VISION-BASED MULTI-AGENT SLAM FOR HUMANOID ROBOTS

Jonathan Bagot, John Anderson, Jacky Baltes

Department of Computer Science
University of Manitoba
Winnipeg, Canada R3T2N2
bagotj,andersj,jacky@cs.umanitoba.ca

ABSTRACT

A solution to the SLAM problem using multiple homogeneous humanoid robots with limited processing power, noisy sensor data, and inconsistent locomotion is described and implemented on two real humanoid robots. The solution uses particle filters and the concept of frontier-based exploration.

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a critical problem to be solved in any autonomous robotics application where the environment is even partly unknown. A solution to the SLAM problem requires three answering three basic questions [4]: "Where am I?", "Where am I going?", and "How do I get there?". The solution of each of these questions individually may be greatly simplified by knowledge of the environment (e.g. known landmarks allow recognition of one's location more easily). The fact that this knowledge is lacking, along with the fact that these problems must be tackled simultaneously, is what makes the SLAM problem so difficult. In an unknown environment, for example, a robot has no initial landmarks to use for localization, and must develop these as it explores. Similarly, since mapping is done from the perspective of an agent's current pose (coordinates and orientation), the quality of a map is only as good as the quality of an agent's estimation of its pose.

There are many approaches to SLAM in the literature (See section 2). In our work, we are interested in a number of specific elements that complicate this problem significantly, beyond the general case:

- Vision: Our robots rely on vision from a single forward-facing camera. Most applications dealing with the SLAM problem rely on wide-range, high-resolution laser scanners. Single-camera vision provides a large amount of data that must be interpreted in real time, but covers a correspondingly narrower area without depth information. Moreover, the real-time element of this problem means that only very basic interpretation of images can be performed, resulting in far more noise than would be found in other SLAM-based approaches.
- Limited Computational Ability: the processors our robots work with are mobile embedded systems of limited processing power. Much of this limited power must be devoted to interpreting visual frames, as well as to the robot application at hand. This both leaves little remaining computational ability to a SLAM algorithm, and compounds the previous problem in that there is a low frame rate for vision and greater noise in visual interpretation.
- Humanoid Robots: most of the well-known approaches to SLAM were created and demonstrated on wheeled robots operating on flat surfaces. Humanoid robots, on the other hand, have many more degrees of freedom (DOF) in their physical construction. This means that even on a flat surface, a camera attached to a robot will have much variation in its positioning (e.g. as the robot bends to take a step), leading to much greater noise and difficulty in interpreting a stream of images. This makes the SLAM problem significantly more challenging for humanoid robots using vision.
- Efficient Exploration: we wish to cover an unknown domain efficiently; this means attempting to avoid redundant area coverage.

A SLAM solution gradually builds a map by mapping visible spatial area relative to the current estimated pose of an agent. Therefore any odometry error during motion propagates to landmark location in the map. The fact that we are using humanoid robots also greatly increases odometry error, since there are no simple mechanisms such as wheel encoders to provide reasonable dead reckoning during motion.

In addition to these factors, there are other elements that also complicate the SLAM problem in the real world, as opposed to laboratory environments: the presence of obstacles and other robots means that agents must be able to navigate without colliding with obstacles while maintaining a good estimate of their pose, and the fact that some obstacles are mobile means that there will be greater noise that must be overcome in order to create an accurate map.

The constraints of the SLAM problem, along with the desire for efficient exploration and limited computational abilities, point to the use of multiple agents in this problem. Using more than one agent in a SLAM approach should be able to increase the accuracy of a map through multiple perspectives and the ability to reduce one another's odometry error. In addition, if multiple agents can simultaneously select different unexplored targets and avoid exploration overlap, the amount of exploration that can be performed should be able to be significantly increased, up to the limit where physical interference prevents information gain [8].

The presence of multiple agents should also work to counter limitations on individual robots. Assuming communication is available, the amount of information that can be obtained about the environment by multiple agents in communication with one another should have a greater impact on the SLAM problem than that of n agents operating individually, since each new landmark serves to make future work in SLAM more accurate. Another significant limitation is the battery power available on any one robot: working with a single agent would mean that any significant domain would be impossible to completely map. Other forms of individual limitation can be similarly overcome: Battery life may inhibit an agent from mapping a large environment, and some areas may be inaccessible due to a particular agent's locomotion abilities. Multiple agents, possibly heterogeneous, can increase the coverage percentage by using each agent's resources more effectively.

This paper presents a novel approach to Multi-Agent SLAM. While others (most notably [3, 7]) have developed approaches to multi-agent SLAM, we are moving beyond the limitations of these works to add the complexity of working with humanoids, vision, and limited computation discussed in this section. The approach presented here is demonstrated using homogeneous humanoid robots relying solely on the vision capture capabilities and processing power of cellular phones. In order to answer the "Where am I" question, every agent tracks its pose over time with a particle filter given only sensor feedback from a single forward facing camera on a cellular phone attached to the robot. Landmarks are mapped relative to the best particle (estimated agent pose) in an occupancy grid with a recency value associated with each grid cell. Each agent communicates its estimated pose, all landmarks in its local map, and its current target pose to other agents. All agents select target poses such that each target pose satisfies a couple of simple constraints which help increase the coverage percentage of the environment and reduce the time to cover the environment.

2. RELATED WORK

One major contribution to the inaccuracy of maps generated by SLAM algorithms is the accumulation of odometry error caused by dead reckoning. Rekleitis et. al. [7] used two or more agents to reduce the accumulation of odometry error caused by dead reckoning. Agents were modeled as points which could move in any direction with two types of sensors, an object detector which could sense objects nearby and a robot tracker which could determine the distance and orientation of another robot within line of sight. The environment was modeled as a large polygon decomposed into trapezoids which were then cut into stripes. Only one agent moved at a time along a stripe of a trapezoid while all other agents observed its movement. When the moving agent stopped, its location was updated based on the observations of other agents and its role was reversed to an observing agent, allowing the next agent to move. If the moving agent remained within line of sight of at least one other agent at all times no dead reckoning was ever required, therefore this approach reduced odometry error and increased map accuracy. In this work, we operate with only a single sensor, and no restrictions on the movements of any agent.

One hurdle multiple agents must overcome to solve the SLAM problem that a single agent is not faced with is map merging. Birk and Carpin [2] merged maps from multiple agents without knowledge of an agent's pose relative to others. The best merging is determined by maximizing overlap between two maps. A search for the best merging requires keeping one map fixed and continuously rotating and translating the other. Birk and Carpin used a random walk algorithm guided by a heuristic which required a relatively large amount of processing power. The search space for the best merging is extremely large, therefore map merging may have to be centralized when the poses of agents relative to one another are unknown, unless mobile agents have the resources necessary to perform many rotations and translations. Our work does not require the merging of occupancy grids, but uses the exchange of landmarks to update maps individually.

In order to take advantage of the distribution of multiple agents to reduce the time to cover the environment, agents must somehow be directed to unexplored areas. Yamauchi [9, 10] introduced the concept of a frontier (a region bounding open and unexplored space) and frontier-based exploration using multiple agents in a form of occupancy grid. An occupancy grid associates a probability with each grid cell representing the probability that cell is occupied. If the stored probability is less than the prior probability, the grid cell is open. Similarly, If the probability is equal to or greater than the prior probability, the grid cell status is unknown and occupied, respectively. In Yamauchi's work, agents had both a global and local occupancy grid. The local occupancy grid of an agent was constructed and sent to all other agents when its target frontier was reached. The global occupancy grid of an agent was an integration of all agents local occupancy grids. A decentralized, asynchronous approach allowed addition and removal of agents to the exploration team without consequence since information was shared and control was independent. If agents were allowed to navigate to the same frontier, the solution was not optimal due to interference.

Anderson and Papanikolopoulos [1] improved on this work, comparing Local Target Search and Shared Target

Search. For Local Target Search, agents shared local information about open search areas only, while for Shared Target Search, agents had a shared map with a search strategy that relied on a global list of unexplored areas and methods for preventing agents from selecting the same targets. This comparison provides evidence that multi-agent search with lightweight communication protocols can still improve performance without explicit coordination.

Burgard et. al. [3] coordinated exploration such that agents do not select the same frontier. Their approach used occupancy grids and the concept of frontiers once again, but made the assumption agents knew their relative position. Target frontiers were selected for each agent with a utility/cost metric. The utility was the expected visibility range from the target frontier given the probability another agent's target frontier may have visual overlap. The cost was the optimal path from the agent to the frontier. The algorithm iteratively chose a target frontier for an agent and reduced utility of nearby unexplored cells. Their experimental results show that preventing the simultaneous selection of targets by agents reduce the time to cover the environment with 2 real robots and 3 robots in simulation.

3. HOMOGENEOUS HUMANOID ROBOTS

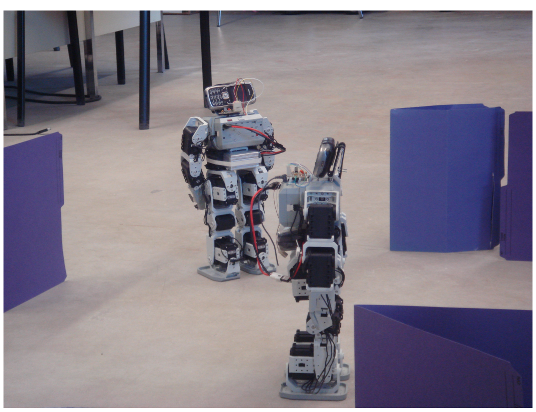

Figure 1: Rogue and Storm.

The homogeneous robots used to conduct this research (Figure 1) are humanoid robots with eighteen degrees of freedom based on Robotis's Bioloid kit. Each robot is equipped with eighteen Dynamixel AX-12 servos (three in each arm, five in each leg, and two in the torso) capable of producing 16.5kgf.cm of torque at 10V. One servo in the arm effects motion in the sagittal plane, while the other two effect motion in the frontal plane. Three servos in the leg effect motion in the sagittal plane and the other two effect motion in the frontal plane. Both servos in the torso effect motion in the traverse plane.

An on-board Atmel AVR ATmega128 micro-controller and Nokia 5500 cellular telephone are interfaced by a custom made infrared data association (IrDA) board containing a Microchip MCP2150 standard protocol stack controller supporting data terminal equipment (DTE) applications. The on-board micro-controller is predominately used for communication with the servos, including but not limited to tasks such as position interpolation and load checking. It is also used for the storage and playback of static motions created by our motion editor software written by us. This is all made possible by our custom firmware running on our multi-threaded real time operating system (RTOS) FreezerOS also written by us.

The Nokia 5500 provides a full C++ development environment, robust operating system (SymbianOS 9.1 series 60 release 3.0), camera, communication mediums (Bluetooth and IrDA), an ARM 9 235MHz processor, and a three axis accelerometer (LIS302DL). The Nokia's processor is used for state generation, image processing, sensor data smoothing, and application programs (including the SLAM approach described here).

Everything except the Nokia 5500, which has it's own battery is powered by one lithium-ion polymer battery pack. We have been custom-modifying these robots since April, 2007, and various versions have competed in the (FIRA) 2007 HuroCup in San Francisco, and RoboCup 2007 in Atlanta. They will also be competing in the FIRA 2008 HuroCup and RoboCup 2008.

4. ENVIRONMENT

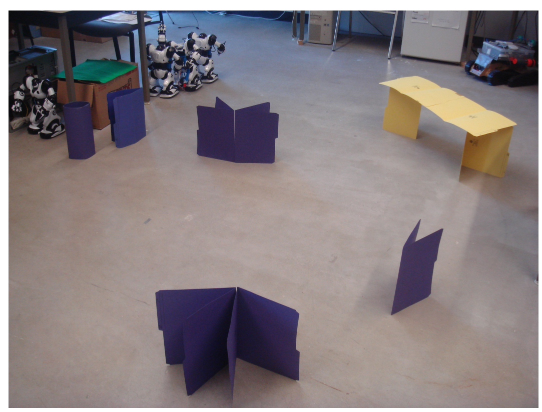

Figure 2: Several Wall obstacles (dark) and a Gate obstacle (upper right, light).

The environment in which we are exploring multi-agent SLAM is composed of randomly-placed *wall* and *gate* obstacles as shown in Figure 2. These are simply cardboard folders, colored for recognition purposes. The purpose of differentiating two types of obstacles is to allow the robots to deal with them in physiologically-specific ways (e.g. moving around a wall, while crawling under a gate). This environment is significantly more challenging for SLAM purposes than many robotic applications (e.g. soccer), since there is no knowledge of how many of each obstacle exists, or any relationship in obstacle placement.

5. METHODOLOGY

Our SLAM approach, consists of the use of a particle filter on individual robots to allow an estimation of their current pose, a methodology for mapping, a methodology for exchanging and merging mapped information, and a method for selecting frontiers to reduce redundant exploration. Each of these are explained in the following subsections.

5.1. Particle Filter

The particle filter we employ is a variation on that used by Rekleitis [6], differing in the motion model and particle weight update method. Each particle in the filter is a weighted estimate of the agent's pose. After an action (e.g. a left, right, forward, or backward rotation or translation) the pose estimate of each particle is updated based on the motion model, then the weights are updated based on the sensor feedback.

Any action by a robot will not necessarily produce the same physical result. The motion model in a particle filter estimates changes in orientation and position after an action by modeling rotations and translations with some random noise added. Rekleitis used wheeled robots, and a wheeled robot cannot translate in a direction 90° from its current orientation without first performing a rotation. A humanoid robot does not have this restriction (e.g. it can perform a side step). The motion model has been adapted to account for such a translation.

Pose estimation with dead reckoning in our work is also problematic compared to Rekleitis, due to significantly greater odometry error present in humanoid robots and the fact that we are using a single camera whose view is altered by the DOF of the robot, rather than a fixed laser scanner. After an action, the pose estimate of each particle is updated based on the motion model. If there was no sensor feedback, the pose estimate of each particle would suffer from this accumulation of odometry error. Our image processing returns the polar coordinates of objects in the camera's field of view, but camera data during the humanoid robot's locomotion is extremely noisy due to motion blur. Our weight update method uses a certainty factor in the camera data and a constant decay. The best particle is the weighted average of all particles. The particle population size is 100, which is manageable with our limited processing power, but successful results [5] have been reported with a particle population size of 50. Population depletion is handled with a simple select with replacement re-sampling algorithm as used by Rekleitis [6].

5.2. Map Representation

Every agent's local map is stored as an occupancy grid with 25x25cm grid cells. A recency value $[0, 255]$ is associated with each grid cell instead of the more common posterior probability. If the recency value of a grid cell is greater than zero, a landmark exists in the corresponding grid cell.

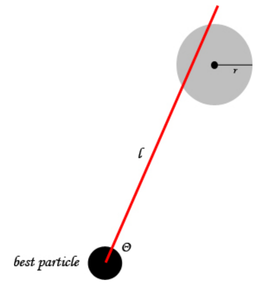

Figure 3: Recency update method.

The recency value in occupancy grid cells is updated by an increment or decrement depending on the current sensor reading. If the sensor senses an object, and the coordinates of the object relative to the best particle in the particle filter map to a grid cell with a recency value greater than zero, then the recency value is incremented; otherwise, the grid cell recency value is initialized to 128. If the sensor does not sense an object, landmarks are extended to circles with radius r, if a line segment with length l (maximum sensor range) extended from the best particle intersects a landmark circle, the recency of the corresponding grid cell is decremented(Figure 3).

The advantage of using recency values instead of posterior probabilities is that floating point calculations are avoided, which is desirable given our limited processing power. The recency value can also be useful for handling erroneous sensor readings or dynamic environments since the recency value will eventually decrease to zero under these conditions.

5.3. Communication and Map Merging

A decentralized, asynchronous communication approach is used between agents via Bluetooth over the logical link control and adaptation protocol (L2CAP) layer. No agent ever waits or relies on information from other agents. An agent uses only what information is available, therefore agents can join or leave the SLAM team at any time without consequence. This also means unreliable communication links between agents are not a problem, beyond the lack of information that results when communication goes down: each agent can still operate independently. Each agent communicates its estimated pose, all landmarks in its local map, and its current target pose to other agents in messages encoded such that the size of each message is as small as possible.

Because entire maps are not exchanged, there is no merging of occupancy grids. Instead, communicated landmarks are integrated into the agent's own map individually through recency update. There are two important elements in this, understanding the local coordinates of others, and actually integrating this information.

To foster a global coordinate system, we adopt the sequential deployment technique of [1]. Agents enter the environment one after another from the same pose, which results in the same unique origin in a local coordinate system for each agent. Thus, when describing the location of a landmark, no rotation or translation information is required. The weakness of this is the error in local pose estimation, but that itself should be improved over time as SLAM unfolds. Internally, each agent maintains the pose of the nearest agent and their current target pose.

To integrate communicated landmarks, we use the recency update method described previously, and assume agents can trust one another (in the sense that there is no duplicity in communication, and that each agent is running an approach such as this one to limit localization error). If the landmark already exists in the agent's map, the greater recency value is selected and the corresponding grid cell is updated. If the landmark does not exist in the agent's map, the corresponding grid cell is simply updated with the received recency value.

5.4. Frontier Selection

Unlike [3], we use no centralized method for dealing with frontier selection, and unlike [1], we do not maintain a shared set of targets. Instead, we attempt to minimize coverage overlap between agents without communication, by exploiting the fact that each agent maintains an estimation of the pose of the closest agent to it. As the SLAM approach described in this section unfolds, agents select target poses (desired coordinates and orientation for motion planning) such that each must be a frontier on its local map, but in addition, the Euclidean distance from the target pose to the nearest agent must also be greater than the maximum sensor range.

6. EXPERIMENTATION

To examine this approach, we compared using one and two agents in a set of three environments with random wall and gate obstacles such as that shown in Figure 2. The accuracy of the map, coverage percentage of the environment, and the time to cover the environment were recorded and compared. The accuracy of the generated maps were compared to hand constructed maps as shown in the left half of Figure 4, and ranked based on the following scoring method; for each grid cell, if the grid cell in the generated and hand constructed map were both occupied or both unoccupied by a landmark, the score was incremented by one, otherwise it was decremented by one. For a 16x16 grid with 25x25cm grid cells the maximum and minimum scores are thus 256 and -256 respectively. The coverage percentage of the environment was determined by examining the distinctive clusters of landmarks in the generated map and their location compared to the hand constructed map (the right half of Figure 4). The time to cover the environment was recorded where agents were allowed a maximum of 5 minutes to map the environment or stopped if the coverage percentage was equal to 100%.

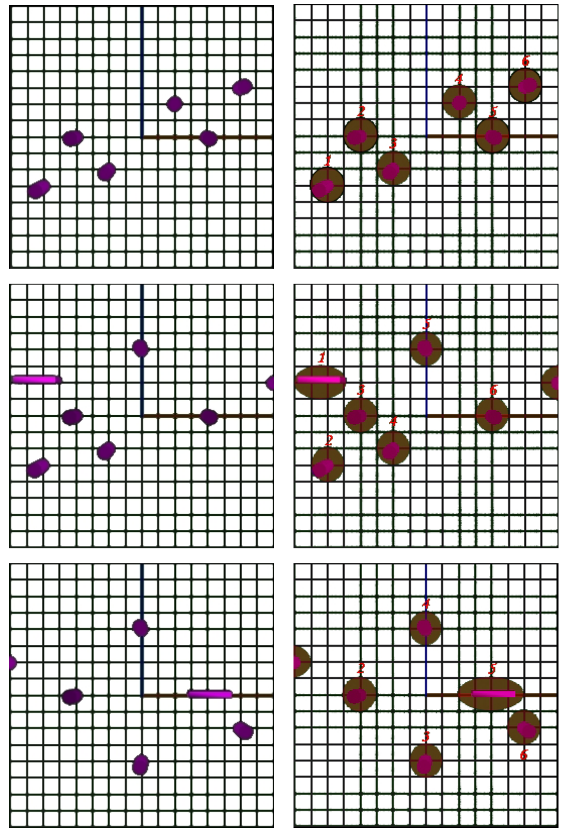

Figure 4: Hand constructed maps (left) and associated clusters (right) for three different environments.

Figure 5 illustrates snapshots of the maps in progress, created by one- and two-agent teams (left and right respectively) for the three environments. Yellow and purple markers differentiate the contribution of landmarks from each agent, and as the recency of a landmark increases, its color becomes brighter. Dark and light blue arrows represent the best particle of each agent, and dark and light red spheres represent agent targets. A video demonstration of this is also available at `http://aalab.cs.umanitoba.ca/videos/MASlam.mpg`.

7. DISCUSSION

In this paper an approach to multi-agent SLAM for humanoid robots was presented, where each robot uses only a single forward-facing camera. The concept of recency values for occupancy grid cells was introduced as an alternative to posterior probabilities. By using two simple constraints agents were able to select target poses which helped increase the coverage percentage of the environment and reduce the time to cover the environment. The solution described was implemented and demonstrated on two real humanoid robots.

The results in Table 1 show that the accuracy of the maps do not vary appreciably by having a two-agent team. This is in part an indication of just how challenging the SLAM problem is in this environment: the approach used

Table 1: Results

Number of agents	Map	Accuracy score	Coverage percentage	Time to cover
1	1	248	33.33%	5:00min
1	2	238	14.29%	5:00min
1	3	243	50.00%	5:00min
2	1	240	50.00%	5:00min
2	2	226	42.86%	5:00min
2	3	239	50.00%	5:00min

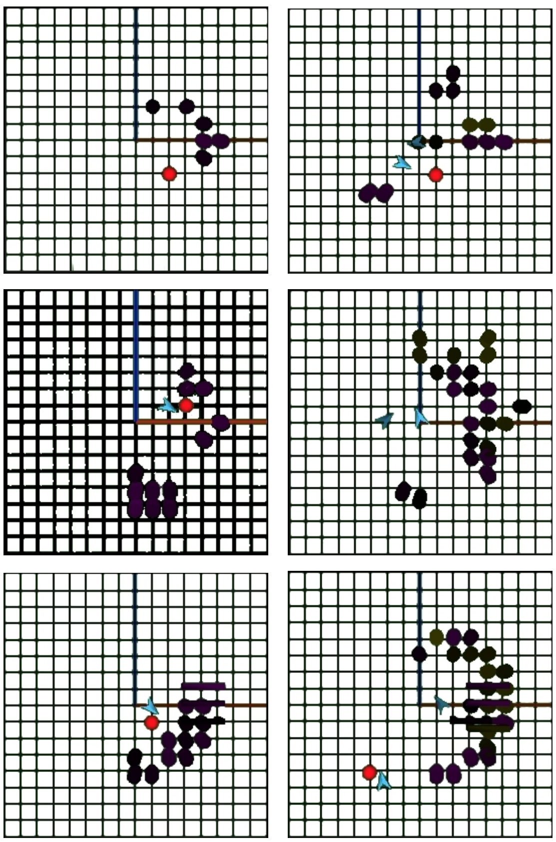

Figure 5: Generated maps by one agent (left), and two agents (right) for each of three environments.

here would be easily deployable in an environment like soccer, where the recognition of one landmark (e.g. a goal) greatly constrains the possible positions of the remainder. While we are pleased with the coverage percentage increase, the processing power limits the accuracy of perception and therefore localization in a domain with unrelated landmarks, and two robots is not enough to provide an increase in confidence of values to show the power of this approach. Further work will thus involve demonstrating this with additional agents.

We also intend to combine data from the accelerometer and a model of the robot's walking gait to allow the inclusion of distance traveled as an input to the particle filter, which should greatly improve these results. this data will also improve the estimation of the position of the camera as well. Ultimately, we intend to operate with heterogeneous agents without requiring a sequential deployment strategy.

8. REFERENCES

[1] ANDERSON, M., AND PAPANIKOLOPOULOS, N. Improving multirobot, cooperative search via local target queues. In *Proceedings of IEEE IROS-07* (San Diego, CA, USA, 2007), pp. 2590–2595.

[2] BIRK, A., AND CARPIN, S. Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE 94*, 7 (July 2006), 1384–1397.

[3] BURGARD, W., FOX, D., MOORS, M., SIMMONS, R., AND THRUN, S. Collaborative multi-robot exploration. In *Proceedings IEEE ICRA-00* (San Francisco, CA, USA, 2000), pp. 476–481.

[4] FRESE, U. A discussion of simultaneous localization and mapping. *Autonomous Robots 20*, 1 (February 2006), 25–42.

[5] KWOK, C., AND FOX, D. Map-based multiple model tracking of a moving object. In *Proceedings of RoboCup-2004* (2004), pp. 18–33.

[6] REKLEITIS, I. *Cooperative Localization and Multi-Robot Exploration*. PhD thesis, McGill University, January 2003.

[7] REKLEITIS, I., DUDEK, G., AND MILIOS, E. E. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proceedings of IJCAI-97* (Nagoya, Japan, 1997), pp. 1340–1346.

[8] ROSENFELD, A., KAMINKA, G., AND KRAUS, S. A study of scalability properties in robotic teams. In *Coordination of Large-Scale Multiagent Systems*, P. Scerri, R. Vincent, and R. Mailler, Eds. Springer-Verlag, 2006, pp. 27–51.

[9] YAMAUCHI, B. Frontier-based exploration using multiple robots. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents* (New York, NY, USA, 1998), pp. 47–53.

[10] YAMAUCHI, B., SCHULTZ, A., AND ADAMS, W. Mobile robot exploration and map-building with continuous localization. In *Proceedings of IEEE ICRA-98)* (Leuven, Belgium, 1998), pp. 3715–3720.