# Experiments with Robots and Sensor Networks for Mapping and Navigation

Keith Kotay<sup>1</sup>, Ron Peterson<sup>2</sup>, and Daniela Rus<sup>1</sup>

- <sup>1</sup> Massachusetts Institute of Technology, Cambridge, MA, USA {kotay|rus}@csail.mit.edu
- <sup>2</sup> Dartmouth College, Hanover, NH, USA rapjr@cs.dartmouth.edu

**Summary.** In this paper we describe experiments with networks of robots and sensors in support of search and rescue and first response operations. The system we consider includes a network of Mica Mote sensors that can monitor temperature, light, and the movement of the structure on which they rest. We also consider an extension to chemical sensing in simulation only. An ATRV-Mini robot is extended with a Mote sensor and a protocol that allows it to interact with the network. We present algorithms and experiments for aggregating global maps in sensor space and using these maps for navigation. The sensor experiments were performed outdoors as part of a Search and Rescue exercise with practitioners in the field.

Key words: sensor network, search and rescue, robot navigation

# 1 Introduction

A network of robots and sensors consists of a collection of sensors distributed over some area that form an ad-hoc network, and a collection of mobile robots that can interact with the sensor network. Each sensor is equipped with some limited memory and processing capabilities, multiple sensing modalities, and communication capabilities. Sensor networks extend the sensing capabilities of the robots and allow them to act in response to events outside their perception range. Mobile robots extend sensor networks through their ability to bring new sensors to designated locations and move across the sensor field for sensing, data collection, and communication purposes. In this paper we explore this synergy between robot and sensor networks in the context of search and rescue applications.

We extend the mapping and navigation algorithms presented in [8] from the case of a static sensor network to that of a mobile sensor network. In this algorithm, the sensor network models the sensor readings in terms of "danger" levels sensed across its area and creates a global map in sensor space. The regions that have sensor values above a certain threshold are represented as danger. A protocol that combines the artificial potential field of the sensors with a notion of "goal" location for a mobile node (perhaps to take a high resolution picture) computes a path across the map that maintains the safest distance from the danger areas. The focus of this paper is on particular issues related to building systems of sensors and robots that are deployable in realistic physical situations. We present sensor network mapping data from our participation in a search and rescue exercise at Lakehurst, NJ. We then show how these kinds of maps can be used for navigation in two settings: (1) in a simulated scenario involving chemical spills and (2) in a physical scenario implemented on Mica Motes [3] that sense light and guide an ATRV-Mini robot.

# 2 Related Work

This work builds on our research agenda for providing computational tools for situational awareness and "googling" for physical information for first responders [5]. We are inspired by previous work in sensor networks [2] and robotics [6]. [7] proposes a robot motion planner that rasterizes configuration space obstacles into a series of bitmap slices, and then uses dynamic programming to compute the distance from each point to the goal and the paths in this space—this is the inspiration for our distributed algorithm. This method guarantees that the robot finds the best path to the goal. [4] discusses the use of an artificial potential field for robot motion planning. The concept of using a sensor network to generate a potential field of sensed "danger" and then using this information to generate a path of minimum danger from a start location to a goal location was proposed in [8]. In this paper, the proximity to danger is based on the number of hops from nodes which broadcast danger messages, and the total danger is the summation of the danger potentials from all nodes which sense danger. Then, given start and goal node locations, it is possible for the network to direct the motion of the agent from node to node along a path which minimizes the exposure of the agent to danger. In a related work, [1] addresses coverage and exploration of an unknown dynamic environment using a mobile robot and beacons.

# 3 Guiding Algorithm

To support guidance, the sensor network computes an adaptive map in perception space. The map is used by mobile nodes to compute safe paths to goal locations. The goals may be marked by a user or computed internally by the network. The map is built from locally sensed data and is represented globally as a gradient starting at the nodes that trigger sensor values denoting danger, using the artificial potential protocol described in [8]. Given such a map, we

<sup>2</sup> Keith Kotay, Ron Peterson, and Daniela Rus

3

Algorithm 1 Algorithm for following a path to the goal node.

```
1: GoalId = 19
2: QueryId = NONE
3: NextNode.Id = NONE
 4: NextNode.Potential = POTENTIAL_MAX
5: NextNode.Position = (0, 0)
6: Error = RobotSynchronize()
 7: while !Error AND (NextNode.Id != GoalId) do
8:
     if QueryId == NONE then
9:
        Address = TOS\_BCAST\_ADDR \{send query to all nodes\}
10:
     else
11:
        Address = QueryId {send query to the next node on the goal path}
12:
     NextNode.Id = NONE {set this to detect 0 responses}
13:
     NodeQuery(Address, GoalId) {send the query}
14:
     for all Node query responses, R_i do
        if (R_i.Potential < NextNode.Potential) OR ((R_i.Potential = 
15:
        NextNode.Potential) AND (R_i.Hops < NextNode.Hops)) then
16:
          NextNode.Id = R_i.Id
17:
          NextNode.Potential = R_i.Potential
          NextNode.Hops = R_i.Hops
18:
19:
          NextNode.PriorId = R_i.PriorId
20:
          NextNode.Position = R_i.Position
21:
     if (NextNode.Id == NONE) OR ((QueryId != NONE) AND
     (NextNode.Id != QueryId)) then
22:
        QueryId = NONE \{no valid response, go back to broadcast\}
23:
     else
24:
        QueryId = NextNode.PriorId {PriorId is the next node on the goal path}
25:
        Error = RobotMove(NextNode.Position) \{move to position of best R_i\}
```

modify the algorithm in [8] to compute safe navigation paths as shown in Algorithm 1.

Once a path query message is sent, the replies are processed to select the best path available. This is done by selecting the response with the lowest danger potential. If two or more replies with the minimum danger potential are received, the reply with the minimum number of hops to the goal is used to select the path.

# 4 Sensor Experiments

## 4.1 Search & Rescue Experiments

Experiments were conducted on February 11, 2005 at the Lakehurst Naval Air Base, NJ as part of the New Jersey Task Force 1 search and rescue training exercise. The purpose of the experiments was to validate the utility of sensor networks for mapping, to characterize the ambient conditions of a typical environment for search and rescue missions. 34 Mica Motes with light,

#### 4 Keith Kotay, Ron Peterson, and Daniela Rus

temperature, and acceleration sensors were manually placed on a large pile of rubble which is used to simulate the environment of destroyed buildings. The rubble consists mostly of concrete with embedded rebar or steel cable, though various destroyed appliances and scraps of sheet metal are also constituent elements. In this experiment, node locations were given to the Motes via radio messages after deployment.

The sensors were in place by 11:15am and gathered data until 12:45pm. The sensor data was stored on each Mote and downloaded at a later time. The day was cold (below freezing at the start of tests), clear and sunny, and very windy. The sensors were placed at approximately 1.2 meter intervals to form a 6x6 meter grid. People and robots were traversing the rubble while readings were taken. The particular section of rubble where the sensors were placed can be seen in Fig. 1.



Fig. 1. Wide angle view of sensors placed on rubble. Most of the sensors are behind outcroppings and hence are not visible. The circles show the locations of a few of the sensors.

To protect them from dust and weather, the sensors were placed in ziploc freezer bags. All had fresh batteries and were placed so the light sensor was generally pointing up, though most of the sensors were not on level surfaces. After about 35 minutes, sensor 23 blew off its concrete perch and fell down a two meter hole. This event is discernable in the graphs that follow. The strong winds also rearranged some of the sensors during the course of the experiment. Four sensors failed, most likely due to temperature extremes, and hence produced no data.

#### Sensor Radio Connectivity

The connectivity between the sensors was measured by each sensor sending and acknowledging 20 ping messages. The results are shown in Fig. 2 (left).



**Fig. 2.** Connectivity and acceleration data for sensors placed on rubble. The left image shows connectivity between sensors in the rubble field. The relative thickness of the lines indicates the connection strength. Those sensors near the top of the rubble pile had poor connectivity. The right images show X-axis (top) and Y-axis (bottom) acceleration data for Sensor A, shown in Fig. 1 as the leftmost black circle.

While the lower laying sensors, which were mostly on flat slabs of concrete, had reasonable connectivity, the sensors embedded in the more diverse rubble higher in the pile had poor connectivity. In previous experiments we have found that we get fair connectivity with a distance of two meters between sensors lying on earth. Even with the shorter distance we used between sensors here (1.2 meters) the metal in the environment has reduced connectivity dramatically. The three dimensional nature of the sensor deployment is also likely to have had an effect, since the antennas of the sensors are not on the same plane, and have varying orientations. The antennas have a toroidally-shaped reception pattern with poor connectivity along the axis of the antenna.

## Light Sensing

Fig. 3 shows a two dimensional light intensity map at three different times during the experiment. The map is bilinearly interpolated from the sensed light values, with the nearest two sensors contributing to the points in the map between them. Because the light sensors were pointed at the sky and it was a bright day, they saturated at their maximum value for most of the test, even if they were in shadow, although near the end of the test shadows from the moving sun brought some of the sensors out of saturation. The light reading for sensor 23 goes down after it falls into a hole. Later, light again reached the sensor from another angle. Other sensors experienced some brief

## 6 Keith Kotay, Ron Peterson, and Daniela Rus

changes in intensity due to the shadows of people walking by. The shadows from the rubble and the wind causing sensors to shift account for the rest of the changes.



Fig. 3. Sensed light intensity map at times of 35, 55, and 90 minutes.

#### **Temperature Sensing**

Fig. 4 shows a two dimensional temperature map at three different times during the experiment. The temperature varied dramatically over time and based on sensor location. The Motes got quite warm (40C, 105F), which is surprising since the day was cold and there was a strong wind blowing, though the day warmed up gradually. The Motes were in plastic bags and the black plastic of the battery holder and the black heat sensor itself were exposed directly to sunlight. The bags acted as insulators from the cold, holding warm air inside and the sunlight on the black plastic heated the air in the bags. The black heat sensors themselves were also heated to higher than surrounding temperature by the sun. This is quite interesting since it shows that a sensor in an environment isn't necessarily sensing that environment. It needs some direct connection to the outside world, or else it is only sensing it's own microclimate.

Mote 23 fell into a hole at about 35 minutes and cools down slowly (when viewing the complete data set). The sensors near the base of the rubble and on the peak of the rubble were mostly laying exposed on flat surfaces. The in-between sensors were in amongst jumbled rubble and recorded cooler temperatures. The sensors most exposed to the sun became the warmest. The changes in temperature were caused by the sun warming the air as it rose higher, the shifting of shadows in the rubble, and sensors being shifted by the wind.

## Acceleration Sensing

In addition to the light and temperature sensors, three acceleration sensors were deployed. The sensing element was an Analog Devices ADXL202E, 2-

7

axis device, capable of measurements down to 2 milli-gs, with a +/-2g range. Due to a higher data rate, we were only able to record about a half hour of readings.



Fig. 4. Sensed temperature map at times of 5, 25, and 65 minutes.

Fig. 2 (right) shows the readings from the sensor laying on the ground, The readings at the start of each graph show the sensors being placed in position. Sensor A was picked up and then replaced by a task force member halfway through the experiment. It apparently slowly tipped over during the course of the next four minutes. The other large shifts in the readings are due to wind blowing the plastic bags the sensors were in. We have collected similar data from several other sensors placed on loose rubble at various points up the rubble pile.

Between wind, weather, shifting rubble, people moving about, lighting and temperature changes due to the motion of the sun, local variations in line of sight, and the jumbling of the radio environment by sheet metal and rebar, this is clearly a challenging environment for wireless sensing applications.

## 4.2 Chemical Sensing Experiment

In many first response calls the presence of deadly, invisible chemicals is first noticed when people start coughing or falling ill. Even after the presence of a gas has been verified, unless it is visible it is difficult to avoid exposure due to air motion. Chemical sensing networks can provide a first warning of nearby toxins, and more importantly, can tell us where they are, where they are moving towards, and how to avoid them.

As part of ongoing work in medical and environmental sensors for first responders, we devised a simulated air crash scenario that involves a chemical leak. The crash throws some debris into a nearby farmers field where a tank of anhydrous ammonia used as fertilizer is present on a trailer attached to a tractor. Anhydrous ammonia, when released into the atmosphere, is a clear colorless gas, which remains near the ground and drifts with the wind. It attacks the lungs and breathing passages and is highly corrosive, causing damage even in relatively small concentrations. It can be detected with an appropriate sensor such as the Figaro TGS 826 Ammonia sensor. We ran experiments designed to map the presence of an ammonia cloud and guide a first responder to safety along the path of least chemical concentration. The sensors were Mica Motes, programmed with the same potential field guidance algorithm described above in Section 3. Light sensors on the Motes were used instead of ammonia sensors, due to the difficulty of working with ammonia gas.

The sensors were programmed with locations arranged in a grid with 50 feet between sensors. The experiment was carried out on a tabletop with the RF transmission range of the sensors reduced to match a physically larger deployment. Radio messages between sensors were limited by software to one hop, using the known locations of the sensors, to reduce message collisions. Potential field messages were repeated five times to ensure their propagation. The computed field values were read from the sensors every four seconds to update a command and control map display. It took 10 to 15 seconds for the potential field to propagate each new event and stabilize. Chemical detections were triggered at sensors 9, 20, and 32.



**Fig. 5.** (Left) Potential field danger map computed by sensors in response to the simulated presence of a chemical agent. (Right) Safest path computed for a trapped first responder by the sensor field.

Fig. 5 (Left) shows a danger map computed by a 38 sensor field after the ammonia has been detected. The danger in the areas between the sensors is computed using a bilinear interpolation of the stored potential field values from the nearest two sensors to each point on the map.

After the ammonia has moved into the locale of the field operations, a first responder located in the lower left corner (the + symbol there) needs guidance to safely find a way to the operations base (the + symbol in the upper right corner.) The guidance algorithm uses the potential field to compute all safest directions from one sensor to the next, and then computes the overall safest

<sup>8</sup> Keith Kotay, Ron Peterson, and Daniela Rus

and most direct path for the first responder to follow, which minimizes the exposure to the chemical agent. Fig. 5 (Right) shows the safest path computed by the sensor field.

Such a system can not be relied on by itself for guiding people through danger. This is due in part to the presence of obstacles which the sensors know nothing about, such as fences, bodies of water, vehicles, etc. In addition, the commander on the scene may have additional knowledge which overrides the path chosen by the sensors (e.g., if there's a tank of jet fuel which is overheating along part of the path, it may be best to try a different way, even if the chemical haze is thicker.) Thus, although the sensors guidance computations can not be the sole source for guidance, they can be a very useful addition to the knowledge which the commander and responders in the field use to choose a way to safety.

# 4.3 Robot Navigation Experiment

We implemented the algorithm used for the simulations in Fig. 5 and the algorithm for safe path following, Algorithm 1, on a system of physical robots and sensors. In our implementation we have a notion of "obstacle" or "danger" much like in the chemical spread simulation. The values for danger could come from any of the sensors deployed and tested as described above.

#### **Experimental Setup**

Our experimental setup consists of a network of Mica Motes suspended above the ground and an iRobot ATRV-Mini robot being guided through the network space. The network space was above a paved area surrounded by grass. The goal was to guide the robot from a start location to a goal location along a curved path, while avoiding any grassy area which corresponds to "danger".

#### Network

The network is comprised of 18 Mica Motes attached to a rope net suspended above the ground (see Fig. 6). Motes were attached at the rope crossing points, spaced 2 meters apart. Deploying nodes above ground level improves radio performance and prevents damage from the robot wheels. Although the use of a net is not representative of a typical real-world deployment, it allowed research to proceed without fabricating protective enclosures for the Motes and it does not invalidate the algorithmic component of the work.

In our implementation, the location of the nodes in the network is known a priori, since the Mica Motes are not capable of self-localization without extra hardware. When the robot communicates with the next node on the path to the goal, the node passes its location to the robot. The robot then uses its compass and odometry to move to the node location.



Fig. 6. The network configuration for the guided navigation experiments. 18 Mica Motes are located at the junctions of the ropes, indicated by the ID numbers. Nodes 10, 11, 12, and 18 broadcast "danger" messages, and node 19 is the goal. Node number 1 is on the robot, communicating with the network and guiding the robot. The robot is shown in the starting position for the experiments.

The presence of "danger" was detected by uncovering the light sensor on the Mica Mote sensor board. This would cause the node to broadcast five separate danger messages, which would then flood the network due to each receiver relaying the messages to its neighbors. Multiple messages were used as a means of determining "reliable" communication links—if the number of received danger messages is above a threshold based on the number of expected danger messages, then the link is determined to be reliable, and is therefore suitable to be on a guiding path [8]. In the experimental setup shown in Fig. 6, nodes 10, 11, 12, and 18 sensed danger, equivalent to being over grass in this case. The goal node was number 19, and the robot starting position is shown in Fig. 6. For these condition, the optimal path consists of nodes 2, 3, 4, 9, 13, 16, 19.

The robot used in our experiments is an iRobot ATRV-Mini with a fourwheel skid-steer drive. It is equipped with an internal computer running Linux, as well as odometry, sonar, compass, and GPS sensors. For our experiments, the sonar sensors were only used to avoid collisions by stopping the robot if any sensor detected an object with 25cm of the robot. The GPS sensor was not used in our experiments, since its resolution was inadequate for the size of the network. Odometry was used to measure forward motion of the robot, and the compass was used to turn the robot to a new heading.

The interface to the network is by means of an onboard Mote connected to the robot by a serial cable. In fact, the onboard Mote is in command of the system and the ATRV-Mini is merely a locomotion slave. The path algorithm described in Section 3 is run on the Mote, which sends motion commands to the robot and receives acknowledgements after the move is completed.

#### **Experimental Results**

Experiments were performed with the setup shown in Fig. 6. A sequence of snapshots of an experimental run is shown in Fig. 7. Initial experiments were hampered by poor compass performance on the ATRV-Mini robot, due to the compass being mounted too close to the drive motors. Despite turning the robot drive motors off to minimize the compass deflection the compass heading was not precise, due to the large amount of steel in the adjacent building and buried electrical cables. The result is some additional deviation from the path node positions as shown in Fig. 7.



Fig. 7. Six snapshots of a guided navigation experiment. The Mica Motes are located as shown in Fig. 6. The optimal node sequence is 2, 3, 4, 9, 13, 16, 19. Because the viewing angle is not directly from above and there is some distortion in the net, the robot does not line up exactly with the node locations.

Despite the difficulties with the compass, the navigation was successfully performed many times. Although the final location of the robot is offset from node 19, the robot did follow the path of minimum danger and avoided the danger areas while being guided by the sensor network. We plan to conduct further experiments to get better statistics on the precision of this navigation algorithm.

## Discussion

This implementation demonstrated robot guidance by a sensor network. The sensor network is very effective at computing useful global maps in perception space. However, the precision of the navigation system is greatly dependent on the robot hardware. Navigation by compass can be problematic if environmental factors such as electrical cables and steel structures exist. Although it

#### 12 Keith Kotay, Ron Peterson, and Daniela Rus

is possible to compensate for these effects in a known environment, a search and rescue scenario may not permit elaborate offline calibration.

Another option would be to use a directional antenna in place of (or in addition to) the compass. The standard Mica Mote antenna is omnidirectional, but using a directional antenna would allow the robot to determine a bearing to the next node in the goal path. This technique, coupled with the use of radio signal strength (RSSI) to determine the proximity of the robot to a node would make network localization optional, since the robot could directly sense its proximity to a node.<sup>3</sup> Since localization is sometimes not possible in a sensor network due to the cost of additional hardware such as GPS sensors or acoustic transducers like those on the MIT Crickets, enabling the robot to move through a non-localized network is a useful feature. It is also cost effective since adding extra hardware to a small number of robots is less costly than adding localization hardware to all the nodes in a large sensor network.

## **5** Acknowledgements

This work has been supported in part by Intel, Boeing, ITRI, NSF awards 0423962, EIA-0202789, and IIS-0426838, the Army SWARMS MURI. This project was also supported under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security.

# References

- M. Batalin and G.S. Sukhatme. Efficient exploration without localization. In Int. Conference on Robotics and Automation (ICRA), Taipei, May 2003.
- D. Estrin, R. Govindan, and J. Heidemann. Embedding the internet. Communications of ACM, 43(5):39–41, May 2000.
- J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In ASPLOS, pages 93–104, 2000.
- D. Koditschek. Planning and control via potential fuctions. Robotics Review I (Lozano-Perez and Khatib, editors), pages 349–367, 1989.
- 5. V. Kumar, D. Rus, and S. Singh. Robot and sensor networks for first responders. *Pervasive Computing*, 3(4):24–33, December 2004.
- 6. J.-C. Latombe. Robot Motion Planning. Kluwer, New York, 1992.
- J. Lengyel, M. Reichert, B. Donald, and D. Greenberg. Real-time robot motion planning using rasterizing computer graphics hardware. In *Proc. SIGGRAPH*, pages 327–336, Dallas, TX, 1990.
- 8. Q. Li, M. de Rosa, and D. Rus. Distributed algorithms for guiding navigation across sensor networks. In *MOBICOM*, pages 67–77, San Diego, October 2003.

<sup>&</sup>lt;sup>3</sup> Although RSSI cannot provide reliable absolute distance estimation, it still may be sufficient for determining the point of closest approach to a stationary Mote.