation of bidding rules for auction-based robot coordination,” in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*. Springer, 2005, pp. 3–14.
15. M. Cummins and P. Newman, “Highly scalable appearance-only slam fab-map 2.0,” *Proc. Robotics: Science and Systems Conf.*, pp. 1–8, 2009.
16. M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
17. H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Proc. European Conference on Computer Vision*. Springer, 2006, pp. 404–417.
18. H. Zhou, Y. Yuan, and C. Shi, “Object tracking using sift features and mean shift,” *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.
19. P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Robotics-DL*. International Society for Optics and Photonics, 1992, pp. 586–606.
20. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
21. N. Kiryati, Y. Eldar, and A. M. Bruckstein, “A probabilistic hough transform,” *Pattern Recognition*, vol. 24, no. 4, pp. 303–316, 1991.
22. G. R. Waissi and D. F. Rossin, “A sigmoid approximation of the standard normal integral,” *Applied Mathematics and Computation*, vol. 77, no. 1, pp. 91–95, 1996.
23. A. Aydemir, P. Jensfelt, and J. Folkesson, “What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments,” *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4675–4682, 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6386110
24. F. M. Waltz and J. W. Miller, “Efficient algorithm for gaussian blur using finite-state machines,” in *Photonics East*. International Society for Optics and Photonics, 1998, pp. 334–341.
25. C. Zhang, T.and Suen, “A fast parallel algorithm for thinning digital patterns,” *Communications 27*, pp. 235–239, 1984.
26. P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
27. C. Stachniss, “Robotics datasets,” http://www2.informatik.unifreiburg.de/ stachnis/datasets.html, accessed: 2016-012-13.
28. O. S. R. Foundation, “Robot operating system,” http://www.ros.org/, accessed: 2/7/2017.
29. B. Gerkey, “gmapping -package summary,” http://wiki.ros.org/gmapping, edited: 2015-12-09.
30. S. Juri-Kavelj, “Rosaria -package summary,” http://wiki.ros.org/ROSARIA, edited: 2016-05-25.
31. K. Zhao, “rplidar - package summary,” http://wiki.ros.org/rplidar, edited: 2016-12-15.
32. H. K. Daniel Stonier, Jihoon Lee, “rocon -package summary,” http://wiki.ros.org/rocon, edited: 2015-04-07.
33. J. J. Chung, “Pioneer demo code,” https://github.com/JenJenChung, edited: 2016-12-06.
34. G. A. Hollinger and G. S. Sukhatme, “Sampling-based robotic information gathering algorithms,” *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.
35. B. Charrow, G. Kahn, S. Patil, and S. Liu, “Information-theoretic planning with trajectory optimization for dense 3d mapping,” *Proc. Robotics Science and Systems Conf. 2015*.
36. K. Cesare, R. Skeele, S.-h. Yoo, Y. Zhang, and G. Hollinger, “Multi-uav exploration with limited communication and battery,” *Proc. IEEE International Conference on Robotics and Automation*, no. 1, 2014.