Deployment and Connectivity Repair of a Sensor Net with a Flying Robot

P. Corke[1] and S. Hrabar[2] and R. Peterson[3] and D. Rus[4] and S. Saripalli[2] and G. Sukhatme[2]

[1] CSIRO ICT Centre
Australia, peter.corke@csiro.au
[2] Center for Robotics and Embedded Systems
University of Southern California, Los Angeles, California, USA
shrabar@robotics.usc.edu, srik@robotics.usc.edu, gaurav@robotics.usc.edu
[3] Dartmouth Computer Science Department, Hanover, NH 03755 USA,
rapjr@cs.dartmouth.edu
[4] Computer Science and Artificial Intelligence Laboratory
MIT, Cambridge MA 02139, USA, rus@csail.dartmouth.edu

Abstract. We consider multi-robot systems that include sensor nodes and aerial or ground robots networked together. Such networks are suitable for tasks such as large-scale environmental monitoring or for command and control in emergency situations. We present a sensor network deployment method using autonomous aerial vehicles and describe in detail the algorithms used for deployment and for measuring network connectivity and provide experimental data collected from field trials. A particular focus is on determining gaps in connectivity of the deployed network and generating a plan for repair, to complete the connectivity. This project is the result of a collaboration between three robotics labs (CSIRO, USC, and Dartmouth.)

1 Introduction

We wish to develop distributed networks of sensors and robots that perceive their environment and respond to it. To perform such tasks there needs to exist a synergy between mobility and communication. Sensor networks provide robots with faster and cheaper access to data beyond their perceptual horizon. Conversely robots can assist a sensor network by deploying it, by localizing network elements post deployment [6], by making repairs or extensions as required, and acting as "data mules" to relay information between disconnected sensor clusters.

In this paper we describe our algorithms and experiments for deploying sensor networks using an autonomous helicopter. The static sensor nodes are Mica Motes and the mobile node is the autonomous helicopter. Once on the ground, the sensors establish an ad-hoc network and compute their connectivity map in a localized and distributed way. If the network is disconnected, a localized algorithm determines waypoints for the helicopter to drop additional nodes at.

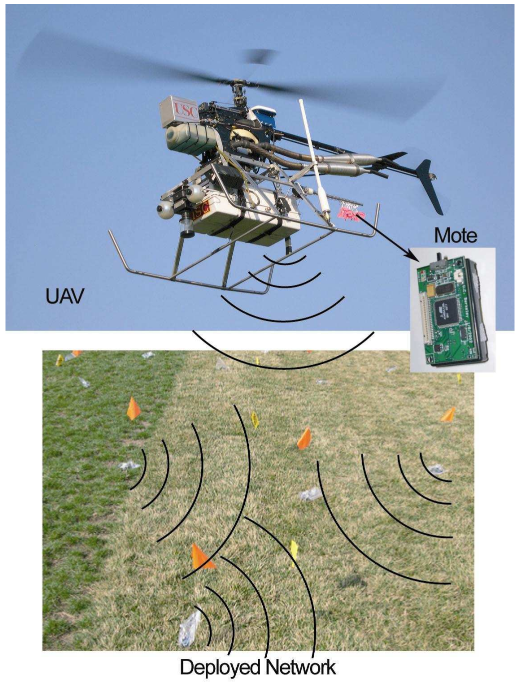

Fig. 1. AVATAR Autonomous Helicopter with a sensor interface for deploying sensors

2 Related Work

Our work builds on important previous work in sensor networks [8, 11, 14] and unmanned aerial vehicles [3, 16]. It bridges the two communities by integrating autonomous control of flying vehicles with multi-hop message routing in ad-hoc networks. Autonomous aerial vehicles have been an active area of research for several years. Autonomous model helicopters have been used as testbeds to investigate problems ranging from control, navigation, path planning to object tracking and following. Flying robot control is a very challenging problem and our work here builds on successes with hovering and control for two autonomous helicopters [3,17]. Several other teams are working on autonomous control and other varied problems with helicopters. A good overview of the various types of vehicles and the algorithms used for control of these vehicles can be found in [17] . Recent work has included autonomous landing [16,19], aggressive maneuvering of helicopters [9] and pursuit-evasion games [21].

Research in sensor networks has been very active in the recent past. An excellent general introduction on sensor networks can be found in [8]. An overview of hardware and software requirements for sensor networks can be found in [12] which describes the Berkeley Mica Motes. Algorithms for positioning a mobile sensor network includes even dispersal of sensors from a source point and redeployment for network rebuilding [2, 13]. Other important contributions include [1, 4, 10, 15, 18].

In [6] we describe a decentralized and localized algorithm called *robot-assisted localization* for localizing a sensor network with a robot helicopter. In [7] we describe an algorithm called *network-assisted navigation* in which a sensor network guides a robot helicopter. In [5] we describe an algorithm and preliminary experiments for deploying a sensor network with a robot helicopter. Here we extend this work to include deployment and connectivity repair and discuss our field experiments using a autonomous helicopter and a 55-node sensor network.

3 Approach

Our approach consists of three phases. In the first phase, an initial autonomous network deployment is executed. In the second phase, the entire network measures its connectivity topology. If this topology does not match the desired topology, a third phase is employed in which waypoints for the helicopter are computed at which additional sensors are deployed. The last two phases can be run at any point in time to detect the potential failure of sensor nodes and ensure sustained connectivity.

3.1 Deployment Algorithm

Given a desired network topology for the deployed network deployed, and a deployment scale (usually the inter-sensor distance between the nodes in the network), we embed the topology in the 3-dimensional hyper-plane at the given location and extract desired node locations from the resulting embedding. The resulting locations are the (x, y, z) co-ordinates where the sensors need to be deployed.These are given as way-point inputs to the helicopter controller. The helicopter then flies to each of these way-points autonomously, hovers at each of them and then deploys a sensor at the specified location.

3.2 Connectivity Measurement Algorithms

Two methods were used to measure network connectivity: a ping-based connectivity measure and a token-passing based measure. For the ping-based measure, a Mote sensor that has been specially modified to add physical user interface controls (a potentiometer and switch) is used to control and configure the sensor side of the ping connectivity tests prior to Algorithm 1 executing.

For the token based connectivity measure each node assumes its network ID as its token. All nodes broadcast and trade tokens as described in Algorithm 2. Tokens are only propagated amongst nodes in connected regions. Thus, disconnected regions will have differing token values. This algorithm is run automatically at 30 second intervals.

Slight differences in connectivity were observed when comparing the ping and token measurements of connectivity and were found to result from the differences

Algorithm 1 Ping connectivity algorithm for ground deployed motes.

Wait for experiment configuration/start message
Initialization: Set configuration *mode* = air-to-ground, ground-to-ground, or ground-to-air. Set *count* = number of ping iterations.
Send a multi-hop forwarding of start message to other motes.

Thread 1
for i=1 to *count* **do**
  if *mode* = ground-to-ground OR *mode* = ground-to-air **then**
    broadcast a ping message.
  Sleep a random interval

Thread 2
while Listen for messages **do**
  if message is a ping **then**
    if *mode* = air-to-ground OR *mode* = ground-to-ground **then**
      reply to ping.
  else if Message is a ping reply. **then**
    tabulate reply.

Termination: broadcast counts of replies per mote ID in response to download message.

in message length. Pings are very short messages (1 byte payload) while token messages are longer (10 byte payload). The longer message length increases the chance of collisions and reduces the probability of reception of token messages.

3.3 Connectivity Repair Algorithm

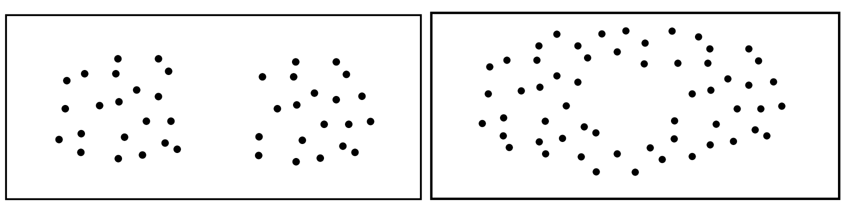

Fig. 2. (Left)Two disconnected components in a sensor network field. (Right) A single network which is not fully connected.

The token based connectivity algorithm is a localized and distributed algorithm for computing connected components in the deployed network. Each node ends up with one token that denotes the group to which it belongs. These tokens are collected by the helicopter during a sweep of the field. If more than one token is collected, the network is not connected and new sensor deployments are needed. The locations of the collected tokens can be used to determine the repair regions.

We have developed two algorithms for repairing network connectivity. In the first algorithm, the robot helicopter estimates the location of the gap between two disconnected components by estimating the locations of the fringe nodes (see Figure 2(Left)). The repair locations are interpolated between the fringes, based on average sensor communication range, which is known.

Algorithm 2 Distributed algorithm for identifying the connected components in a sensor network. All the nodes in one connected component will have the same *component* value as a result of this protocol.

for each node in the sensor network **do**
  component = id
for each node in the sensor network **do**
  broadcast node id.
  while listen for new_{id} broadcasts **do**
    if received id $>$ *component* **then**
      $component = new_{id}$
    broadcast new_{id}
Helicopter collects all *component* values
Helicopter determines unique *component* values as the number of connected components.

In the second algorithm the sensor field computes a potential field to regions of "dark" sensors (see Figure 2(Right)) discovered within it and guides the helicopter there using the potential field algorithm in [7]. This second algorithm handles both complete disconnections and holes in the middle of the sensor field.

For our field experiments we used a hand computed version of the first algorithm described above, averaging the fringe locations to determine a center and averaging the fringe gap distance to determine interpolated repair locations used in the autonomous repair deployment phase.

The general connectivity matching problem remains open. This problem reduces to computing subgraph embeddings which is intractable for the optimal case. We hope to identify a good approximation.

4 Experiments and Results

We have implemented the deployment algorithms on a hardware platform that integrates hardware and software from three labs: USC's autonomous helicopter, Dartmouth's sensor network, and CSIRO's interface between a helicopter and a sensor network. Over January 23–25 the three groups met at USC and conducted joint experiments which demonstrate, for a desired network topology, (1) autonomous deployment of a 40 node sensor network with a robot helicopter, (2) autonomous and localized computation of connectivity maps (3) autonomous determination of disconnected network components and autonomous repair of the disconnections.

4.1 The Experimental Testbed

The experimental testbed consists of three parts (a) An autonomous helicopter (b) ”Mote” sensors and (c) Helicopter-sensor interface. The helicopter [20] is a gas-powered radio-controlled model helicopter fitted with a PC-104 stack augmented with sensors (Figure 1). Autonomous flight is achieved using a *behavior-based* control architecture [16]. Our sensor network platform is the Berkeley Mica Mote [12]. The operating system support for the Motes is provided by TinyOS, an event-based operating system. Our testbed consists of 50 Mote sensors deployed in the form of a regular 11×5 grid, see Figure 4.1. An extra Mote sensor is fitted to the helicopter to allow communications with the deployed sensor network and is connected to the helicopter's Linux-based computer. For further details the reader is referred to [5]. Several applications were run onboard the helicopter, depending on the experiment. The `ping` application sends a broadcast message with a unique id once per second and logs all replies along with the associated Mote identifier. This data allows us to measure air-ground connectivity. The `gps` application receives GPS coordinates via a network socket from the helicopter navigation software and broadcasts it. Simple algorithms in each Mote are able to use these position messages to refine an estimate of their location [6].

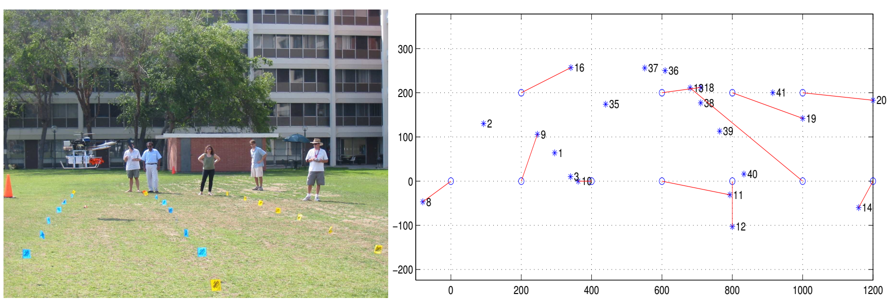

Fig. 3. (Left) The sensor network field with flags marking desired sensor locations. (Right) The locations of a sensor network deployed autonomously by the robot helicopter. The desired locations are denoted by $*$ and they are on a grid. The actual locations are denoted by o.

4.2 Experimental Results

Our field experiments have been performed on a grass field on the USC campus (see Figure 3(left)). We marked a 11×5 grid on the ground with flags. We used an empirical method to determine the spacing of the grid. We established that on that ground, the Mote transmission range was 2.5 meters. We selected the grid spacing at 2 meters so that we would guarantee communication between any neighbors in the field.

4.3 Deployment and Connectivity Results

Figure 3(Right) shows the desired and actual location of the deployed sensors. The deployment error has multiple causes: (a) error in release location compared to desired location (error due to inherent error present in GPS). (b)error in location of markers on ground compared to release location (wind, downwash, and bounce induced error.)

After being deployed the ground sensors establish autonomously an ad-hoc network whose connectivity topology is shown in Figure 4(Top Left). Although there was error in the deployed location, the resulting network is fully connected. We then manually removed 7 nodes down the center of the network to simulate node failure and create a disconnection in the network. The network automatically computed a new token connectivity map as described in Algorithm 2. Figure 4(Top Right) shows the disconnected components as computed by the token algorithm. Finally the robot helicopter autonomously deployed new nodes to repair connectivity resulting in the connectivity map shown in Figure 4(Bottom). Note that some network links were lost in the final graph. Besides some nodes failing due to being out in the hot sun for a day, the introduction of new nodes results in changes in message timing which changes collision rates and hence overall connectivity, even for nodes remote from the area of repair. Mote communication is inherently unreliable as well. The communication range is dependent on relative antenna orientation, shielding (eg. obstacle between two Motes), ground moisture, current receiver autogain levels, etc. The communication links are asymmetric and congestion is a significant concern. We believe that error, uncertainty, and asymmetry are significant factors that should be explicitly included in any model and approach for networked robotics.

4.4 Localization Results

During localization the flying robot followed a preprogrammed path, see Figure 6(Left). The computer onboard the helicopter obtained its current coordinates and broadcast this via the mote attached to the helicopter once every 100ms. Each ground mote recorded all the X,Y broadcasts it received and used them to compute a centroid based location for itself. Figure 6(Right) shows the helicopter height. Figure 5 shows the location of each of the the motes broadcasts received. It is clear that the motes do not receive messages uniformly from all directions. We speculate that this is due to the non-spherical antenna patterns for transmitter and receiver motes, as well as non-uniform height of the helicopter itself during flights.

5 Conclusion

We have described control algorithms and experimental results from sensor network deployment, localization and subsequent repair of the sensor network with an autonomous helicopter. By sprinkling sensor nodes, we can reach remote or dangerous environments such as rugged mountain slopes, burning forests, etc. We believe that

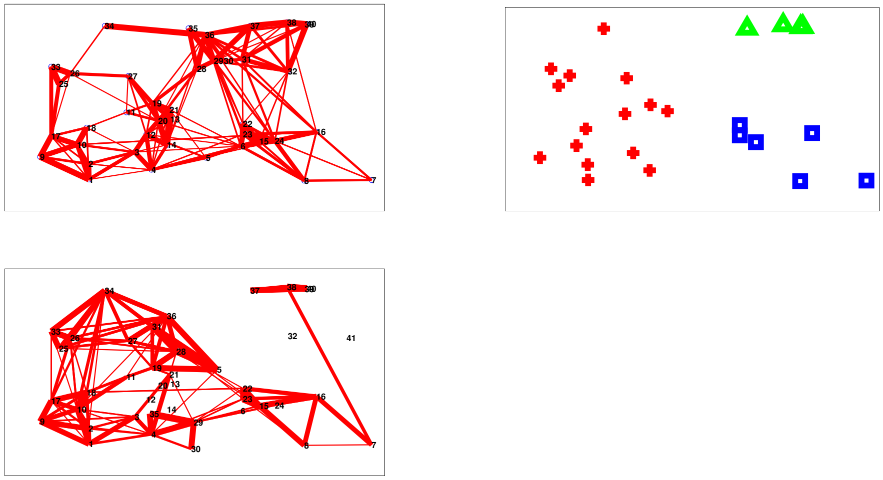

Fig. 4. (Top Left) Connectivity of the initial deployment. (Top Right) Token groups showing connected components after several nodes were removed from the field. (Bottom) Connectivity after the deployment of additional sensor nodes to repair connectivity.

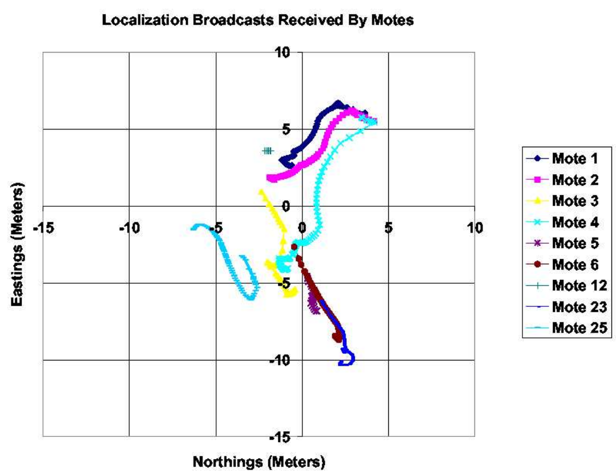

Fig. 5. Location broadcasts heard by some of the motes in the network.

this kind of autonomous approach will enable the instrumentation of remote sites with communication, sensing, and computation infrastructure, which in turn will support navigation and monitoring applications. From what we've learned in these experiments we plan to develop systems for automatic network repair. This will

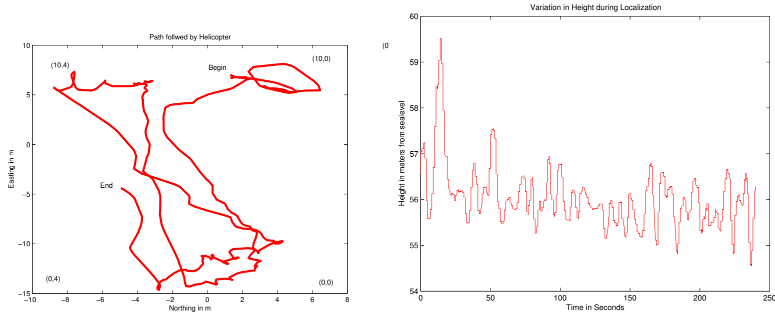

Fig. 6. (Left) The path taken by the helicopter while broadcasting location messages. (Right) The height of the helicopter during the process.

require the ground sensors and helicopter to cooperate to identify network disconnections and guide the helicopter to appropriate locations for autonomous sensor deployment.

Acknowledgment

Support for this work was provided through the Institute for Security Technology Studies, NSF awards EIA-9901589, IIS-9818299, IIS-9912193, EIA-0202789 and 0225446, ONR award N00014-01-1-0675 and DARPA Task Grant F-30602-00-2-0585. This work is also supported in part by NASA under JPL/caltech contract 1231521, by DARPA under grants DABT63-99-1-0015 and 5-39509-A (via UPenn) as part of the Mobile Autonomous Robot Software (MARS) program. Our thanks to our safety pilot Doug Wilson for keeping our computers from (literally) crashing.

References

1. J. Agre and L. Clare. An integrated architeture for cooperative sensing networks. *Computer*, pages 106 – 108, May 2000.
2. M.A. Batalin and G.S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Distributed Autonomous Robotic Systems 5*, pages 373–382, 2002.
3. G. Buskey, J. Roberts, P. Corke, P. Ridley, and G. Wyeth. Sensing and control for a small-size helicopter. In B. Siciliano and P. Dario, editors, *Experimental Robotics*, volume VIII, pages 476–487. Springer-Verlag, 2003.
4. Y. Chen and T. C. Henderson. S-NETS: Smart sensor networks. In *Seventh International Symposium on Experiemental Robotics*, Hawaii, Dec. 2000.
5. P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1143–8, 2004.
6. P. Corke, R. Peterson, and D. Rus. Networked robots: Flying robot navigation with a sensor network. In *ISRR*, 2003.
7. P. Corke, R. Peterson, and D. Rus. Coordinating aerial and ground robots for navigation and localization. In *Submitted to Distributed Autonomous Robotic Systems*, 2004.

8. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *ACM MobiCom 99*, Seattle, USA, August 1999.
9. V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. Control logic for automated aerobatic flight of miniature helicopter. In *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, USA, Aug 2002.
10. P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.
11. J. Hill, P. Bounadonna, and D. Culler. Active message communication for tiny network sensors. In *INFOCOM*, 2001.
12. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *ASPLOS*, 2000.
13. A. Howard, M.J. Mataric, and G.S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable so lution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308, 2002.
14. Q. Li, M. DeRosa, and D. Rus. Distributed algorithms for guiding navigation across sensor networks. In *MOBICOM*, 2003.
15. G. J. Pottie. Wireless sensor networks. In *IEEE Information Theory Workshop*, pages 139–140, 1998.
16. S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–381, June 2003.
17. S. Saripalli, J. M. Roberts, P. I. Corke, G. Buskey, and G. S. Sukhatme. A tale of two helicopters. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, Oct 2003. (To appear).
18. A. Scaglione and S. Servetto. On the interdependence of routing and data compression in multi-hop sensor networks. In *ACM Mobicom*, Atlanta, GA, 2002.
19. O. Shakernia, Y. Ma, T. J. Koo, and S. S. Sastry. Landing an unmanned air vehicle:vision based motion estimation and non-linear control. In *Asian Journal of Control*, volume 1, pages 128–145, September 1999.
20. University of Southern California Autonomous Flying Vehicle Homepage. http://www-robotics.usc.edu/˜avatar.
21. R. Vidal, O. Shakernia, H. J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Transactions on Robotics and Automation*, Oct 2002.