Sensorfly: Swarm Deployment

Brionne S. Godby

ABSTRACT
In this paper we aim to provide a brief overview of the work done and theories developed for the Sensorfly project currently under development at Carnegie Mellon Silicon Valley up to the beginning of August 2010.

General Terms
Sensor Networks, Swarm Intelligence, Ant Colony Optimization

Keywords
Sensor Networks, Swarm Intelligence

1. INTRODUCTION
The Sensorfly system is a low-cost mobile sensor network being developed at Carnegie Mellon- Silicon Valley by professor Pei Zheng and PhD student Aveek Purohit. The Sensorfly system is comprised of a large number of autonomous miniature helicopters (nodes) working together asynchronously to scan a large area. Each node in the swarm has limited scanning and computational power on its own in order to maintain low-cost.

2. WHY SENSORFLY?
The Sensorfly system is designed to be sent into

2.1 Advantages of Sensorfly
The swarm-based design of Sensorfly offers advantages in scalability, mobility, loss tolerance, and cost. Scalable in that the number of nodes deployed in any swarm can be adjusted based on the size of the area that must be searched. Mobility comes from the small size of the individual node allows them to pass through areas a larger single device could not traverse. The swarm system is designed so the loss of an individual node or even several nodes will not render the rest of the swarm to be unable to continue. Finally there is a cost advantage in that one node can be replaced and introduced into an existing swarm for relatively little (each node cost ~$100 to manufacture).

Illustration 2: A Sensorfly Node read for flight

3. THE INDIVIDUAL NODE
The individual Sensorfly node is not very capable and contains only a few essential sensors necessary for

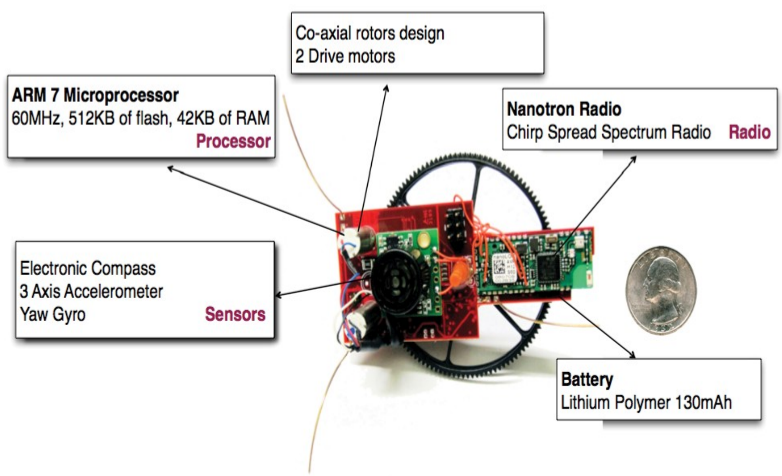

Illustration 1: A breakdown of the sensors located on an individual node

controlled flight. Currently the components onboard each node are an ARM7 microprocessor, CSS radio with short distance time-of flight ranging capabilities, 3-axis accelerometer, 2-axis gyro, magnetic compass, motor control, helicopter mechanical drive, and an extension port for additional sensors. The extension port is currently planned to allow for connection of a video camera, microphone, or thermal camera. The use of optional assists in keeping the cost of individual nodes low, while allowing for specialization within the swarm.[1]

4. GUIDELINES FOR CONTROL

Before looking at the specifics of the swarm control being developed, a few goals of the swarm control system must be stated.

4.1 Mission Objective

The goal of any deployment is for the swarm to fully collect environmental data about a given area. The nature of this data may vary from one mission to the next (i.e. gathering thermal data from burning buildings, video recording for disaster relief, etc).

4.1.1 Assumptions

There are a few key assumptions made about what a full scan constitutes. First and most simply, the entire target area must be scanned if possible, if a section is inaccessible, a note must be made. Second the environment is assumed to be changing, making previously collected data inaccurate as time progresses.

4.2 Independence

Each node must operate relatively independently from other nodes. The main concern is that no node should ever need to wait for a response from a specific node. This is that if one node ceases to function during a deployment, the rest of the swarm may continue uninterrupted.

4.3 Cooperative

Of course independence cannot preclude cooperation; the swarm would never function if each node acted based solely on its own abilities.

5. MAPPING

Before we can tackle the principle concern of the scouting algorithm, a consistent map system had to be developed. The Sensorfly nodes have no means of accessing external positioning networks such as GPS. A GPS system could not be relied upon when the node is conducting its mission inside of a building due to structural interference.

However the radio on a Sensorfly node is able to calculate distance between itself and another target node based on radio delay. By using multiple recordings from fixed nodes (base nodes), a node is able to triangulate its own position. If all nodes in flight (scout nodes) triangulate their coordinates relative to the same base nodes their coordinates will fall on the same grid.

For this purpose the first step in swarm deployment is the positioning of the base nodes. For accuracy Sensorfly uses a minimum of three base nodes per deployment. During initial launch one node will stay where it is and designate itself as the origin node, then one node will fly forward in a straight line (a perfect flight is not required for our map) until it reaches the edge of the target area or the edge of the radio's effective range (about 30m), the line between this node and the origin will function as the y-axis of our graph with the positive in the direction of this node. Finally one node will fly perpendicular to the y-axis until it has reached the edge of the target area or half of the radio's effective range; this line will serve as the x-axis of our graph with the positive in the direction of this node. These three nodes will communicate their distance from one another to all other nodes in the swarm to be used in the triangulation calculations. In cases where the area to search is larger than the radio's range this process is repeated, except instead of starting from an initial deployment new base nodes nominate themselves from the currently active scout nodes, and the origin lies upon one of the previous X or Y base nodes (this same node is used as the new origin, another does not need sit next to it). This map is then divided into equal sections on a coordinate grid (x, y).

6. SWARM INTELLIGENCE

With a mapping system established, the principle challenge can be confronted properly, the intelligent yet independent control of multiple nodes in a swarm. Since the goal is to scan a wide area rather then find the shortest route to a goal we will use a reverse of ant colony optimization (RACO). To maintain autonomy each node will go through four basic phases in the decision making process: target selection, report target, scan target, and accuracy decay. In addition an asynchronous task of receiving data from other nodes is in place.

6.1 Reverse Ant Colony Optimization

Normal ant colony optimization (ACO) is based on the process by which a colony of ants located the shortest route between a source of food and the home colony. Initially every ant searches in a random direction for food, all paths being perceived as equally likely to contain food at one point. When ants reach their destination (the source of food) they emit a pheromone trail during return to the home colony. When any ant detects a pheromone trail they are more likely to follow the trail rather then wonder on an unexplored path. In addition pheromone trails naturally dissipate over time. The end result is that the shortest (and most well traveled) path has the strongest concentration of pheromones at any time, and therefore all ants within the colony are likely to follow this path, reach the food, and then return to the colony, reinforcing the pheromone trail along the way.[2][3]

However Sensorfly is based on completely scanning an area, rather than finding one optimal route. Therefore we use RACO, which follows the same basic process as ACO: paths initially equal, trail left for other ants to follow, path dissipates over time, and reinforced by reuse. However in RACO, instead of being attracted, other ants are repelled by the pheromones (even their own). Of course in the case of Sensorfly the ants are really the individual helicopters and pheromones are replaced by probability tables.

6.2 Assumptions

There a few assumptions made in the following process. 1 Each section of the grid is too large to be fully scanned as a node flies past through the section on the way to its destination.

2 The target area is completely unknown before the search mission commences, no outside results are factored in.

3 Height is not a factor when searching commences, i.e. a section does not have to be scanned at the 1 meter level, then the 2 meter level, etc.

4 The target area is changing, and older data increasingly less useful as time passes.

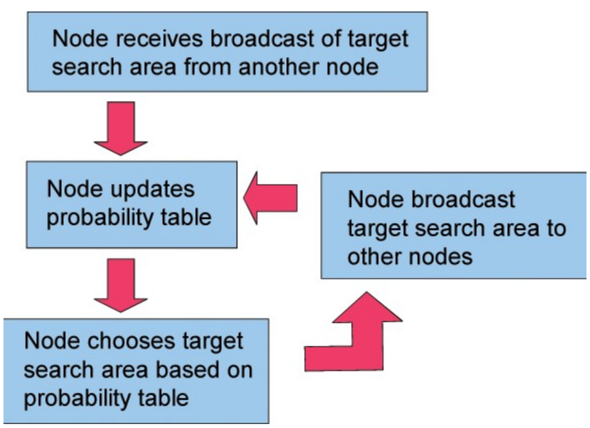

Illustration 3: A flow chart of the node's decision making process

6.3 Target Selection

Target selection is done using weighted probability tables. Each section on the map is assigned an initial score (P) for the mission (for this example we shall use a P of 50 and 8 sections in a 2 x 4 layout). The node will then calculate the sum of the P for each section within its search range to be 400. The node then picks a number randomly between 1 and 400, for this example the number is 127. The node determines that 127 represents section (1,3). In human terms, one can imagine a table built with boxes 1-50 holding the coordinate (1,1), boxes 51-100 holding (1,2) etc. then one box being selected at random. The node then decreases the score of the section to be searched (1,3) to deter (but not prevent) this node from re-searching the section again, for this example we will decrease the score by 10 to 40.

6.4 Report Target

Before moving to the target destination the node broadcasts a short message to every other node in the swarm. This message is composed of three parts. The first is a designation number used to uniquely identify each node. The second is a type designation, used to differentiate between the different type of sensors available on each node (camera, thermal, etc.). The final part is the target area itself. This information is broadcast before the node begins its searching procedure for a few reasons, in the event a search takes along period of time, other nodes will not begin searching the same area; in case the node is damaged or destroyed the area it was traveling to is noted (so it may be avoided if it presents a constant hazard). How this data is interpreted by other nodes is outlined in section 6.7.

6.5 Scan Target

At this time the node will perform whatever tasks necessary to completely scan the section one time. The specifics of the task will vary depending on the sensor attached to the node; a node with a camera attached may have to make multiple passes over the search area, while a node with a microphone or sonar may need to hover steadily for a fixed period. The results of this scan may be reported to a human controller if desired, and to other nodes if deemed necessary. It is at this time that a node may request the assistance of another node for verification (if a microphone node finds a sound pattern resembling human speech a camera node may be requested to verify human presence) or to report potential hazards to other nodes if encountered.

6.6 Accuracy Decay

Once a node has finished searching an area it will then increase the score of all areas by its entropy rate (1 for this example). This decay is the result of the data previously collected being less useful over time, and represent the gradual decay of pheromones in ACO.

6.7 Receiving Data

When a node receives a target report from another node it decrease the score of the reported target, (1,3) in this example, by 5. If the node reporting the target is of the same type as the node receiving the message, an additional deduction will be made, by 3 for this example. The second deduction is based on the principle that a variety of data types about any section is better than a variety of sources for one data type.

6.8 Repetition

The steps outline above continue until each section in the area to be scanned has a score below a desired level of accuracy or the scanning mission is terminated (which ever comes first). The numbers used in this example are given for the sole purpose of adding some structure to an otherwise overly vague example, and are by no measure set. As test with this protocol are run it may even be found necessary to allow for adjustments on a per mission basis.

6.9 A Single Cycle Numerically

It may be easier for some readers to follow this process numerically to see the end result.

From the perspective of node 1:

P[(1,1)-(2,4)]=50

Probability Table (PTable) is built from P array

Rand(400)=127

PTable[127]=(1,3)

Transmit (#1 – Type – Destination (1,3))

Move to Area (1,3)

Scan Area (1,3)

P[(1,3)]-=10
Reduce the probability of scanning (1,3) again by 10

P[(1,1)-(2,4)]+=1
Increase probability of Scanning any area by 1

7. Future Enhancements

The swarm control method outlined in section 6 will continue to be further enhanced as necessary. There are some future enhancements currently being investigated further feasibility and overall benefit to control.

7.1 Proximity Influenced Selection

Currently, proximity plays no role in determining which section will be used as the next target area; this can lead to a suboptimal search route where nodes begin by scanning the far corners then move inwards to the central location. Sensorfly may be able to reach better search times by temporarily increasing the score of adjacent sections each time a target decision must be made, however steps must be taken to avoid the nodes opting solely for the closest section and ignoring further destinations that are in greater need of searching.

7.2 Partial Data Collection

Nodes currently disregard data collected en-route to a target search area. A node may in-fact completely search one section (depending on sensor) while routing between its neighbors, but this unsolicited search has no effect on the section's score.

7.3 Hazardous Terrain

Nodes need the ability to determine whether another node has been rendered inoperable after it fails to report new target for so many cycles.

8. CONCLUSION

Sensorfly is still in development, and it remains to be seen if the swarm control method proposed in this paper is the one ultimately used in the final product. The swarm control method proposed seeks to meet all the goals of the Sensorfly project, and should prove customizable and expandable if ultimately adopted.

9. REFERENCES

[1]Aveek Purohit and Pei Zheng. Sensorfly: a controlled-mobile aerial sensor network. In ACM Conference on Embedded Networked Sensor Systems, pages 327–328, 2009.
[2]Marco Dorigo, Lica Marra Gambadella, Mauro, Birattari, Alcherio Martinoli, Riccardo Poli, Thomas Stützle. Ant Colony Optimization and Swarm Intelligence
[3]James Kennedy, Russel C Eberhart. Swarm Intelligence