Heterogeneous Teams of Modular Robots for Mapping and Exploration

Robert Grabowski, Luis E. Navarro-Serment, Christiaan J.J. Paredis, Pradeep K. Khosla

Institute for Complex Engineered Systems,
The Robotics Institute, and
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{grabowski, lenscmu, cjp, pkk}@cs.cmu.edu

Abstract

In this article, we present the design of a team of heterogeneous, centimeter-scale robots that collaborate to map and explore unknown environments. The robots, called Millibots, are configured from modular components that include sonar and IR sensors, camera, communication, computation, and mobility modules. Robots with different configurations use their special capabilities collaboratively to accomplish a given task. For mapping and exploration with multiple robots, it is critical to know the relative positions of each robot with respect to the others. We have developed a novel localization system that uses sonar-based distance measurements to determine the positions of all the robots in the group. With their positions known, we use an occupancy grid Bayesian mapping algorithm to combine the sensor data from multiple robots with different sensing modalities. Finally, we present the results of several mapping experiments conducted by a user-guided team of five robots operating in a room containing multiple obstacles.

1 Introduction

A team of robots has distinct advantages over single robots with respect to sensing as well as actuation [1][19]. When manipulating or carrying large objects, a given load can be distributed over several robots so that each robot can be built much smaller, lighter, and less expensive [24][25]. As for sensing, a team of robots can perceive its environment from multiple disparate viewpoints. In such a system, a task is not completed by a single robot but instead by a team of collaborating robots. Team members may exchange sensor information, help each other to scale obstacles, or collaborate to manipulate heavy objects. A single robot, on the other hand, can only sense its environment from a single viewpoint, even when it is equipped with a large array of different sensing modalities. There are many tasks for which distributed viewpoints are advantageous such as, surveillance, monitoring, demining and plume detection.

Distributed robotic systems require a new design philosophy. Traditional robots are designed with a broad array of capabilities (sensing, actuation, communication, and computation). Often, the designers will even add redundant components to avoid system failure from a single fault. The resulting systems are large, complex, and expensive. For robot teams, the design can be approached from a completely different angle, namely: "Build simple inexpensive robots with limited capabilities that can accomplish the task reliably through cooperation." Each individual robot may not be very capable, but as a team they can still accomplish useful tasks. This results in less expensive robots that are easier to maintain and debug. Moreover, since each robot is expendable, reliability can be obtained in numbers; that is, if a single robot fails, little if any capabilities are lost, and the team can still continue the task with the remaining robots.

Because the size of a robot determines to a large extent its capabilities, we are developing a hierarchical robot team at Carnegie Mellon University. As is shown in Figure 1, the team consists of large All Terrain Vehicles (ATVs) [6][9], medium-sized Tank robots (based on a remote control Tamiya tank model) [5], a set of Pioneer robots and centimeter scale Millibots (7×7×7cm). The ATVs have a range of up to 100 miles, are completely autonomous and carry extensive computational power. They are capable of transporting and deploying groups of smaller robots to distant areas of interest while providing higher-level computational support of the extended team. The Pioneer robots are platforms for the development of "Port-Based Adaptive Agents" [8] that will allow the team to dynamically exchange algorithms and state information while on-line. The Tank robots are medium-sized, autonomous robots complete with infrared and sonar arrays, a swivel-mounted camera and an on-board 486 computer. Each tank robot is capable of individual missions or can serve as the leader and coordinator for a team of smaller, centimeter-scale robots called Millibots. These small and lightweight robots can be easily carried by their larger counterparts higher-up in the robot hierarchy. They can maneuver through small openings and into tight corners to observe areas that are not accessible to the larger robots. Being small, they are also

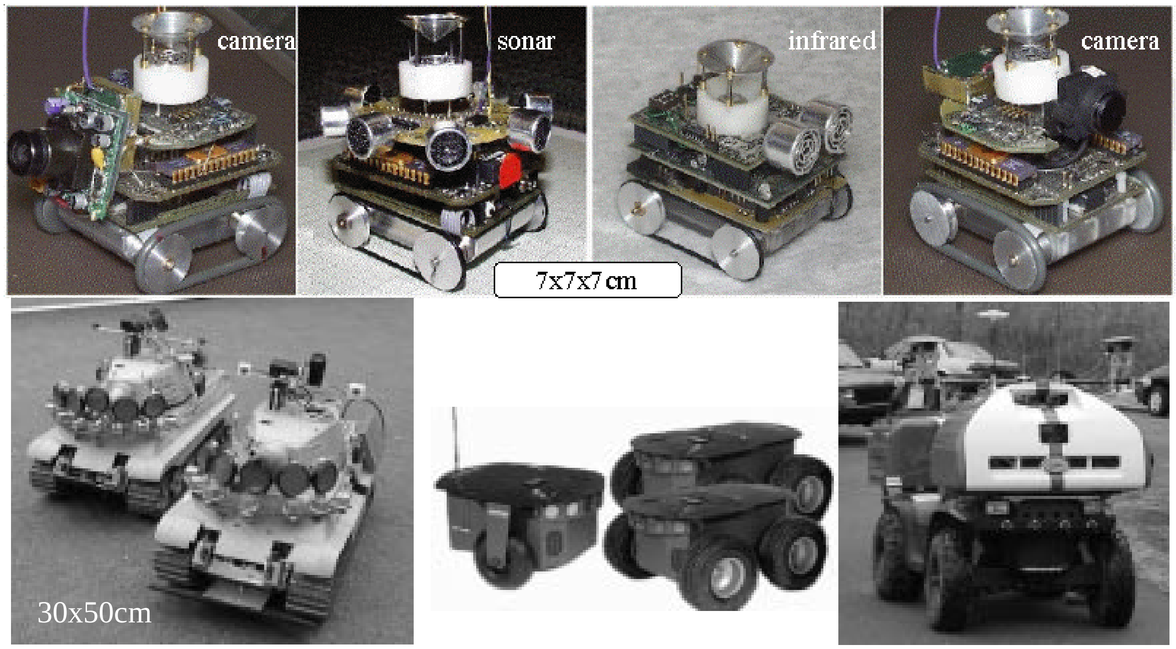

Figure 1: A hierarchical team of robots consisting of Millibots (top), Tanks, Pioneers and ATVs

less noticeable allowing for more covert operations in hostile territory. Such a hierarchical organization allows us to combine the autonomy and computation power of the large ATVs, the maneuverability of the tanks and the distributed sensing capabilities of a large number of covertly operating Millibots. In this paper, we will focus primarily on the design and operation of the smallest units of our team: the Millibots.

2 The Millibots

The primary factor that determines what a robot can do and where it can go is size. The most obvious advantage of a smaller robot is that it can access spaces restricted to its larger counterparts. Small robots can crawl through pipes, inspect collapsed buildings, or hide in small inconspicuous spaces. For surveillance and exploration tasks, this increased accessibility dramatically impacts the overall functionality of the robot. However, with small size comes the disadvantages of limited mobility range, limited energy availability, and possibly reduced sensing, communication and computation ability due to size and power constraints.

2.1 Small Robots

Several efforts for building small mobile robots have been reported in the literature [13][20][21][29]. Although these robots are feats of technological ingenuity, they tend to lack the capabilities necessary for performing tasks going beyond the complexity of follow the leader, move towards the light source, etc. Often a small robot must sacrifice one feature to achieve another.

One exception is the Khepera robots that have achieved both small size and computing complexity [21]. Khepera robots are 5cm in diameter and are capable of significant on-board processing. Khepera robots are modular and support the addition of sensor and processing modules. They are designed to work alone or communicate and act with other robots. However, the Khepera robot lacks a significant feature that would allow it to operate in an unknown environment, combine sensor information and act as a central, cohesive unit: *self-localization*. Khepera must either rely on a fixed position global sensor (overhead camera) or internal dead-reckoning. Both methods make them ineffective as a deployable set of robots. As we will discuss later, the Millibots have developed a set of sensor modules that allows a group of Millibots to self-localize and move as a coordinated entity while maintaining relative position information about the group.

Another potential limitation of the Khepera robots is their choice of propulsion. Khepera robots achieve mobility from a pair of centimeter sized wheels housed in the center of the robot. This form of mobility is good for flat surfaces but restricts the robot’s clearance to about 3 mm—significantly limiting the environments in which the Khepera robots can operate. On the other hand, Millibots can be configured with various mobility platforms allowing them to operate in different environments. For example, when configured with a thick rubber tread design, Millibots have a clearance of about 15mm allowing them to climb inclines and small obstacles. The same Millibot can be equipped with a wheel and caster design when operating on a flat hard surface.

A set of robots, called Ants [20], is also on the same scale as the Millibots. These robots are also designed to be used in groups or teams. However, since these small robots were developed primarily to explore reactive social behaviors they are very limited in sensing. They do not support a real-time communication link nor are they equipped to exchange sensor information necessary to produce maps or models of the environment. Rather they are designed to convey simple messages such as “have food” or “it” via a short-range infrared transmitter, sense objects with a simple touch switch and sense orientation with a simple light detector. Millibots, on the other hand, are equipped with various sensor arrays such as a multi-element sonar and on-board video that provide more detailed information. These sensors are capable of identifying objects and building maps of the environment.

To achieve their scale, the Ants were built with a fixed architecture. Propulsion, sensing and processing are combined to optimize size constraints such that the addition of any new functionality would require a complete redesign. The Ants suffer from the inability to localize as well. They rely on the presence of a strong light source for orientation and encoders for dead-reckoning. Even if they were able to communicate more detailed information between them, without a means for determining position, they would have little context in which to evaluate the data.

Examples of small-scale *cooperating* robots are the FIRA and RobotCup competitions (Federation of International Robot-soccer Association) [29]. Each team of soccer robots consists of a group of five robots that are limited in size to 7.5×7.5×7.5 cm. These robots coordinate to perform complex actions like passing a ball and defending a goal against a coordinated attack. Like Millibots, the team of soccer robots acts as a set of distributed mobility platforms tasked by a central controller. However, most soccer robot teams are extremely limited in their sensing capabilities. Position sensing for the soccer robots is accomplished via a global camera positioned above the playing field. Most teams have little or no sensors on the robots themselves. Without the external camera, the robots are blind and unable to respond to real-world events.

The examples discussed above illustrate some of the limitations imposed by small scale. Small robots must sacrifice mobility, sensing and power to achieve their desired scale. To remain effective, they must adopt new techniques to overcome these limitations.

3 Specialization and Collaboration

Our approach to overcoming the disadvantages imposed by small robots is based on *specialization* and *collaboration* (section 4). Specialization is achieved by exploiting the nature of a heterogeneous team. Instead of equipping every robot with every sensor, computation, and communication capability, we are building robots that are each specialized for a particular aspect of the task. In one type of scenario, the robot team may be composed of robots with various range and position sensors but only limited computation capabilities. In this case, the robots act as distributed sensor platforms remotely controlled by a team leader who performs the high-level planning. In another task, the same group of Millibots may be equipped with more computation. Data can be collected and stored locally until it is ready to be retrieved. Some missions may dictate a collection of robots of differing strengths. The choice of platforms and how the platforms are used depends only on the task.

By omitting the capabilities that are unnecessary for a particular scenario, power, volume, and weight of the robot can significantly be reduced. However, specialization has the disadvantage that many different robots need to be available to address the specific requirements of a given task.

3.1 Modular Architecture

To achieve this level of specialization without the need for a large repository of robots, we have chosen to develop the Millibots in a modular fashion. A Millibot is constructed by assembling a set of sub-systems ranging from computation to communications to sensors. Even the mobility platform is modular and can be selected based on the terrain of the mission. To support modularity, each of the subsystems has been implemented as a self-contained module complete with processor and interface circuitry.

Specialization through modularity also allows the Millibots to optimize resources. By constructing a robot with only mission specific modules, the size and cost of the robot can be kept to a minimum. Reduction of unnecessary payload means the robot will have less weight and consume less power. Furthermore, some robots require less computational complexity allowing them to be equipped with smaller processors that in turn consume less power. Smaller and cheaper means that robots can be built and deployed in large numbers to achieve dense sensing coverage, team level adaptability, and fault tolerance.

As shown in Figure 2, each Millibot is composed of a main processor with optional communication and sensor modules housed on a mobility platform. The modules interface with each other through a standardized bus for power and inter-module communication. Each module contains its own microprocessor that implements the inter-module communication protocol and performs low-level signal processing functions for sensor and actuator control.

Coordination between modules is accomplished in two ways. Data and timing signals can be shared either through a set of dedicated slots or via a common I^2C bus. The dedicated slots are fixed connections that can support up to six sensor or actuator modules. The choice of module and slot is determined by the operator and fixed in software. Information is passed back and forth via these slots in the form of serial communications. Dedicated slots are reserved for modules that may need

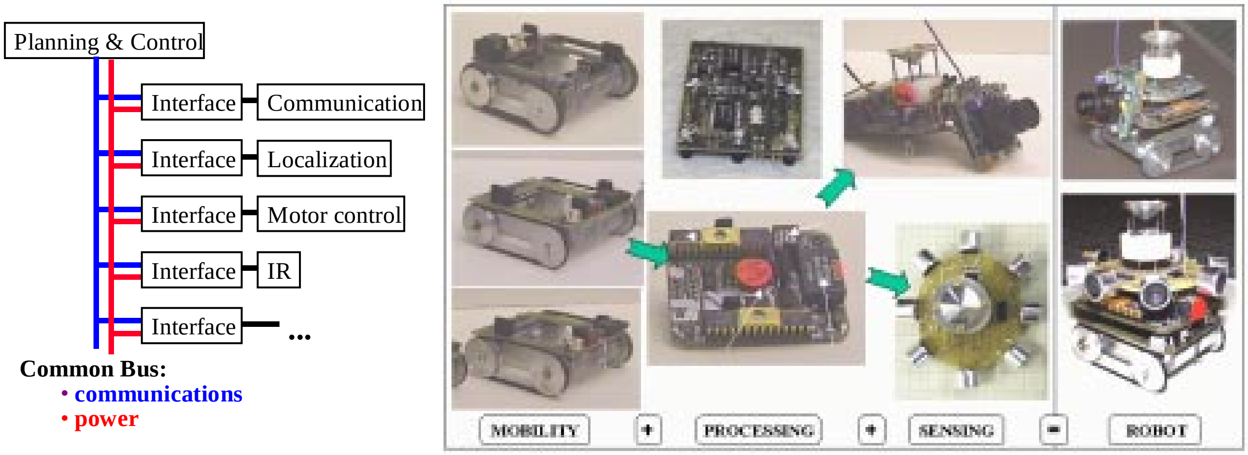

Figure 2: The Millibot's architecture and subsystems.

additional data lines or cannot handle the computational complexity of I^2C.

An I^2C bus [17] provides a second option for connection of modules. I^2C is a bus design and communications protocol that allows multiple modules to be connected to a common two-wire bus. One wire provides a high speed, synchronous clock while the other provides a two-way data communication line. Target modules are distinguished from one another by pre-appending each message on the data line with an address header. Only the module that matches the address acts on that message This interface is less restrictive than the dedicated slot method because it allows many modules to be connected to the same processor without having to designate on which pins the modules reside. The choice of which type of connection is determined by the operator and easily configured in software.

Any module that does not violate the size and power constraints of the Millibots and provides a serial interface for data exchange can utilized by the Millibot. Sensor modules of this type include ranging sensors, proximity detectors, chemical sensors, magnetic field detectors, and radiation monitors.

3.2 The Millibot subsystems

Currently the Millibots can be composed from a suite of seven subsystems: the main processor module, a communication module, an IR obstacle detection module, two types of sonar modules, a motor control module, and a localization module.

Communications is essential in a coordinated team. Without explicit communications, a robot can only interact with team members using its sensors (e.g. vision-based “follow the leader” behavior) [4] [19]. However, collaborative mapping and exploration requires the exchange of detailed and abstract information that cannot be easily conveyed implicitly. To provide two-way communications within the group, each Millibot is equipped with a radio frequency transmitter and receiver. These units can exchange data at 4800 bps at a distance of up to 100 meters. The choice of units is based primarily on size and power considerations Units with higher data transfer rates exist but at the prohibitive cost of size. This tradeoff between size and functionality is a common theme in constructing small robots. We expect that smaller more powerful transmission units will become commercially available in the future as miniaturization in solid state progresses.

To perceive the world, a robot must have sensors. There are currently three sensor modules available to each Millibot. The first two are a set of ultrasonic sonar modules that provide focused range information about obstacles. One sonar module type provides short-range distance information for obstacles between 0 and 0.5m. The module consists of eight sonars arranged in a ring around the center of the robot. Though limited in range, this module provides detailed information about the area directly surrounding the robot. A short-range sensor module such as this is ideal for Millibots that have to work in tight or cluttered areas. For robots at this scale, the ability to measure extremely short range is essential.

The second sonar module type provides long-range information for obstacles between 0.15m and 1.8m. Long-range sensors are more effective in environments that are more open such as hallways or open office spaces. Because of the construction of this type of detector, a Millibot can currently support only one sonar pair per robot. This unit can return long-range distance information but in a tightly focused cone of only 40 degrees. To obtain more complete information about the environment, this type of detector would have to rely on movement of the robot to increase coverage. For some tasks, it may be desirable to have both short and long-range sonar sensing available. This can still be achieved by equipping some Millibots on the team with short-range modules and some with long-range sonar modules.

A potential complication with any ultrasonic based sensor is the probability of interference with similar modules on other robots. Most sonar elements operate at a fixed frequency determined by their mechanical construction. Therefore, two robots using ultrasonic sensors in the same area will most likely cause interference for the other. To overcome this problem while still providing continuous rudimentary obstacle detection, a Millibot may opt to carry an infrared proximity module. The proximity module provides an array of five tunable, infrared emitter-detector pairs that trigger when an obstacle intrudes within its cone of emission. The proximity elements can be calibrated to provide readings of up to 0.25m. Although the proximity detectors cannot be reliably used for range determination of objects, they can be used very effectively in conjunction with a sonar detector module. The proximity module can be continuously sensing an area around the robot. Upon detection of an object by the proximity detector, the robot can coordinate with the team to acquire better range information through its sonars.

Except under the most controlled conditions, the sensors discussed so far cannot provide enough detail to resolve many of the problems facing a real robot. Real situations are fraught with anomalies. A method is needed to provide high bandwidth information during a mission for analysis by a higher level process or operator. To provide this service, Millibots can be equipped with a camera module. The camera module provides an external mini camera, video transmitter and power circuitry. Currently because of the limited processing capabilities of the Millibot, video signals cannot be processed on-board. A small video transmitter is included with the module to transmit the raw video signal to an external processor or remote viewing station. The camera module includes circuitry that allows the camera and its transmitter to be switched on and off via control signals from the Millibot. Control of the camera aids in effective power management. The camera need only be powered when an image is desired. The ability to remotely power down an individual transmitter also allows multiple robots to carry similar camera modules while using the same transmitter frequency. Interference is prevented by powering only one transmitter at a time. Additionally resources are minimized since only one receiving station and associated monitoring device is needed per Millibot group. However, though the camera module provides valuable visual information, it operates on the threshold of the Millibot’s power budget. The current camera dissipates about 1.5 Watts of power during operation. Due to the limited size of the battery, this type of sensor cannot be used continuously like other sensor packages.

Not all scenarios will support robots designed with the same modes of propulsion. For example, a small robot equipped with a set of thin rubber tracks will perform well on a flat, slippery surface, such as a floor or table but performs poorly on a shag rug. Conversely, the same robot may outperform a robot equipped with wheels in another scenario. In the Millibot group, modularity has been extended to the mobility platforms as well. A mobility platform is selected for a particular Millibot and the main processor and its set of support sensors is added to make it a robot. In most cases, the mobility platforms will utilize a similar set of dc motors. Therefore, the same motor control module can be selected and only the software needs to be changed. For platforms that differ, they need only to include their own motor control module and conform to the software interface. Currently the Millibots have implemented three sets of platforms each utilizing skid steering. Some Millibots are equipped with a plastic chain design which is ideal for rough surfaces like rugs while others are equipped with rubber tread designs of differing widths which allows them to crawl on smooth inclined surfaces.

4 Collaboration

In addition to specialization, Millibots use collaboration to overcome the limitations imposed by small scale. By nature, Millibots are small, mobile robots with limited capabilities. Yet, by collaborating with each other as a team, they are able to overcome their individual limitations and accomplish important tasks ranging from localization to surveillance, mapping, and exploration.

When distributed robotic applications require robots to share sensor information (e.g. mapping, surveillance, etc.) it is critical to know the position and orientation of the robots with respect to each other. Without this knowledge, it becomes impossible to interpret the sensor data in a global frame of reference and integrate it with data obtained by other robots. Millibots exploit collaboration to obtain relative position and orientation of the team with respect to each other, even as the team moves. Without an external means of localization, this knowledge is essential for the team to move to predetermined locations, avoid known obstacles, or reposition themselves for maximum sensor efficiency.

Conventional localization systems do not offer a viable solution for Millibots. Dead reckoning, a common localization method, generally suffers from accuracy problems due to integration errors and wheel slippage [3]. This is even more pronounced for systems that rely on skid steering for which track slippage is inherent to the steering mechanism. Camera-based localization, such as

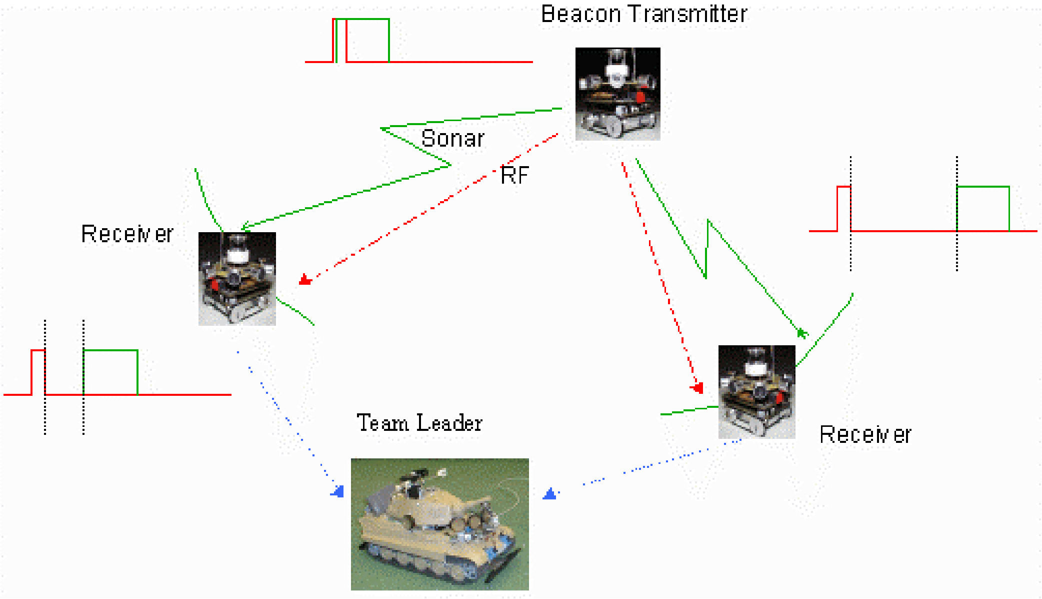

Figure 3 : Ultrasonic distance measurement.

those used by the soccer robots, is not feasible in many of the environments in which small robots can be exploited. On the other hand, larger robotic systems often rely on Global Positioning Systems (GPS) and compass for determining their position and orientation in a global frame of reference [12]. However, due to its size, limited accuracy, and satellite visibility requirements, GPS is not appropriate for use in small robots that operate mostly indoors. Conversely, systems that are based on landmark recognition [2][15] or map-based positioning [27] require excessive local computational power and sensing accuracy to be implemented on Millibots.

To overcome the problems encountered in the implementation of existing localization methods for a team of Millibots, we have developed a novel method that combines aspects of GPS, land-mark based localization, and dead reckoning [23]. This method uses synchronized, ultrasound pulses to measure the distances between each robot on a team and then determines the relative positions of the robots through trilateration. Similar systems have been developed [14]. However, they are both too large and too expensive for operation on Millibots. Moreover, the system described in this article is more flexible because it does not require any fixed beacons with known positions, which is an important relaxation of the requirements when mapping and exploring unknown environments.

4.1 Collaborative Localization

The Millibot localization system is based on the trilateration [3], i.e., determination of the position based on distance measurements to known landmarks or beacons [16] [18]. GPS is an example of a trilateration system; the position of a GPS unit on earth is calculated from distance measurements to satellites in space. Similarly, the Millibot localization system determines the position of each robot based on distance measurements to stationary robots with known positions.

To derive team positions, each Millibot is equipped with a localization module that utilizes ultrasound and radio pulses to measure the distances between it and other robots. Each localization module is designed to act as both emitter and receiver. Periodically, each module emits a series of localization pulses that emit radially away from the robot. To synchronize the timing between robots without having to use accurate timers, each module actually emits both a radio frequency (RF) pulse and a series of ultrasonic pulses. As is illustrated in Figure 3, The RF pulse, traveling at the speed of light (3×10^8 m/s), arrives at each receiver almost instantaneously. The ultrasonic

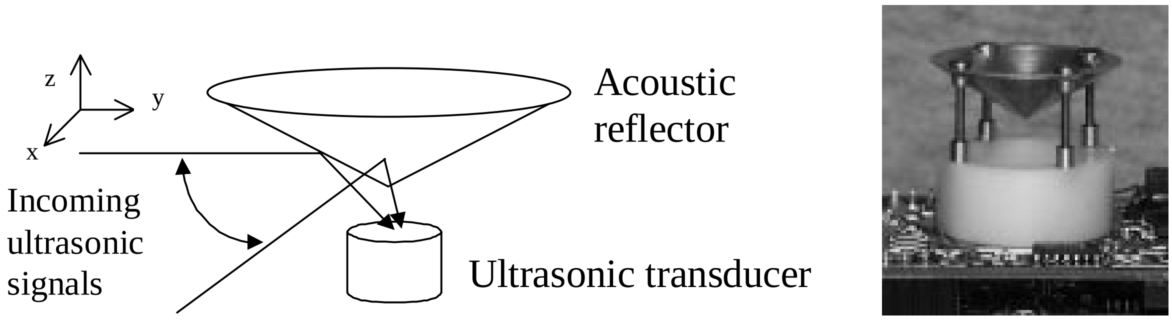

Figure 4 : The acoustic reflector

pulses, traveling only at 343 m/s (assuming 20°C air temperature), arrive at each receiver delayed by a time proportional to its distance to the emitter. The timing between each robot pair is stored locally by the module until it can be reported to the team leader.

Following each sequence, the team leader collects the timing information between each pair. This information is used to determine the actual position of every Millibot using a maximum likelihood estimator. In the future, we plan to calculate the Millibot positions on the local processor of each Millibot. However, the current processor does not have the necessary computation power to perform these floating-point computations.

To produce and detect beacon signals, each Millibot is equipped with a modified, low-cost ultrasonic transducer. This transducer can function either as a receiver or as an emitter. For localization to be effective, it is important that the sensor is able to detect signals coming from any direction around the Millibot. As illustrated in Figure 4, an ultrasonic transducer is positioned to face straight up and all incoming and outgoing sound waves are reflected by the aluminum cone. The result is a detector with a coverage of 360 degrees in the horizontal plane. The ultrasonic transducer with reflector is about 2.5cm tall. It can measure distances up to 3m with a resolution of 8mm while consuming only 25mW. The construction and design of this detector was paramount in achieving a localization system at this scale.

4.2 The Localization Algorithm

To determine the position and orientation of the robots relative to each other, we use a maximum likelihood estimator. If all the distance measurements were perfectly accurate, we could use a simple geometric trilateration algorithm to determine the position of the robots relative to each other. However, measurements are noisy and sometimes missing. As a result, the set of equations resulting from a purely geometric approach is over-constrained and does not always yield a solution. Instead, we use a maximum likelihood estimator that determines the most likely position and orientation of all the robots, given their previous positions and orientations, their movements, and the sonar-based distance measurements.

Assume that we know the position an orientation, (x^0, y^0, φ^0), of all the robots at time t_0. The question is: how do we determine the position, (x^1, y^1, φ^1), of the robots at time t_1, after they have moved? We can estimate the new positions based on the following information:

- **Dead reckoning**: Since all the Millibots are equipped with encoders, their position at time t_1 can be estimated by integrating the encoder signals. This can be further simplified in our case, because the Millibots always move according to "vector commands" (i.e., rotation in place over an angle α, followed by a forward straight-line motion over a distance, d). A stiff controller guarantees that the commanded motion, (α, d), is realized, eliminating the need to query the robot after the motion is completed. In addition to the parameters α and d, we assume that the vector command is characterized by the angle β. As is illustrated in Figure 5, β is the angle over which the robot rotates while moving forward. This unplanned rotation is due to wheel slippage and calibration errors in the controller. There is a one-to-

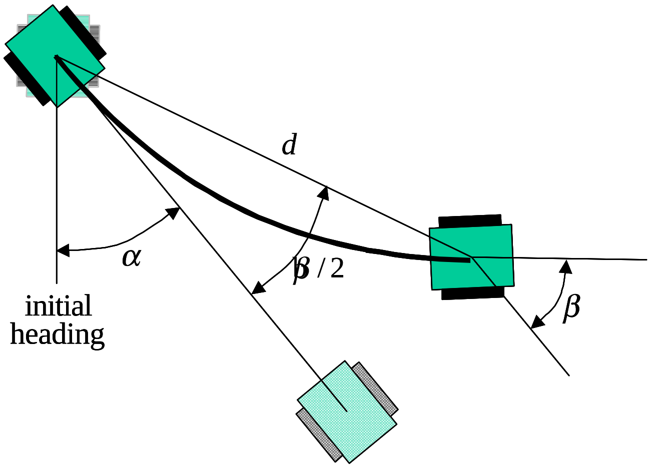

Figure 5: Vector command, (α, d), with the deviation from straight-line motion.

one mapping between the incremental motion of the robot, from (x^0, y^0, φ^0) to (x^1, y^1, φ^1), and the parameters (α, β, d):

$$
\begin{aligned}
d &= \sqrt{\Delta x^2 + \Delta y^2} \\
\alpha &= 2\,\mathrm{atan2}(\Delta y, \Delta x) - \Delta\varphi \\
\beta &= 2\Delta\varphi - 2\,\mathrm{atan2}(\Delta y, \Delta x)
\end{aligned}
\qquad (1)
$$

- **Distance measurements**: After all the robots have come to a halt, each robot that moved pings its localization beacon to determine its distance to all the other robots. The resulting distance measurements provide accurate data that allow us to overcome the drift typically encountered in localization algorithms based on dead reckoning alone.

We have carefully calibrated the motion controller and localization beacon so that, in addition to the nominal measurement, we have an estimate of the corresponding standard deviation. As illustrated in [23], the distribution of the localization data closely resembles a Normal distribution. Assuming that both the dead reckoning data and the distance measurements are normally distributed, we can compute the likelihood of a particular set of measurements occurring for a given robot position:

- **Dead reckoning**: The likelihood that a robot moved over an angle, $\alpha \pm \sigma_\alpha$, and a distance, $d \pm \sigma_d$, given its initial position (x^0, y^0, φ^0) and final position (x^1, y^1, φ^1) is:

$$
P(\alpha, d \,|\, x_i^1, y_i^1, x_j^1, y_j^1) = N(\frac{\alpha - \hat{\alpha}}{\sigma_\alpha}) N(\frac{\beta - \hat{\beta}}{\sigma_\beta}) N(\frac{d - \hat{d}}{\sigma_d}) \qquad (2)
$$

- **Distance measurements:** The likelihood that the measured distance between two robots, i and j, is equal to D_{ij} is:

$$P(D_{ij} \left| x_i^1, y_i^1, x_j^1, y_j^1 \right.) = N(\frac{D_{ij} - \sqrt{(x_i^1 - x_j^1)^2 + (y_i^1 - y_j^1)^2}}{\sigma_D}) \tag{3}$$

The total conditional likelihood function $P_{tot}(\alpha_i, d_i, \ldots, D_{ij}, \ldots \left| x_1^0, y_1^0, \varphi_1^0, \ldots, x_n^0, y_n^0, \varphi_n^0, x_1^1, y_1^1, \varphi_1^1, \ldots, x_n^1, y_n^1, \varphi_n^1 \right.)$ is the product of all the conditional likelihoods introduced above. The most likely robot positions are found by maximizing P_{tot} with respect to the new robot positions $(x_1^1, y_1^1, \varphi_1^1, \ldots, x_n^1, y_n^1, \varphi_n^1)$.

The maximum likelihood estimator requires that the initial positions of the robots are known with respect to one another. This requires a slightly modified approach at start up. After collecting distance measurements between all possible robot pairs, a conditional probability density function is defined which only consists of distance measurement terms. In addition, one arbitrary robot is assigned the position (0,0) and a second robot is assigned a position on the X-axis. This defines a frame of reference in which the position of all other robots is determined by maximizing the conditional probability density function. However, based on distance measurements alone, there remains an ambiguity about the sign of the Y-coordinates of each robot. To resolve this ambiguity, the team leader commands one robot to follow a short L-shaped trajectory and recomputes its position. If the robot turned to the left, but the assigned coordinate system indicates a right turn, the signs of the Y-coordinates of all robots are reversed.

4.3 Implementation Issues

The optimization of the conditional probability density function can be formulated as a weighted nonlinear least-squares problem, which we solve using the BFGS nonlinear optimization algorithm [11]. The dead reckoning data provides a good starting point, so that only a few optimization iterations are necessary to reach the optimum. During the initialization stage of the robot team, when no prior information about the robot positions is available, the BFGS algorithm may get stuck in a local minimum. Based on experimentation, we have found that taking the best-out-of-five randomly initialized runs never fails to find the global optimum.

To obtain good results with the above algorithm, it is very important to filter the raw measurement data. Even though the sensors are very accurate and reliable, it is possible that they have returned false measurements. This occurs for instances where the direct path between two robots is obstructed by an obstacle or another robot. The beacon sensors will always return the time corresponding to the first incoming ultra-sound pulse. In this case, the first pulse is the result of some multi-path rather than the direct-path pulse. As a result, the measured distance can be significantly larger than the actual distance. A similar error occurs when there is a multi-path pulse that destructively interferes with the direct pulse. In this case, the ultrasonic pulse is not detected at all.

It should be noted that the difference between a good and a bad distance measurement cannot be recognized based on the measurement data alone. Indeed, multiple measurements will all result in the same (possibly erroneous) reading. Erroneous readings can still be rejected, however, based on dead reckoning information. Even though dead reckoning is unreliable when integrated over a long time, for a single robot action, it can provide a reasonable estimate of the robot's position. By comparing the encoder distance measurements with the distances computed for the estimated positions, it is possible to reliably reject erroneous measurements due to multi-path.

Furthermore, the accuracy of the algorithm was improved significantly by using more than the minimally required three robot beacons. In our experiments, we used a team of five robots in which, at any time, four served as beacons. The extra distance measurements improve the accuracy of the position estimates, especially, when the direct path to one or more of the robot beacons is obstructed by obstacles. Further accuracy improvements were obtained by pinging each beacon multiple times. Median and mean filtering were then used to significantly reduce the standard deviation of the distance measurement, resulting in a more accurate position estimate.

5 Mapping and Exploration

The primary utility of the Millibots is exploration and mapping. The team coordinates movements and collection of sensor data to produce maps and explore unknown spaces. The ability for a single robot to map any significant area is difficult, especially for robots at this scale. Even with its long-range sonars, the Millibot is limited to a detection range of only about 50 centimeters. However, a group of Millibots can be equipped with similar sensors to cover more area in less time than a single robot. During operation, each robot collects information locally about its surroundings. This data is transmitted to the team leader where it is used to build a local map centric to that robot. The team leader (or human operator) can utilize the robot's local map information to direct the Millibot around obstacles, investigate anomalies or generate new paths.

For missions in an unknown environment, mapping, exploration, and movement are a coordinated effort. Robots move to collect new information about the environment and build composite maps. In turn, the maps provide clues about the most viable areas of exploration that will further increase the knowledge about the environment. In addition, evaluation of the map aids in path planning for the movement and positioning of the team during exploration. Planning is necessary to establish

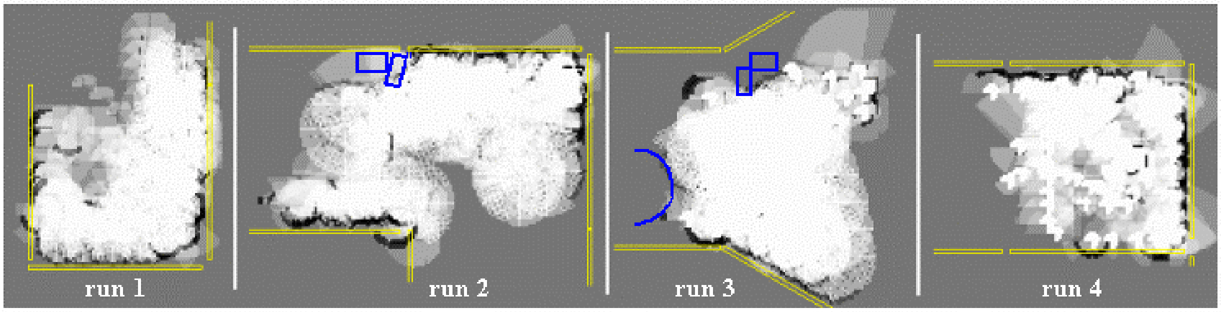

Figure 6 : Mapping Experiments

good sensor coverage while maintaining localization.

This last part is essential in that the team relies on line-of-sight beaconing for maintaining position. If an individual robot moves out of sight of too many robots, the team will lose positioning of that robot and possibly corrupting the map. Movement around obstacles in the field requires coordination at the team level.

More importantly for the group, the team leader can merge the information from several local maps into a single global map to provide a more comprehensive view of the environment to the user. To produce maps of the environment, one of our methods is to build an occupancy grid with a Bayesian update rule. This method allows the combination of sensor readings from different robots and different time instances [10][22][26][28]. In an occupancy grid, the environment is divided into homogeneous cells. For each cell, a probability of occupancy is stored. An occupancy value of zero corresponds to a free cell, a value of one corresponds to a cell occupied by an obstacle. Initially, nothing is known about the environment and all the cells are assigned a value of 0.5 (equally likely to be occupied or free).

The mapping algorithm uses a Bayesian update rule [22]:

$$\mathrm{Occup}(c \mid S^{(1)},\cdots,S^{(T)}) = 1 - \left(1 + \frac{\mathrm{Occup}(c \mid S^{(T)})}{1-\mathrm{Occup}(c \mid S^{(T)})}\frac{\mathrm{Occup}(c \mid S^{(1)},\cdots,S^{(T-1)})}{1\text{-}\mathrm{Occup}(c \mid S^{(1)},\cdots,S^{(T-1)})}\right)^{-1} \tag{4}$$

Equation (4) updates the occupancy probability for cell *c*, $\mathrm{Occup}(c \mid S^{(1)},\cdots,S^{(T)})$, based on the current sensor reading, $\mathrm{Occup}(c \mid S^{(T)})$, and the *a priori* probability, $\mathrm{Occup}(c \mid S^{(1)},\cdots,S^{(T-1)})$. Any sensor that can convert its data into a probability that a particular cell is occupied can be merged into the same map. This means that data generated by a short-range proximity detector can be merged with data from a sonar range module or even a camera.

The tests were conducted with a team of five robots and a single team leader. For this experiment, three of the Millibots were equipped with an eight-element sonar array and a localization module. The remaining two were equipped with cameras that were used to provide fault recovery and obstacle identification. The map from each run was merged to generate a composite map of the room. In these experiments, the robots collaborated to determine their position and to combine their sensor data into a global map. However, the planning of individual robot motions was performed by the human operator.

6 Results

We have conducted a series of experiments to test the effectiveness of a team of Millibots to explore and map a given area (Figure 6). Each series of runs was designed to slowly increase the complexity of operation to expose the strengths and weaknesses of the team. The task in each mission was to explore and map as much area as possible before the team failed. Possible failures included: loss of localization, loss of battery power or loss of communications. As an added utility, each series of runs was performed in different sections of the hallway. By merging maps from these individual runs, we were able to construct a composite map that represented the entire area

(Figure 7). The operator was positioned away from the experiments and could only receive information about the environment via the team by viewing video from the robot cameras and observing sensor information.

For each experiment, the team was composed of five Millibots. Three of the Millibots were equipped with sonar arrays for collecting map information while the remaining two were equipped with camera modules to aid in obstacle identification and provide a level of fault tolerance. In addition, each robot also housed a localization module that allowed them all to participate in localization. This heterogeneous makeup mixed the utility of sonar for extended range mapping with the utility of high bandwidth cameras. A homogenous team of either type would have proved ill prepared for the tasks. A team of camera robots could not produce effective range maps while a team of sonar robots would be unable to detect certain types of obstacles.

Five robots were chosen to provide a degree of fault tolerance. Under a few initialization assumptions, three robots are sufficient to achieve and maintain localization. With four robots, one

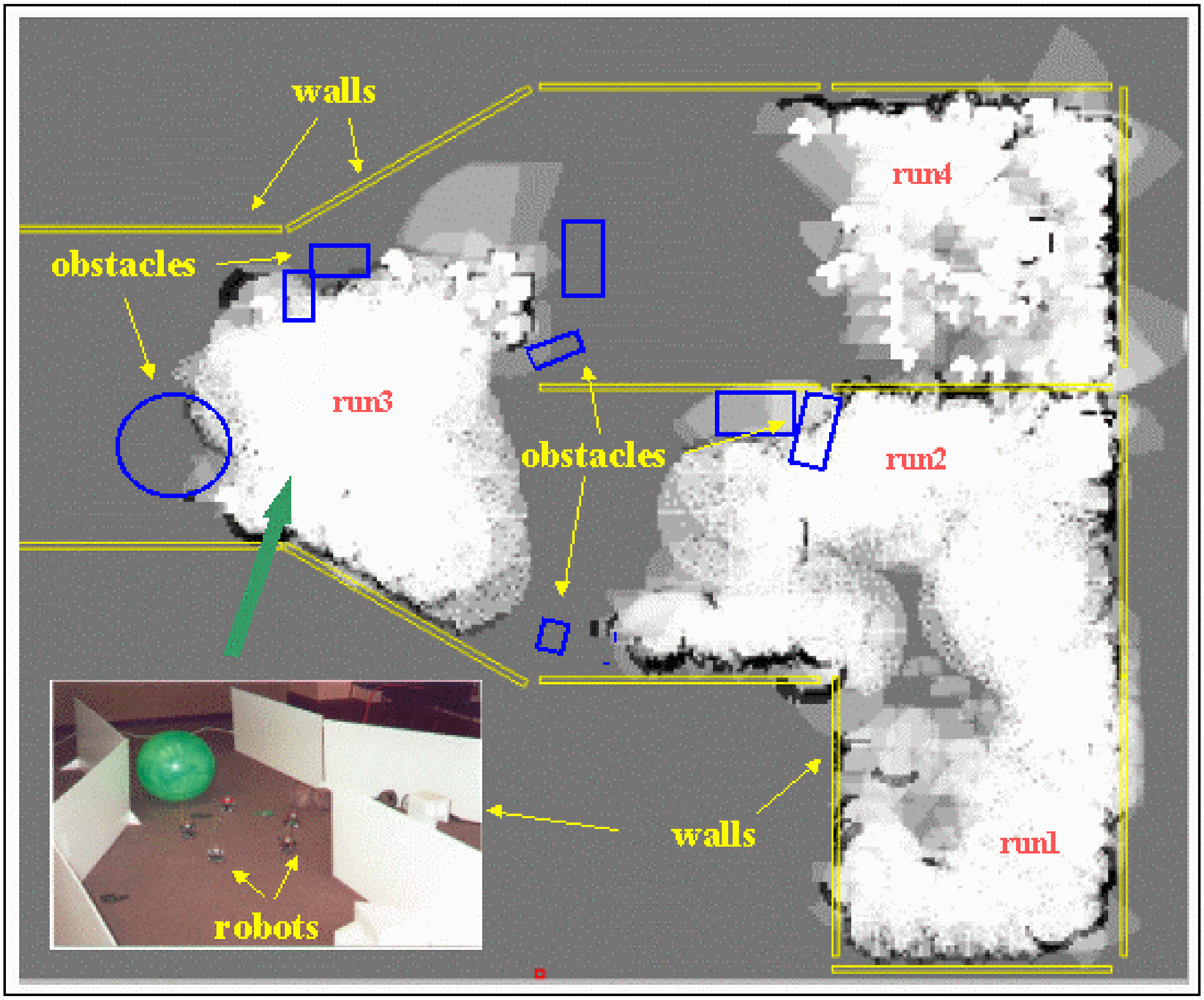

Figure 7: Merging Maps

moving robot can be unambiguously localized with respect to the three stationary robots. A fifth robot provides a measure of fault tolerance such that the loss of a single robot does not compromise the ability of the remaining robots to maintain knowledge of their positions.

Tasking of the team was accomplished through a distributed control system called CyberRAVE [7]. CyberRAVE is a client-server architecture that allows multiple, heterogeneous teams to coordinate operation and share data via a central control server and a set of distributed Graphical User Interfaces (GUI). Through the GUI, the operator is able to direct the robots by setting goals, querying maps and viewing live sensor data. The operator can task individual robots or the team as a whole. The operator's control ranges from tasking of high-level operations such as 'explore area' or 'track target' to low level actions such as 'go-to-goal' or 'ping sonars'.

At this stage of development, much of the high level skills are provided by the operator in response to feedback from the robot teams via the interfaces. Based on the information presented by the GUI, the operator tasks the team. For instance, he decides which robots to move, plans the overall movement of the team, and identifies obstacles. Low-level skills, such as multiple step moves and the coordination of localization and sonar pings, are controlled automatically by the robot team. Research is ongoing to transition from operator-assisted control to a fully autonomous system for which only high-level tasks are required, such as 'map room' or 'find target.' As we develop better planning and control algorithms, operational functionality will move from the operator to the team leader and eventually to the Millibots themselves.

6.1 Operation of the Team

In the first series of experiments, the robots were positioned in the lower left corner of a hallway without obstacles as shown in Figure 7. These experiments were primarily used to test and verify the team's ability to localize and collect map data. At the same time, the operator learned how to coordinate the efforts between robots, that is, which robot to use for mapping, where to explore, and when and how to use the cameras.

The second series of experiments started at the center right of the hallway which included a cluster of objects against one of the walls. The robots were able to detect and avoid the obstacles and remained operational for more than an hour. During one experiment, the battery on a camera robot failed. Since the team was composed of five robots, the loss of a single robot did not jeopardize the group's ability to continue. The inoperative robot was left behind and the mission was continued with the four remaining robots.

In the third and most difficult series of experiments, we added a large number of obstacles, some of which were small and low to the ground making them invisible to the sonar sensors. In these experiments, the camera modules played a significant role. Prior to moving any robot, the camera was used to scan the area in front of the robot. If an object was detected, the area was marked manually on the GUI display and the robots were directed to move around it. The extensive use of the cameras reduced the exploration speed, but allowed the team to progress without colliding with obstacles. This series of tests exposed the inherent weakness of relying on a single mode of sensing and illustrated the utility of a heterogeneous team.

The last experiments were conducted in the top right of the hallway. These tests focused on how much area the robot team can map in a fixed amount of time. At this point the exploration rate is relatively small, because a significant amount of time is needed for the human to react to sensor feedback and make decisions. We expect the exploration rate of the robot team to increase significantly when more low-level tasks are handled by the team rather than the human operator.

6.2 Metrics

To evaluate and compare the performance of the team, we have developed a set of metrics. Well-defined metrics allow comparison between Millibot teams with different compositions, teams using different algorithms, as well as other robots and teams of robots.

Figure 8 lists the metrics obtained for the four runs described above. The first three columns report the logistical measures including the number sensing and acting commands. The last four columns represent the performance metrics.

The column labeled "# Vector Cmds" lists the number of motion commands given to the team during a run. Each motion command corresponds to a vector with a relative angle and distance. The robot first turns over the specified angle and then moves forward over the commanded distance. The next two columns report the number of distance measurement pairs and the number of sonar readings taken by the team. After each motion command, the team performs distance measurements using the beacon sensors to update the robot position estimates. To improve sensing reliability, both distance measurements and sonar reading are the filtered result of eight separate measurements.

The performance of the robot team is characterized by the duration to perform a mission, the area explored per unit of time (coverage rate), the dimensional accuracy, and the power consumption. The dimensional accuracy is obtained by comparing the distance between mapped features (e.g., two walls) with the actual distance. The energy consumption is measured by comparing the battery charge for each robot before and after the experiment. An electronic battery monitor allows us to measure the remaining battery charge accurately.

The metrics reflect the difficulties encountered in the experiments. For example, the third run (with a large number of obstacles) required the extensive use of the camera. This resulted in a reduced coverage rate and larger power consumption. The final experiments also resulted in larger power consumption, because the robots spent relatively more time moving around, which is very energy

	# Vector Cmds	# Distance Meas (x 8)	# Sonar Pings (x 8)	Duration	Coverage Rate	Dim. Accuracy	Power Con-sumption
run1	63	303	63	30 min	3.2 m^2/hour	2.5 %	3.2 W
run2	92	496	96	63 min	1.9 m^2/hour	2.5 %	3.2 W
run3	90	485	465	76 min	1.2 m^2/hour	*	3.4 W
run4	86	431	450	30 min	3.3 m^2/hour	1.6 %	3.8 W
Total	331	1715	1074	199 min			

Figure 8: Experiment data

intensive. The dimensional accuracy reflects a combination of sonar accuracy and localization accuracy. A deviation of less than 3% in the distance measurement between two walls of the mapped hallway is surprisingly good for team of robot of this size.

7 Summary

In this article, we have presented the design of a distributed robotic system consisting of very small mobile robots called Millibots. Although the Millibots are small, they still contain a full set of integrated capabilities including sensing, computation, communication, localization, and mobility. To expand the capabilities even further, the Millibots have been designed in a modular fashion allowing one to easily create specialized robots with particular sensing configurations. By combining several such specialized robots, one can create a team with a very broad range of capabilities while still maintaining a small form factor.

An important component of the Millibots is a novel ultrasound-based localization system. This system has the important advantage over currently existing systems that it does not require any fixed beacons. By using the Millibots alternately as beacons and as localization receivers, the team as a whole can reposition while maintaining accurate localization estimates at all times.

Tracking robot positions accurately is especially important for the mapping and exploration application that we have implemented. Each robot explores an unknown environment with its sonar and IR sensors. A team leader collects all the sensor information and integrates it into a global view of the environment. The team leader uses an occupancy grid representation with a Bayesian update to fuse the sensor data over time to build a composite map of the area.

Acknowledgements

The authors would like to thank all the current and past members of the CyberScout team for their contributions, specifically, Curt Bererton, Pete Boettcher, Ethan Bold, Ben Brown, George Chow, Elliot Delaye, Brian Dougherty, Francine Gemperle, Dave Harden, Chris Inacio, Tony Nolla, and Rebecca Schreiber.

This research is funded in part by the by the Distributed Robotics program of DARPA/ETO under contract DABT63-97-1-0003, and by the Institute for Complex Engineered Systems at Carnegie Mellon University. L.E. Navarro-Serment was supported by CONACyT and ITESM Campus Guadalajara.

References

[1] Arkin, R.C. and Balch, T.R. 1998. Cooperative Multiagent Robotic Systems AI-based Mobile Robots: Case Studies of Successful Robot Systems. Kortenkamp, D., Bonasso, R.P. and Murphy, R. (eds). MIT Press.

[2] Atiya, S. and Hager, G. 1993. Real-time Vision-based Robot Localization. *IEEE Transactions*

on Robotics and Automation, Vol. 9, No. 6, pp. 785-800.

[3] Borenstein, J., Everett, H. R., and Feng, L., 1996. *Navigating Mobile* Robots*: Sensors and Techniques*, Wellesley, MA: A. K. Peters, Ltd.

[4] Brooks, R. A. 1986. A Robust Layered Control System for A Mobile Robot. IEEE Journal of Robotics and Automation. Vol. RA-2. No. 1. March. pp 14-23.

[5] Conticelli, F., and Khosla, P.K. 1999. Image-Based Visual Control of Nonholonomic MobileRobots. Technical Report, ICES04-05-99. The Institute for Complex Engineered Systems. Carnegie Mellon University. Pittsburgh, PA 15213.

[6] Diehl, P.D, Saptharishi, M., Hampshire, J.B., and Khosla, P.K. 1999. Collaborative Surveillance Using Both Fixed and Mobile Unattended Ground Sensor Platforms. *SPIE's 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense)*, 5–9 April, Marriott's Orlando World Center, Orlando, Florida USA.

[7] Dixon, K., Dolan, J., Huang, W., Paredis, C., Khosla, P., "RAVE: A Real and Virtual Environment for Multiple Mobile Robot Systems," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), Kyongju, Korea, October 17-21, 1999.

[8] Dixon, Kevin R., Theodore Q. Pham, and Pradeep K. Khosla, "Port-Based Adaptable Agent Architecture", to appear in Proceedings of the International Workshop on Self-adaptive Software, 2000.

[9] Dolan, J. M., Trebi-Ollennu, A., Soto, A., and Khosla, P. K, 1999. Distributed Tactical Surveillance with ATVs. *SPIE's 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense)*, 5–9 April, Marriott's Orlando World Center, Orlando, Florida USA.

[10] Elfes, A. 1989. Occupancy Grids: A Probabilistic Framework for Mobile robot Perception and navigation. Ph.D. Thesis. Department of electrical and computer engineering. Carnegie Mellon University.

[11] Fletcher, R. 1987, Practical methods of optimization (second edition). New York, NJ: J. Wiley & Sons, Ltd.

[12] Getting, I. A. 1993. The Global Positioning System. *IEEE Spectrum*, December, pp. 36-47.

[13] Hollis, R. 1996. Whither Microbots? Proc. 7th. Int'l. Conf. On Micromachine and Human Science (MHS '96). Nagoya, Japan. October 2-5.

[14] ISR - IS Robotics, Inc., 1994. "RR-1/BS-1 System for Communications and Positioning - Preliminary Data Sheet." IS Robotics, Twin City Office Center, Suite 6, 22 McGrath Highway, Somerville, MA 02143, 617-629-0055.

[15] Jenkin, M., Milios, E., Jasiobedzki, P., Bains, N., and Tran, K. 1993. Global Navigation for

ARK. *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems*. Yokohama, Japan, July 26-30, pp. 2165-2171.

[16] Kleeman, L. 1992. (May, Nice, France) Optimal estimation of Position and Heading for Mobile Robots Using Ultrasonic Beacons and Dead-reckoning. *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*. pp.2582-2587.

[17] Lekei, D., 1997. "Using a PIC16C5X as a Smart I^2C Peripheral", AN541, Microchip Technology, Inc. Chandler, AZ.

[18] Leonard, J. F. and Durrant-Whyte, H. F., 1991. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions on Robotics and Automation*. Vol. 7. No. 3. pp. 376-382.

[19] Mataric, M. 1995. Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems,* 16(2-4), Dec. 1995. pp. 321-331.

[20] McLurkin, J.D. Using Cooperative Robots for Explosive Ordnance Disposal. *Technical Document*. Massachusetts Institute of Technology. Artificial Intelligence Laboratory. Cambridge, MA USA 02139.

[21] Mondada, F., Franzi, E., and Ienne, P. 1993. Mobile Robot Miniaturization: a Tool for Investigation in Control Algorithms. *ISER'93*, Kyoto, Japan, October 1993.

[22] Moravec, H. P., "Sensor fusion in evidence grids for mobile robots," AI Magazine, pp 61-74, 1988.

[23] Navarro-Serment, L.E, Paredis, C.J.J., and Khosla, P.K. 1999. "A Beacon System for the Localization of Distributed Robotic Teams," In *Proceedings of the International Conference on Field and Service Robotics*, Pittsburgh, PA, August 29-31, 1999.

[24] Parker, L.E. 1999. Adaptive Heterogeneous Multi-Robot Teams, *Neurocomputing, special issue of NEURAP '98: Neural Networks and Their Applications 1999*, vol. 28, pp. 75-92.

[25] Rus, D., Donald, B.R., and Jennings, J. 1995. Moving Furniture with Teams of Autonomous Mobile Robots, in *Proc. IEEE/Robotics Society of Japan International Workshop on Intelligent Robots and Systems, (IROS)*. Pittsburgh, PA.

[26] Salido, J., Paredis, C. J. J., and Khosla, P. K. 1999. Continuous Probabilistic Mapping by Autonomous Robots. To appear *in Proceedings of the International Symposium on Experimental Robotics*.

[27] Stuck, E. R., Manz, A., Green, D. A., and Elgazzar, S., 1994. Map Updating and Path Planning for Real-Time Mobile Robot Navigation. *1994 International Conference on Intelligent Robots and Systems (IROS '94)*. Munich, Germany, Sept. 12-16, pp. 753-760.

[28] Thrun, S. 1997. Learning Maps for Indoor Mobile Robot Navigation. *AI Magazine*.

[29] Veloso, M., Stone, P., Han, K. and Achim, S. 1998. The CMUnited-97 Small Robot Team. In *Proceedings of RoboCup-97: The First Robot World Cup Soccer Games and Conferences*, Kitano, H. (ed.). Springer Verlag, Berlin.