Localization of Multiple Robots with Simple Sensors

Mike Peasgood and Christopher Clark
Lab for Intelligent and Autonomous Robotics
Department of Mechanical Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada
{*mike, chris*}*@lair.uwaterloo.ca*

John McPhee
Motion Research Group
Department of Systems Design Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada
mcphee@real.uwaterloo.ca

***Abstract*— This paper presents a distributed particle filter algorithm for localizing multiple mobile robots that are equipped only with low cost/low power sensors. This method is applicable to multi-micro robot systems, where size limitations restrict sensor selection (e.g. small infrared range finders).**

Localization of three robots in a known environment is conducted by combining measurements from a small number of simple range sensors with inter-robot distances obtained through an acoustic range finder system. The localization problem is formulated as estimating the global position and orientation of a single triangle, where corners of the triangle represent the positions of robots. The robot positions relative to the centroid of the triangle are then determined by trilateration using the inter-robot distance measurements. Each robot uses an identical particle filter algorithm to estimate the global position of the triangle. The best estimates determined by each particle filter are distributed among the robots for use in the following iteration. Simulations demonstrate the ability to perform global localization of three robots, each using a compass and two range finders. The results illustrate that this method can globally localize the robot team in a simulated indoor environment. The results are compared to simulations where robots have access to only their own sensor data, which are unable to successfully localize under equivalent conditions.

***Index Terms*— localization, multi-robot, triangulation, particle filter**

I. Introduction

Cooperating teams of robots can add greater simultaneous presence, force multiplication, and robustness to a robotic mission. In particular, micro robot teams are valuable in a large variety of applications like space exploration, where weight must be minimized to reduce transportation costs.

Micro robots are typically limited in both computational resources and sensor capabilities, which limits their level of autonomy and intelligent behavior. The computational limitations have been addressed by at least two different approaches. One is to use behavior-based control strategies [1], which often have lower computational requirements. However, such strategies navigate without a world model, making them less useful for mapping applications. An alternative approach, performing planning based control using a world model, may be accomplished using micro robot teams within a hierarchy of robots. In such a system, larger robots integrate sensor information from smaller robots and assist with the higher level computations required for localization and path planning [2]. By centralizing some functions of the team control, however, some of the benefits of redundancy inherent in a multi-robot system are lost.

In this paper we address the second limitation of micro robots, that of limited sensor capabilities, which significantly impacts their suitability for exploration applications. For such applications, a planning based control architecture is applicable, where the robots maintain a model of their environment, either given from prior knowledge of the working area, or learned during exploration. Before using this model to navigate within the environment, the robots must first localize themselves, that is, estimate their positions within the environment. This estimation process has been the subject of a great deal of recent research [3]. Localization has two different sub-problems: position tracking from a known starting location, and (the more challenging) global position estimation where no estimate of the initial location is given.

This paper focusses on the problem of performing global localization of multiple robots, given that their sensors are limited in size, power, and number. Presented is a method of distributing the problem of global localization across a team of robots, where three robots are selected to work cooperatively. This method extends traditional particle filter algorithms in two ways. First, we propose an alternative state representation for a team of three robots, effectively reducing the number of variables to estimate. Secondly, the particle filter calculations are distributed across the team, and the best position estimates are shared at each iteration of the algorithm. By sharing their limited sensory data and computational resources, the team is capable of achieving global localization that cannot be accomplished by an individual robot.

II. Background

A number of researchers have investigated methods of localizing robots relative to one another within a group. Kato [4] presents a method of identifying other robots and determining their relative positions using omnidirectional

[*] This work is partially supported by NSERC

vision sensors. Grabowski [5] presents a method using omnidirectional sonar sensors to estimate the distance between each robot pair, and uses trilateration from three stationary robots to determine the relative positions of others. These relative localization methods determine the position of an individual robot relative to the others in the group, but depend on the knowledge of initial positions to estimate their absolute positions in the environment.

Global localization requires the additional ability to the estimate the robots absolute position in a known map with no prior knowledge of the robot position. Fox [6] describes a Monte Carlo based method for cooperative global localization that synchronizes the beliefs of robots when they detect and recognize one another. The merging of beliefs provides a dramatic improvement in performance over individual localization, assuming that the robots have sensors capable of accurately locating and identifying other robots in the group. The work in [7] also gives a method for improving the performance of localization and exploration by two robots in a polygonal environment, where one robot remains stationary and the other uses it as a reference during movement. Presented in [8] is a distributed Extended Kalman Filter-based algorithm, as applied to the localization of a team of heterogeneous robots operating in outdoor terrain. The algorithm is demonstrated using a variety of sensors including GPS, scanning lasers and cameras, which are beyond the capabilities of the small robots considered here.

The problem of global localization of multiple robots using low cost sensors is addressed in [9]. In that solution, a CMOS camera is used to estimate the distance between robots, and geometric features (straight walls and corners) are identified using IR sensors on a rotating base. The method depends on the ability to identify particular features, such as corners, to generate a set of possible positions of each robot in the environment. The relative distances between robots could then be used to identify a unique solution. In contrast, the method presented in this paper does not rely on the extraction of features from measurements, or in the identification and explicit measurement of the other robots' positions.

In this paper, robots are only equipped with relatively simple sensors, including two fixed range sensors, a compass, odometry, and a measure of the distances between pairs of robots. Despite this limited combination of sensors, a group of three robots is capable of performing global localization using variations on a particle filter algorithm.

III. Problem Definition

The method presented addresses the problem of global localization of a group of three robots within a known map of their environment. It is assumed that the robots each have 2 fixed-position range sensors (IR for example), a compass to sense orientation, and a measure of odometry. In addition, each robot has a mechanism to measure the distance (but not direction) to the other robots, such as an omnidirectional acoustic range sensor (such as that described by Navarro [10] for example). Limitations on computational power and sensor range are not considered in the current simulation, but will be investigated in future work. The task is to estimate the global position of all three robots in the given map.

IV. Localization Algorithm

A. Overview

Particle filter localization uses a large number of particles – i.e., state estimates – to approximate the probability distribution of the robot being at any location in the environment [3]. A weight is associated with each particle, representing the confidence in that particle's estimate. To initialize the filter, m particles are selected from the configuration space with a uniform distribution, and the weights are set to $1/m$. At each iteration of the algorithm, the set of particles are updated with the following processes:

- *Sampling*:
  Particles are drawn from the previous set with probability proportional to their weights.
- *State Update*:
  The state of each particle is updated to account for the robot motion (estimated from odometry) for the current time step.
- *Weighting*:
  Weights are computed for each particle, as a function of the difference between the robot sensor measurements z and the predicted measurements $\hat{z}$ based on the estimated position and map data.

The algorithm presented in this paper uses the particle filter approach described above, but rather than estimate the position of a single robot, it estimates the pose of a triangle with the 3 robots at the corners. That is, each particle represents an estimate of the triangle's pose, defined by the variables $\{x_c, y_c, \theta_c\}$ giving the global position of the centroid and the orientation of the triangle. From the estimate of the position and orientation of the centroid of the triangle, and the measured distances between each pair of robots, the estimated position of each robot can then be computed. The weight of each particle in the particle filter represents the belief in a particular configuration of all three robots.

B. State Representation

The full configuration space for the three robots is defined by the 9-dimensional space of $\{x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3\}$. If the headings $\{\theta_1, \theta_2, \theta_3\}$ are determined solely by a compass on each robot, the remaining variables to estimate are the global position variables $\{x_1, y_1, x_2, y_2, x_3, y_3\}$.

If the distances between the robots are known, the state can be more compactly represented in three variables, $\{x_c, y_c, \theta_c\}$, where the subscript c identifies a reference frame C, the centroid of a triangle with the robots at the corners.

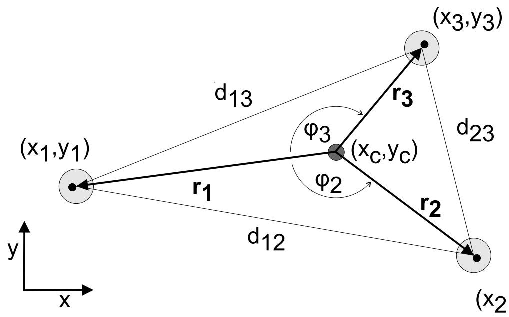

Fig. 1. Graphical estimation of robot positions given the centroid reference frame at (x_c, y_c) and the measured distances between robots $d_{i,j}$. See text for details.

θ_c defines the orientation of the reference frame, where the x-axis is aligned with one (arbitrarily selected) median of the triangle. In reducing the dimension of the state space from six to three variables, the computational complexity of the state estimation problem is significantly reduced.

As derived in [11] for a general triangle, the distances between each robot pair $d_{i,j}$ can be used to calculate the distance r_i from the centroid C to robot i:

$$r_i = \frac{1}{3}\sqrt{2d_{i,j}^2 + 2d_{i,k}^2 - d_{j,k}^2}. \tag{1}$$

The x-axis of the centroid frame is aligned with the vector to the first robot. The angles of the vectors to the robots are given by

$$\phi_1 = 0 \tag{2}$$

$$\phi_2 = \pm \arccos\left(\frac{r_1^2 + r_2^2 - d_{1,2}^2}{2r_1 r_2}\right) \tag{3}$$

$$\phi_3 = \mp \arccos\left(\frac{r_1^2 + r_3^2 - d_{1,3}^2}{2r_1 r_3}\right). \tag{4}$$

Note that two symmetrical solutions are possible from the geometry. The selected solution is determined by a random binary variable that is set at initialization of the particle.

The goal of the particle filter algorithm is now to estimate the values of the reduced set of state variables x_c, y_c, θ_c, from which the estimated absolute position of robot i can be computed:

$$x_i = x_c + r_i \cos(\theta_c + \phi_i), \tag{5}$$

$$y_i = y_c + r_i \sin(\theta_c + \phi_i). \tag{6}$$

C. Distributed Algorithm Processes

The randomized nature of the particle filter algorithm makes it suitable for a distributed, parallel implementation on multiple robots. Each robot can apply the algorithm to an independent set of particles. However, to make effective use of the best estimates found by each robot, the particles with highest weights must be shared amongst the team. At each iteration then, every robot begins with a set of particles including those with the highest weights selected from all three robots.

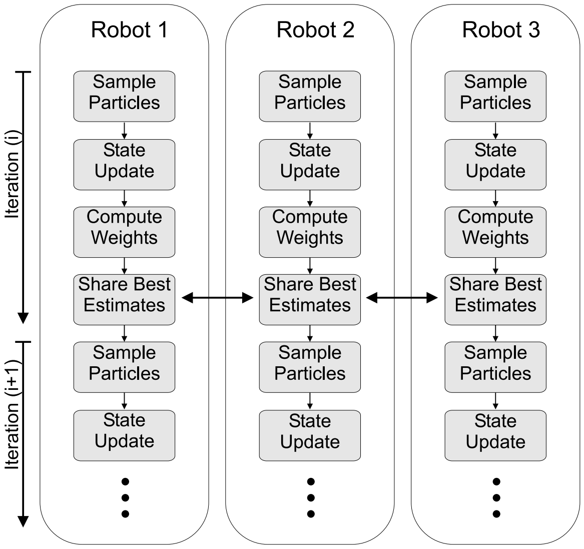

Fig. 2. The sequence of particle filter processes is executed on each robot, and synchronized by the exchange of the best state estimates after each iteration. See text for details of each process.

The sequence of processing and communication involved in the algorithm are shown in Fig. 2. The following sections describe the variations to the three steps in the particle filter algorithm required to integrate the sensor readings from all three robots into a cooperative position estimate of all robot positions.

1) Sampling: At each iteration of the algorithm, a sampling process is required to select a set of particles to propagate forward from the previous iteration. Particles may be selected with probability proportional to their weights. However, a variety of alternative methods can be used to improve the particle selection. As suggested in [3], this method uses a mixture of particles, with a fraction sampled with probability equal to the weights, and a fraction sampled from the best estimates from the most recent sensor measurements. This leads to a denser representation of the belief state in the region of the highest likelihood. As well, as suggested in [12], a small number of particles are added from a uniform distribution of the state space to aid in global localization if the robots become lost after acquiring a confident estimate.

2) State Update: At each iteration of the particle filter algorithm, the state of each particle is updated to reflect the motion of the robots since the last iteration of the algorithm. In a single robot application, the position would typically be

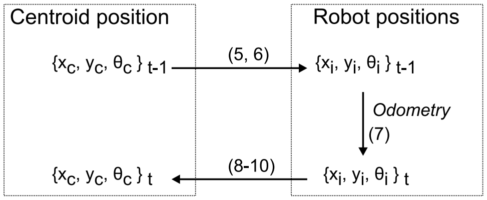

Fig. 3. The state update process computes the new position of the centroid by first transforming the estimate to the robot coordinates, estimating the robot motion, and transforming back to the centroid coordinates.

updated based on the measured odometry and kinematics of the robot. In this method however, the update must reflect the motion of the centroid reference frame C.

As shown in Fig. 3, the motion of C can be computed by first estimating the previous position of the robots, $(x_i, y_i, \theta_i)_{t-1}$, given the previous estimate of the centroid $(x_c, y_c, \theta_c)_{t-1}$ and (5, 6). The updated robot positions, $(x_i, y_i, \theta_i)_t$, are estimated by propagating the previous position through the kinematic equations of motion for the robot with the measured odometry o_{i_t}:

$$\{x_i, y_i, \theta_i\}_t = h\left(\{x_i, y_i, \theta_i\}_{t-1}, o_{i_t}\right). \tag{7}$$

Robot orientations are estimated from compass readings. The updated estimate of C is then computed as the average of the robot coordinates:

$$x_c = \frac{1}{3}\sum_1^3 x_i \tag{8}$$

$$y_c = \frac{1}{3}\sum_1^3 y_i. \tag{9}$$

The state update also requires an update of the orientation of the frame C, which is computed as the angle of the vector from C to the robot at $(x_1,\ y_1)$:

$$\theta_c = \arctan\left(\frac{y_1 - y_c}{x_1 - x_c}\right) \tag{10}$$

3) Weighting: For each particle, a weighting is applied representing the degree of belief in the the position estimate of the particle.

Using the known map of the environment, the expected sensor readings from each robot are predicted, and these are compared to the actual sensor readings. This comparison is used to assign weights to each particle, (i.e. inversely proportional to the difference between actual and expected sensor readings). The particles defining the best estimates (those with the highest weights) are then shared between all the robots. Each robot selects a new set of particles using probabilities proportional to the weights, with a fraction selected around the best estimates, and a fraction randomly selected throughout the configuration space.

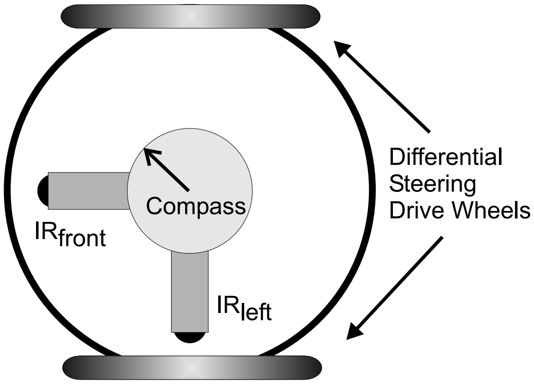

Fig. 4. The simulated robot configuration: The robot is driven with differential steering, and has a compass and two IR sensors oriented at 90° to sense the environment.

This weighting W_p is determined as a function of the error between the predicted IR sensor readings that would be measured from the estimated robot positions and the actual measurements from all three robots:

$$W_p = \frac{1}{\sqrt{\sum_{r=1}^{3}\sum_{s=1}^{2}\left(z_{r,s} - \hat{z}_{r,s}\right)^2}} \tag{11}$$

where $z_{r,s}$ and $\hat{z}_{r,s}$ are the measured and predicted values of sensor s on robot r respectively. The weights are then normalized such that $\sum W_p = 1$.

V. Results

The algorithm was implemented and tested in a simulation of a group of 3 robots, each with two fixed-direction range sensors and a compass, operating in a confined area including obstacles and walls. Each robot begins in a random location in the map. They then create a randomized trajectory by driving forward until they reach an obstacle, after which they turn through a randomly selected rotation. The models of the range sensors, compass, and odometry measurements include injection of Gaussian noise.

During the simulation, each robot performed two localization algorithms. The first was an implementation of the particle filter running in isolation on each robot, using only data from its own sensors. This was used as a baseline for comparison of results. The second algorithm was the distributed method described in this paper, using the combining the sensor readings from all robots and sharing the best estimates.

A. Simulation Environment

The simulation environment is shown in Fig. 5. The small dots indicate the position of particles maintained in the particle filters. The lines radiating from the central cluster represent vectors from the best estimates of the centroid position to the estimated positions of the three robots. The

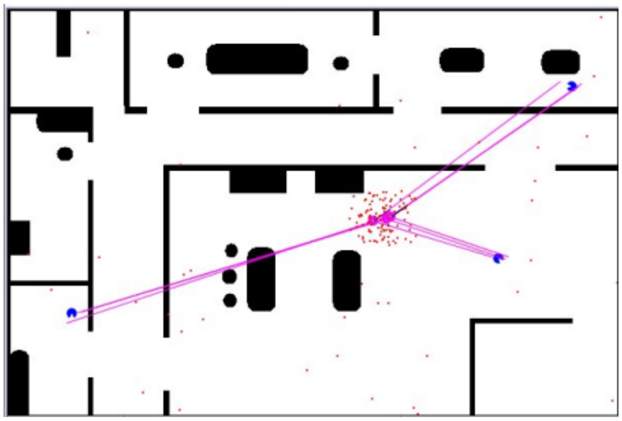

Fig. 5. The simulation environment, a cluttered $6m \times 4m$ area, showing the system state after successful localization.

three circles near the end of the lines indicate the actual positions of the robots. Note that a higher density of particles is maintained around the estimated centroid after completing the localization.

B. *Position Estimation Performance*

In Fig. 6, the average position estimation error is plotted over time for a set of 10 simulations using 800 particles for each robot. On average, after the first 3 seconds of the simulation the distributed localization has converged to the correct estimation of the centroid and robot positions (as shown for example in Fig. 5). This is in contrast to the performance of the individual particle filters operating in isolation, which on average do not converge to the correct solutions for all three robots within the 20 second simulation time.

As shown in Fig. 7, the error between the predicted and actual sensor readings are consistently smaller using the individual particle filters. Operating in isolation, the individual position estimates can lead to many solutions that give sensor readings similar to those from the actual robot position. In contrast, in the distributed method the larger discrepancies between predicted and actual sensor readings reflect the increased constraints on the possible position estimates, imposed by the trilateration calculations. Only those estimates that satisfy the inter-robot distance measurements are considered, leading to less freedom to minimize the sensor prediction error, but a better overall estimate of position.

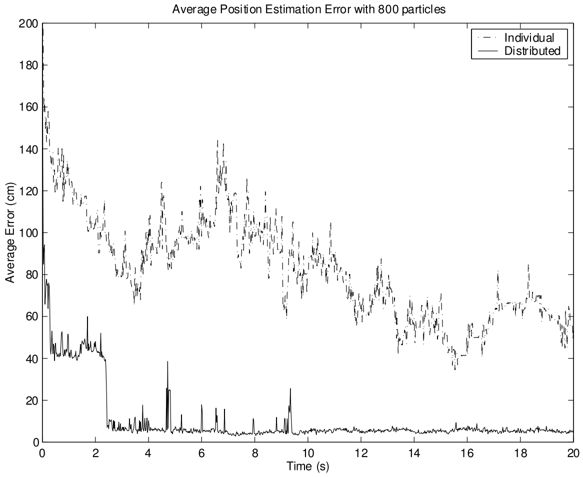

Fig. 6. Comparison of position error over time for the individual particle filters (dotted line) and the cooperative distributed particle filters (solid line). The results are averaged over 10 consecutive simulations of 20 seconds each.

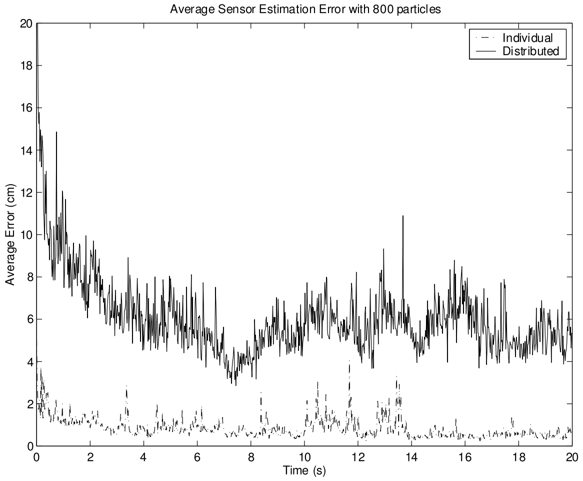

Fig. 7. Comparison of sensor errors over time for the individual particle filters (dotted line) and the cooperative distributed particle filters (solid line).

Fig. 8 shows the performance of the algorithm over a range of sizes for the particle filter, simulating 50 iterations per second on each robot. The vertical scale indicates simulation time, corresponding to the number of iterations of the algorithm with a fixed time step. While the number of iterations required to successfully localize the team decreases as the number of particles increases, the computational cost of each iteration of the algorithm is proportional to the number of particles. This is particularly significant for an algorithm intended for small robots with limited processing capabilities, as the frequency of the algorithm execution will be constrained by the number of particles used. The ideal size of the particle filter for a particular application will be determined based on the computational resources available, the size of the environment, and the required performance of the localization system.

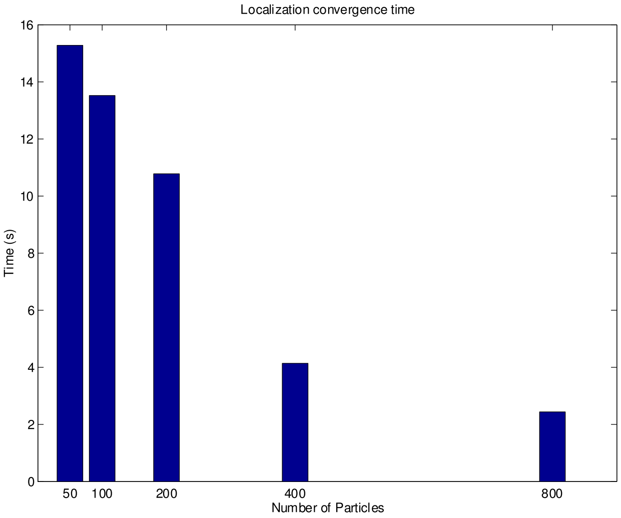

Fig. 8. Comparison of the rates of convergence for varying number of particles.

VI. Discussion

This work presents the initial development of a cooperative method of global localization. A number of features make the method well suited to a distributed implementation:

- Because of the randomized nature of the particle filter approach (in contrast to a Kalman filter based method), each robot can independently generate new estimates, the best of which are then shared throughout the group.
- Little bandwidth is required. Due to the nature of the sensors involved, the raw sensor data can be shared between robots, and the extraction of abstract features is not required.

A number of extensions to the method are apparent for future investigation. The first is a generalization of the method to larger groups of robots, including constraints (such as a line-of-sight requirement) for inter-robot communications. One approach to this generalization is, for each robot, to select a pair of cooperating robots within the communication range of the first. Each robot can then generate an estimate of the positions of itself and two neighbors, using the particle filter method described above. In this case, each particle filter estimates the position of the centroid of a triangle of robots, but the particular three robots (and the estimate of their centroid) may be different for each particle filter. In the *sharing best estimate* process, therefore, the position of the centroid cannot be shared, as it is only relevant to one robot. Instead, the estimated absolute positions of individual robots must be computed and shared. The absolute positions can then be transformed back to the centroid representation required for each local particle filter.

A generalization of the method that does not rely on a compass to estimate orientation would also be beneficial, particular for indoor applications where compass sensors are often subject to interference from structures and electrical equipment. The three robot orientations can be included in the state vector to be estimated, but this increases the configuration space to six dimensions, significantly increasing the computational complexity.

Future work will investigate the implementation of this method on physical robots to validate the simulation results. This will involve an investigation of the required processing power as a function of the number of particles, to determine the optimal parameters for a physical system that is constrained in both sensors and computational resources.

VII. Conclusions

A cooperative, distributed method of global localization for three robots with simple sensors has been presented. Simulation results demonstrate the performance of the algorithm while the number of particles used is varied over a range of relatively small values. Using measures of the distances between pairs of robots, three robots are able to localize themselves in an environment where isolated particle filters on each robot failed to converge.

References

[1] R. A. Brooks, “A robust layered control system for a mobile robot,” *IEEE Trans. Robot. Automat.*, vol. 2, no. 1, 1986.

[2] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla, “Heterogeneous teams of modular robots for mapping and exploration,” *Autonomous Robots - Special Issue on Heterogeneous Multirobot Systems*, 1999.

[3] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, “Particle filters for mobile robot localization,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., 2001.

[4] K. Kato, H. Ishiguro, and M. Barth, “Identifying and localizing robots in a multi-robot system environment,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, 1999.

[5] R. Grabowski and P. Khosla, “Localization techniques for a team of small robots,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2001.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multi-robot localization,” *Autonomous Robots*, vol. 8, no. 3, pp. 325 – 344, 2000.

[7] I. M. Rekleitis, G. Dudek, and E. Milios, “Multi-robot collaboration for robust exploration,” in *IEEE Int. Conf. in Robotics and Automation*, San Francisco, USA, April 2000, pp. 3164–3169.

[8] R. Madhavan, K. Fregene, and L. Parker, “Distributed heterogeneous outdoor multi-robot localization,” in *IEEE Int. Conf. on Robotics and Automation*, May 2002, pp. 374–381.

[9] A. Rynn, W. A. Malik, and S. Lee, “Sensor based localization for multiple mobile robots using virtual links,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.

[10] L. Navarro-Serment, C. Paredis, and P. Khosla, “A beacon system for the localization of distributed robotic teams,” in *Proceedings of the International Conference on Field and Service Robotics (FSR ’99)*, 1999.

[11] J. Casey, *A Sequel to the First Six Books of the Elements of Euclid, Containing an Easy Introduction to Modern Geometry with Numerous Examples*, 5th ed. Hodges, Figgis, & Co., 1888.

[12] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” in *AAAI/IAAI*, 1999.