Market-Based Collaborative Robot Exploration

Declan O’Beirne and Michael Schukat

Abstract This paper considers the problem of using miniature low-cost robots for real-world tasks. The issues of low quality sensor data, inaccurate odometry, low processing capacity and limited power are approached through the adoption of a flexible, distributed control system. The system is employed in the context of collaborative multi-robot exploration. A market framework used to support efficient task selection and formation of coalitions between robots, where collaboration is employed to increase the accuracy of generated maps.

1 Introduction

Low-cost, small-scale robots offer a cheaper alternative to mobile robots often demonstrated in academic domains. They are also easier to setup to perform different tasks and offer flexibility and robustness where deployed in larger groups. They may, however, be subject to constrained computational power, poor sensor quality and inaccurate self-localization. For such robots to offer a viable solution in real world scenarios, the control system should be robust to hardware failure or changes in the robot’s state. Autonomous control should therefore be employed without reliance on instructions from a central coordinator. Additionally, the fact that a large number of robots is deployed should be exploited to overcome inaccurate actuators and poor quality sensors.

To this end, this paper presents a framework for autonomous control, allowing a group of robots to collaborate without requiring centralised coordination. An implementation of the framework in the context of robot exploration is also presented,
employing short-lived coalitions in order to perform cooperative localization and increase the scale and accuracy of the generated map.

The framework encompasses a behavior-based control system. Simple, reactive control schemata are suited to low-cost robots’ limited sensor range and processing power, affording fast response to changes in the environment and adoption of different roles in order to collaborate when appropriate. Robot control is based on a profit-centric system, allowing robots to efficiently select and trade tasks.

Robots were constructed with minimal-cost, off-the-shelf components in order to perform collaborative exploration experiments over small areas and to provide a model for a simulation framework in order to extend the experiments to larger areas and larger numbers of robots. The experiments demonstrate the deployment of a robust, flexible control framework and show that through collaboration a team of robots can improve the accuracy and scale of the generated map.

2 Related Work

In the domain of multi-robot control, distributed approaches offer a number of theoretical advantages over centralised ones, the most immediate being a reduction in complexity of analysis of sensor data and modelling of an environment, flexibility to changes in the environment and robustness to failure.

Swarm-based approaches are inspired by self-coordinating techniques in social animals. Di Caro et al. [9] describe an approach inspired by pheromone diffusion in ant colonies to coordinate a set of robots to form a sensor network over an environment in order to guide robots to a target point. The approach iteratively updates the network and requires that line-of-sight is maintained between robots. Thus, it may not be suited to unknown, potentially large environments where robots may have to move to new areas or adopt different tasks. Li et al. [8] present a swarm-based approach where two robots must collaborate in order to move an object. No explicit communication is employed; instead robots learn an optimal length of time to wait at an object before relenting and moving to a different object. While suited to straightforward task-selection, this approach does not consider if collaborations formed are optimal or allow for close coordination through explicit coalitions.

Parker et al. [17] present a priority-based task allocation technique where tasks are dynamically ordered in terms of priority and assigned to robots based on suitability. The system allows robots to react quickly to a changing environment but requires that the priorities of tasks be dictated to robots. Market frameworks offer efficient coordination by providing an accurate model of benefit and cost with which to direct robot actions. The TraderBots system [1], for example, employs an agent to auction tasks to robots, with robots bidding to perform tasks and trading with each other in return for cooperating on a coordinated plan. When used with low-cost robots, such frameworks may have to function in the presence of noisy sensor data, poor self-localization and unreliable communication. Approaches to overcome these problems include re-auctioning tasks upon determining that a robot has failed
or forming sub-networks in a group within which robots hold auctions [3]. A behavior-based control system described in [4] uses a set of social interaction rules to define how robots initiate auctions and how winners are determined, thus providing a flexible mechanism allowing robots to partake in auctions based on their location and current behavior.

Cooperative localization has been demonstrated in the literature as an effective approach to state estimation in environments where GPS is not available. Rekleitis et al. [5] demonstrate an approach using two robots, one remaining stationary while the other moves, maintaining visual contact at all times and thus achieving lower error accumulation relative to odometry. Approaches described in [6] and [10] allow the entire group of robots to move and combine their relative location estimates using an EKF. Experiments demonstrate that the accumulation of uncertainty depends on the accuracy of the robots' proprioceptive position estimates and the number of robots in the group. This approach is therefore ill-suited for use with a group of inaccurate, low-cost robots exploring concurrently.

Gutirrez et al. [7] describe a swarm-based cooperative localization system, where pairs of robots share estimates upon establishing communication. As all robots move, uncertainty will grow unbounded until either a goal is reached or the robot becomes lost. Thus, while the approach may be suited to a foraging task in a limited area, it is not suited to exploration in an environment of unknown size.

With specific regard to exploration with small, low-cost robots, Rothermich et al. [11] utilise cooperative localization in a SLAM approach developed for use with a robotic swarm. Here, robots use a simple control schema which directs them to navigate or act as landmarks depending on their proximity to other robots and to the state of the environment around them. Robots communicate their desire to move to other robots. Cooperation then emerges as robots competing for the same task will acquiesce to the robot with the highest stated desire. The exploration implementation was quite basic though — robots do not directly sense the environment themselves, but infer free space from visibility of other robots.

The Robot Operating System [15] provides a framework for deploying multiple robots, using XML-RPC to transfer instructions and data between robots and modules. The use of high-level frameworks, RPC and separate processes allow for a modular design and deployment on a range of hardware platforms. However, these features also impose sizeable power and computation overheads on low-cost, low-power processors often designed with a single core. The Fawkes system [16] provides a more light-weight solution, with additional modules provided as dynamic libraries that can be integrated at run-time.

The implementation presented in this work builds on these approaches, allowing further customization of the system according to the hardware and experimental requirements, while being written in native code and remaining light-weight. Behaviors can be added to the system via a plug-in interface, while low-level functionality such as kinematics or sensor processing is abstracted away from the robot logic, allowing the system to be hardware-agnostic.
3 Market-Based Collaborative Exploration

The collaboration framework presented in this work is built on a behavior-based robot control system. A robot’s decision-making process is founded on a motivation to maximise its personal profit. The requirement that a robot receives income in order to act necessitates the existence of an additional agent to remunerate it. The system therefore incorporates one or more agents with which a robot can trade.

Behaviors are adopted based on utility, $u$. Each behavior module calculates the task that will result in the highest $u$, and the winning behavior is adopted. Utility is calculated as $u = \frac{p_g - s}{r}$, where $p_g$ is the quantity of revenue received, $s$ is expenditure, or revenue traded to another robot or agent, and $r$ is resources, or anything which is exhaustible and therefore may limit the robot’s operability, e.g. battery life or time remaining. An example of expenditure, $s$, is revenue traded to a partner robot in exchange for acting as a stationary landmark.

Along with utility-based behaviors, the control system employs imperative behaviors which the robot must adopt in order to avoid damage or remain functional. The robot may temporarily switch to imperative behaviors and then revert to the utility-based behavior once they have completed. Behavior state is maintained using a behavior stack. In an example scenario where an imminent collision is detected while exploring, the control system pushes FollowPath and AvoidCollision onto the stack. AvoidCollision ensures that the robot immediately stops actuators or changing course. Once complete, this behavior is popped and FollowPath adopted. This calculates and follows a path around the obstacle to the target area. Upon successfully traversing the path, this behavior is also popped and the control system reverts to exploration.

Coalitions are used to achieve close coordination between robots when appropriate. Robots collaborate in a self-interested manner, joining coalitions when this will be more profitable than working independently. Coalition formation is instigated by a robot broadcasting a proposal that one or more robots join it to collaborate on a task. Upon receipt of a proposal, robots may bid in order to join. In order to simplify the bidding process, in the experiments described here, all robots calculate bids according to a fixed function and winning bids are determined using a first-price sealed-bid auction.

3.1 Map Building with Multiple Low-Cost Robots

The framework described in this work has been employed for multi-robot exploration. In order to motivate self-interested robots to explore terrain, a Map Aggregator Agent (MAA) was introduced, which pays an agreed amount of revenue for map data submitted to it.

The physical robots used to verify the control framework and the collaborative exploration approach presented in this work were equipped with low-cost monocular cameras. These are inexpensive and capable of providing detailed information.
Each robot was equipped with appearance information and precise dimensions of other robots, allowing for robust relative localization. Limited field of view and low-resolution images with high noise levels, issues often inherent in low-cost cameras, impair the ability to detect and classify landmark features that would be required for a feature-based map representation. Therefore, a grid-based representation is instead used to model the environment.

The MAA pays revenue for map data based on its associated positional accuracy. A covariance is used to approximate a probability distribution over the position estimate of a map cell. An error-magnitude metric, \( e \), is used to quantify uncertainty. This is calculated as \( e = \frac{a}{\pi} \), where \( a \) is the area of the covariance matrix’s error ellipse at one standard deviation. Revenue, or gross profit, \( p_g \), is calculated as \( p_g = k_m (1 - e/e_{\max}) \), where \( e_{\max} \) is the maximum permitted error magnitude in the experiment and \( k_m \) is the map revenue coefficient. The value of \( k_m \) determines the importance placed on generating map data relative to other tasks. In the experiments described in section 4, mapping is the only profitable task so \( k_m \) is set to 1.0.

### 3.2 Loop-Closing with Low-Cost Robots

Low-cost visual sensors may typically be defined by low-resolution and high noise levels, thus limiting the number of landmark features detectable and increasing the likelihood of incorrect classification. For the micro-robots used in this work, having a sensor range of less than 1m, simultaneous localization and mapping is infeasible and robot state is determined by odometry alone, making the robots prone to accumulation of large localization error.

To avail of the presence of other robots, the framework supports behaviors to actively perform cooperative localization. Robots form coalitions, agreeing to adopt complementary behaviors. An exploring robot will submit map data to the global map, and upon detecting a partner robot acting as a stationary landmark it will update its positional estimate and update the set of map scans in the loop just closed. The original map scans may then be wiped from the global map. Thus, at any time a maximum of two state hypotheses is ever maintained.

The task of robustly detecting loop closes by analysing map features has been the subject of much recent research [12][13]. The framework presented here adopts a greatly simplified approach, closing loops only upon detecting a supervisor robot with a distinct, salient appearance, with each robot provided with the dimensions and appearance information of the other robots.

If a robot has performed \( n \) moves along a path and subsequently detected a supervisor robot to update its position estimate, \( p_n \), to \( p_n' \), then the adjustment applied to each position along the path, \( p_i \), will be \( p_i' = p_i + \frac{i-1}{n-1} (p_n' - p_n) \). This makes the assumption that error had been accumulated uniformly over the loop [14]. This assumption was verified experimentally by measuring the actual and estimated position of a single robot performing autonomous exploration for 100 iterations over a 4m\(^2\) demarked area.
Thus, for the situated robots described in section 4, the error covariance associated with a short forward move was $\begin{bmatrix} 48.43 \text{mm}^2 & 14.67 \text{mm}^2 \\ 14.67 \text{mm}^2 & 0.60.72 \text{mm}^2 \end{bmatrix}$. Upon estimating the offset of node $n$ along the path navigated, node $n - 1$ could be estimated with accumulation of uncertainty $\begin{bmatrix} 19.17 \text{mm}^2 & 3.38 \text{mm}^2 \\ 3.38 \text{mm}^2 & 23.08 \text{mm}^2 \end{bmatrix}$.

### 3.3 Behaviors for Collaborative Exploration with Multiple Low-Cost Robots

#### Local-Area Exploration

Local-Area Exploration (LAE) is a profit-based behavior. It enables a robot to gather map data to exchange with another agent for revenue. LAE directs a robot to explore the environment in a greedy manner, considering target areas equal in size to the area covered by the robot’s sensors. Due to limited processing and memory resources, robots consider targets only within a local map centred on their location.

The utility, $u$, of pursuing an LAE target is calculated as $u = p_g - s - r$, where gross profit, $p_g$, is revenue received from the MAA, as described in section 3.1, expenditure, $s$, is revenue that must be attributed to another robot or another behavior and $r$ quantifies the resources required to reach this target.

If the robot had previously employed Wide-Area Exploration (WAE) to navigate to an unexplored region of the environment to explore, then a proportion of the revenue earned by LAE in that region will be attributed to WAE. The value attributed is calculated as $s = p_g k_W$, where $k_W$ is set when configuring the robot to instruct it to either explore more widely or to explore local areas more comprehensively.

The parameters of the exploration mission will determine how $r$ is calculated, but this will typically encompass the units of battery power expended, units of time taken and accumulation of positional uncertainty. Accumulating positional uncertainty will diminish the robot’s ability to earn revenue later in the experiment. The increase in error magnitude, $e$, is translated to resource units, $r_e$, as $r_e = k_e (e' - e)$, where $k_e$ is trained iteratively from experiment data. By plotting the change in gross profit $p_g$ over error magnitude $e$, $\frac{dp_g}{de}$ and integrating over the intervals $e_a, e_{max}$ and $e_a', e_{max}$, the potential loss in profit due to accumulated error can be estimated.

#### Wide-Area Exploration

Wide-Area Exploration (WAE) directs the robot to navigate to unexplored areas of the environment. This behavior does not itself focus on generating map data, but facilitates other behaviors such as LAE in doing so. To allow the robot to adopt such long-term goals in favour of short-term profit, the behavior management module is configured with a set of rules for attributing revenue to external agents or behaviors.
After adopting WAE to navigate to an unexplored region, a specific proportion of revenue earned through LAE in that region must be attributed to WAE.

As with LAE, the utility is calculated as $u = \frac{p_g - s}{r}$. Here, $p_g$ is calculated based on the estimated revenue attributed from LAE targets, $p_g = k_W n_L p_L$. Here, the coefficient $k_W$ determines the proportion of profit LAE targets are obliged to attribute. $n_L$ indicates the estimated number of LAE targets that will be adopted before moving to another region and $p_L$ estimates the mean gross profit per LAE target. These values are estimated from experiment results, based on the proportion of unmapped terrain in the exploration region.

**Loop Closing**

A robot undertaking exploration while in coalition with a supervisor may adopt CloseLoop in order to establish visual contact and close a loop along a path it has travelled. Upon correcting its positional estimate, the robot can propagate the adjustment back through the list of poses along the path it has travelled. If the uncertainty at any pose is reduced, then the map scan taken at that pose can be re-submitted to the global map and further revenue obtained from the MAA.

The behavior’s profitability is calculated as $u = \frac{p_g - s}{r}$, where $p_g$ is revenue received from the MAA for updated map data, $s$ is expenditure, or revenue allocated to the robot’s partner robot acting as supervisor, and $r$ is resources expended in establishing visual contact.

Revenue for updated map data is calculated as $p_g = \sum_{i=1}^{n} k_m m_i \left( \frac{e_i - e'_i}{e_{\text{max}}} \right)$, where the robot has gathered map scans at $n$ poses along the path it has traversed, $k_m$ is a map revenue coefficient as broadcast by the MAA, $m_i$ is the number of map cells submitted in map scan $i$, $e_i$ is the original error magnitude at map scan $i$, $e'_i$ the updated error magnitude and $e_{\text{max}}$ the largest error magnitude permitted by the MAA.

Revenue, $s$, is attributed to the supervisor in accordance with the agreement made in the coalition established between the robots. When considering a proposal to form a coalition, robots bid the proportion of revenue that they will agree to attribute to the supervisor. In the experiments described in section 4, robots calculate this proportion based on the minimum utility that would make joining the coalition more profitable than other profit-based behaviors, LAE, WAE and Supervision. Thus, if $\frac{p_g - s}{r} = k_{\text{CL}} \max(u_{\text{LAE}}, u_{\text{WAE}}, u_{\text{SUP}})$, where $k_{\text{CL}}$ is a constant defining the threshold for adopting CloseLoop, e.g. in experiments in section 4 this is 1.01, then $s = p_g - r k_{\text{CL}} \max(u_{\text{LAE}}, u_{\text{WAE}}, u_{\text{SUP}})$.

**Supervision**

A robot may adopt the Supervision behavior when a proposal to create a coalition has been accepted by one or more partner robots. The robot acts as a landmark
for exploring robots, enabling them to explore the surrounding area and use the landmark to remove error accumulated along their paths. The behavior does not direct the robot to directly earn revenue itself. Instead, it facilitates the generation of revenue by robots implementing other behaviors, for which it receives a specified proportion.

The utility of a Supervision target is calculated as $u = \frac{p_{g}}{r}$. Resources, $r$, that the robot will expend on this target is calculated based on the battery power required to reach the Supervision area, the battery power consumed while acting as a stationary landmark and the time expired. The time required for the region to be explored is estimated from experimental results based on the number of unexplored LAE targets in the region.

Gross profit, or revenue attributed from other robots, $p_{g}$, is calculated by estimating the number of loop-closes that can be carried out before the region is explored and the average revenue attributed per loop-close.

4 Experiments and Analysis

The framework was tested in experiments with two situated robots in unmodified indoor environments, as shown in figure 1, and in simulated experiments with eight robots modelled on the kinematic and sensor data from these. The situated robots were constructed with minimal-cost components including 300MHz ARM processors, CMUCam2 cameras, Bluetooth and off-the-shelf platforms and motors. Due to the low-resolution and high levels of noise in images captured, the cameras were oriented downwards at approximately 30 degrees to the horizon to allow increased reliability in obstacle detection. This resulted in a typical maximum sensor range of 0.59m from the robot’s centre of gravity. The robots measured 0.15m in length, with the camera mounted at 0.1m above the ground. The covariance matrix modelling relative localization uncertainty at the optimal distance of approximately 0.3m was calculated as $\begin{bmatrix} 457.8740mm^2 & 112.8287mm^2 \\ 112.8287mm^2 & 536.4766mm^2 \end{bmatrix}$. The values were calculated from a set of 207 images taken with two robots positioned over a delimited grid, with half the images taken to train an appearance model of the visible robot and half to test detection and localization.

The simulated environment consisted of a grid of cells, 4cm$^2$ in area, marked as occupied or free space. The total area of the simulated environment was 70.46m$^2$. In order to detect obstacles and other robots in simulation, the image processing module used by situated robots was replaced with a module to perform line-tracing against the environment. Error in relative localization of robots and other artefacts was inserted based on test results. Robot motion was simulated based on kinematic data from physical robots. Again, error in self-localization was inserted based on calibration data. Communication between robots and agents was identical for situated and simulated robots, with simulated robots connecting via the localhost instead of via Bluetooth. Thus, the code used in situated and simulated robots was
largely identical with only the sensor and actuator modules replaced based on the build configuration selected.

The simulation supports re-running of situated experiments in order to verify that the simulation accurately represents the physical environment and to provide a debugging facility that is otherwise not available given the situated robots’ limited resources. All sensor data from the situated experiment is serialized and then processed by simulated robots. The log files from simulated and situated experiments can then be shown to be identical, allowing for differences in floating-point accuracy.

![Experimental setup with 2 situated robots.](image)

Figure 2 presents map data of the physical environment depicted in figure 1, generated by two robots performing exploration with and without collaboration. When collaboration was enabled, the two robots were free to join coalitions as they saw fit. They each switched between independent, supervisor and explorer modes.

Due to excessive latency in transferring images from the camera module to the CPU, 5.2 seconds to transfer a 75KB image over a 115,200 baud serial connection, the experiments were limited to 15 minutes. In the case of the independent exploration experiment, the robots became immobilised before this after having accumulated positional uncertainty exceeding the permitted error magnitude.

When collaborating, the robots spent the majority of the experiment in various coalitions, with either of the robots acting as a stationary landmark during this time. The map was therefore generated more slowly than with independent exploration. Thus, as the collaborative experiment was limited by the allocated time, the map generated by the robot was smaller — 35m² compared to 39.8m². Collaboration, however, enabled the robots to achieve much greater accuracy. The mean error magnitude encoded in grid cells in the final map was 4.05 map units² as opposed to 8.53 for independent exploration, i.e. a reduction in 52.5%. A comparison between the resulting maps shows that artefacts have been mapped with much greater accuracy, e.g. in figure 2(b) there is an offset of 0.3m between two sections of the same wall,
C, while the outline of the two boxes, A and B, in the centre of the map are distorted to a much greater degree than in figure 2(a).

![Collaborative exploration](image1) ![Independent exploration](image2)

**Fig. 2** Map generated by 2 robots in the environment shown in figure 1. The two boxes in the foreground of figure 1 are shown as artefacts A and B. The wall in the background is shown as artefact C

Tables 1 and 2 show parameters from simulated experiments with up to 8 robots exploring independently and employing the collaboration framework respectively. The simulated experiments shown here were limited to 800 iterations, equating to approximately 15 minutes of operation with situated robots — the typical battery life before a large degradation in power given the hardware in use. The maximum permitted error magnitude accepted by the MAA was set to 10 map units, or 20cm. When the robot accumulates an error in its positional estimate of this magnitude, it is no longer able to earn revenue and is deactivated.

In all independent exploration experiments shown, the robots exceeded the maximum error magnitude before 800 iterations; on average at 460 iterations. For the experiments utilizing the collaboration framework, however, the robots were able to complete the experiment without exceeding this limit, with a median error magnitude of 9.96. Independent exploration results in map data being generated more quickly, with an average of 35% more grid cells being explored. The accuracy of the resulting map is much poorer than when collaboration is employed though.

For example, when 8 robots are deployed, the average error magnitude for grid cells in the MAA’s final map is 194% greater for independent exploration. While the robots approached the point of exhausting the number of available exploration targets in both cases, the robots employing collaboration could have carried on to explore a much larger error given greater battery power.

Figures 3 and 4 show the maps generated from the simulated environment for independent and collaborative exploration. While there is no explicit allocation of exploration targets to robots within the group, robots are implicitly repelled from
Table 1 Simulated experiments with robots acting independently

<table>
<thead>
<tr>
<th>N robots</th>
<th>Avg error magnitude</th>
<th>Area mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.514748</td>
<td>34310</td>
</tr>
<tr>
<td>2</td>
<td>5.281091</td>
<td>59543</td>
</tr>
<tr>
<td>4</td>
<td>5.236238</td>
<td>112505</td>
</tr>
<tr>
<td>8</td>
<td>4.821904</td>
<td>204165</td>
</tr>
</tbody>
</table>

Table 2 Simulated experiments with robots utilizing the collaboration framework

<table>
<thead>
<tr>
<th>N robots</th>
<th>Avg error magnitude</th>
<th>Area mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.603300</td>
<td>44240</td>
</tr>
<tr>
<td>4</td>
<td>1.452967</td>
<td>76520</td>
</tr>
<tr>
<td>8</td>
<td>1.640644</td>
<td>161923</td>
</tr>
</tbody>
</table>

each other based on the heuristics used to estimate the probability of reaching targets first. By maintaining a shared state of the environment, robots are able to largely avoid duplicated work, with close to a linear increase in area mapped. The environment itself is shown in figure 7(a). Here, cells of value 0 (black) represent obstacles and 255 (white) represent open terrain. In the generated maps, occupancy and error magnitude for each mapped cell are encoded in 1 byte. A default value of 127 signifies that terrain is not mapped, while values in the range 0..126 signify mapped occupied terrain and 128..254 open terrain. Positional error is thus encoded with a granularity of $e_{max}/127$.

Fig. 3 Maps generated in a simulated environment employing independent exploration

Figures 5 and 6 show accumulation of error in the robots’ positional estimates in simulated exploration experiments with and without the capacity to form coalitions to perform cooperative localization. Where the goals of the mission require accurate localization, i.e. the robot’s actions are of reduced or no utility when the robot ac-
cumulates uncertainty beyond a given threshold, the duration and thus the scope of the experiment can be greatly increased through collaboration.

Figure 7 shows a simulated environment and maps generated under different configurations. Robot activity is limited by the maximum permitted error magnitude that the MAA will accept, in this case 20 map units$^2$.

This experiment demonstrates the flexibility of the control framework, in that by adjusting the rule controlling the allocation of revenue from LAE to WAE, the propensity of the robot to adopt one robot over the other can be controlled. It can be seen that, when WAE is allocated 0.8 of revenue, the robot maps more distant parts on the environment but generates a lesser amount of map data.
Fig. 6  Errors in positional estimates in a simulated experiment with 2 robots exploring with collaboration.

![Robot localization error with collaboration](image)

(a) 101.6m$^2$ simulated environment  (b) Map generated with WAE profit = 0.1

(c) Map generated with WAE profit = 0.5  (d) Map generated with WAE profit = 0.8

Fig. 7  Comparison of maps generated with a single robot in a simulated environment where behavior configuration is altered. Here, the behavior management system is configured to impose a rule that WAE targets be attributed a certain proportion of revenue earned by LAE targets.
5 Conclusion

The distributed, market-based robot collaboration framework presented in this work has been demonstrated in the context of multi-robot exploration using minimal-cost robots. The framework has been employed for the task of exploring simulated and real-world environments. By dynamically forming coalitions to perform cooperative localization, the team of robots can increase the accuracy and the scale of the map that it can produce. Due to the extensible design the framework can be applied to further tasks suitable for low-cost robots and to greater numbers of robots.

References