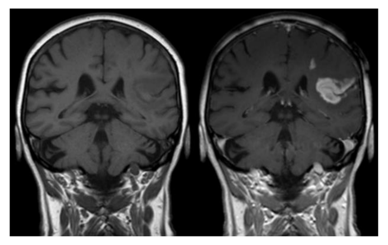
Mapping With Uniformly Controlled Stochastic Swarms

by Adi Pasic and Aaron Becker, Electrical and Computer Engineering

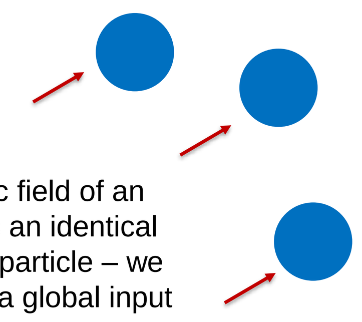
UNIVERSITY of HOUSTON

Motivation

- Swarms can be used for targeted drug delivery and vascular mapping
- We want to use magnetic swarms instead of contrast agents

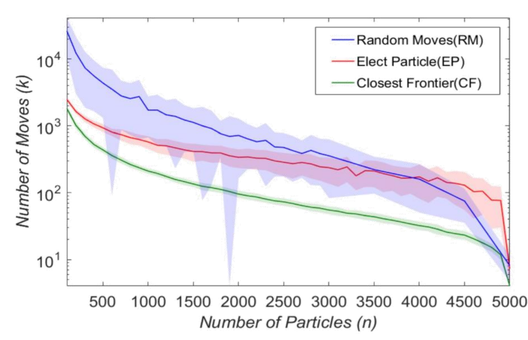


Contrast agents are useful for identifying tumors within an MRI



The magnetic field of an MRI produces an identical force on each particle – we refer to this as a global input

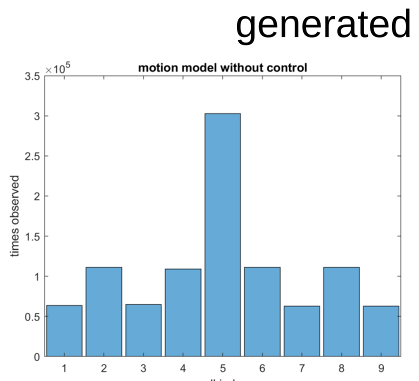
- Particles are too small to house onboard computation, so maps are generated based on the swarm's position
- The swarm is steered to a region of interest using a magnetic field – this requires efficient algorithms
- We want to compare algorithms' efficiency when the particles move randomly



Previous work compared algorithms assuming non-random motion

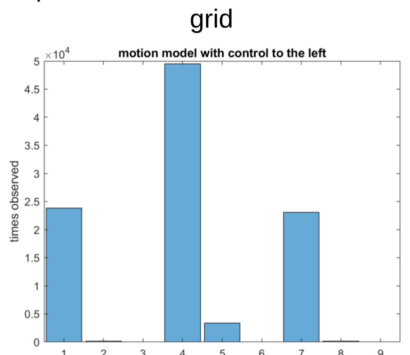
Methodology

We modeled the Brownian motion of a particle suspended in fluid and generated a probability map of its future locations

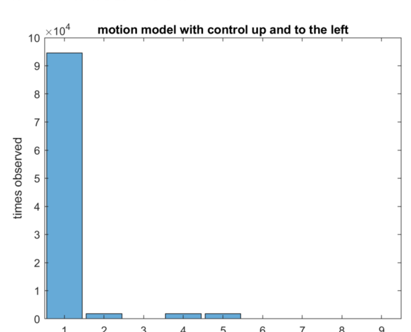


1 million recorded positions of a particle undergoing Brownian motion with no external force

Cell index represents where the particle was observed in a 3x3 grid



100 thousand recorded positions of a particle with external force to the left



100 thousand recorded positions of a particle with external force up and to the left

3/40	1/10	3/40
1/10	3/10	1/10
3/40	1/10	3/40

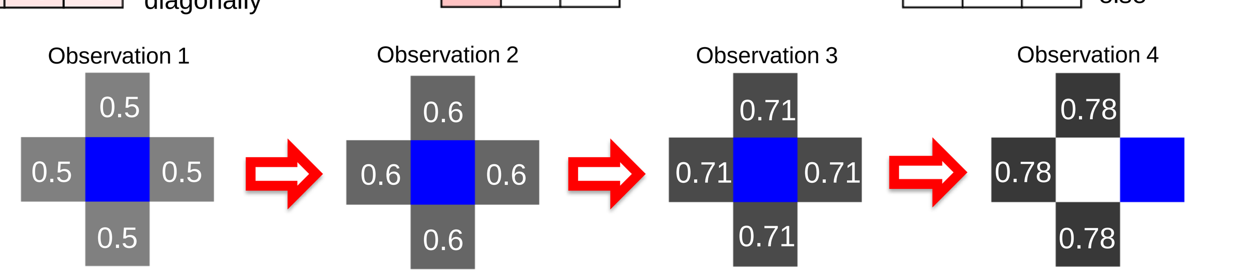
If no external force is present, the particle has a higher tendency to move straight than diagonally

117/500	1/1000	0
1/2	3/100	0
117/500	1/1000	0

If the applied force is to the left, the particle will never move right

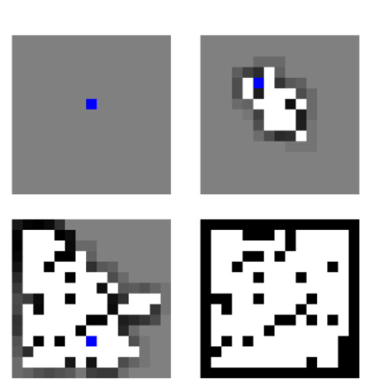
473/500	9/500	0
9/500	9/500	0
0	0	0

If we apply a force to the top left, the particle will almost never move anywhere else



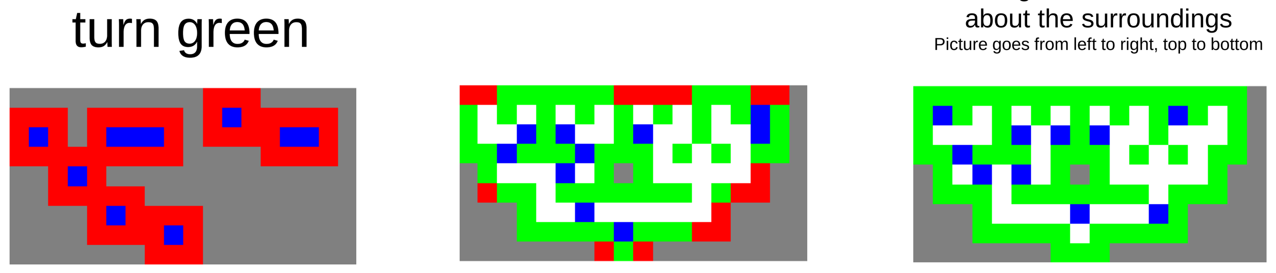
- By observing the particle's motion, we can update the surrounding cells' probabilities and shade them accordingly.
- **Darker** cells are more likely to be obstacles. Any cell that the particle is observed in is treated as free and colored white (P = 0)

- As the swarm is steered, particles discover unexplored cells known as frontiers.
- All frontiers remain red until their probability passes a minimum uncertainty threshold, at which point they become boundaries and turn green



Through continued observation, we can gain more information about the surroundings

Picture goes from left to right, top to bottom



The area is randomly filled with particles, which are then surrounded by frontiers

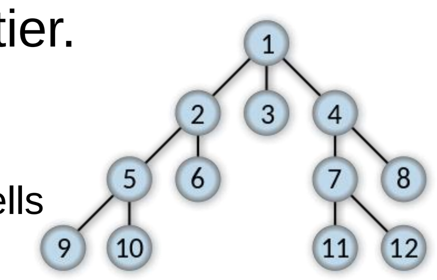
As the particles move, cells are classified as free or occupied

Until the swarm has no more frontiers to explore

Results and Explanation of Elect_Particle and Closest_Frontier Algorithms

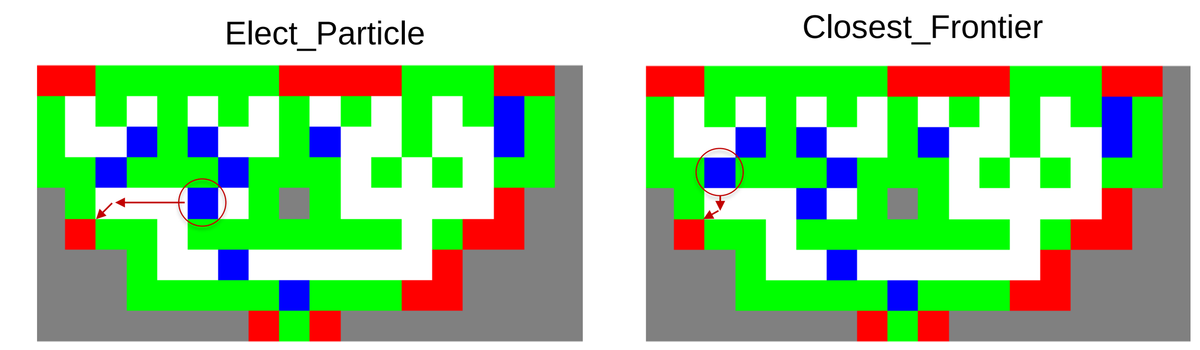
The Closest_Frontier and Elect_Particle both use breadth-first search (BFS) algorithm to find the shortest distance to a frontier.

Breadth first search finds the quickest path between two nodes. For our purposes, the robots and the frontier cells represent the start and end nodes



Elect_Particle finds the shortest path from a chosen particle to a frontier.

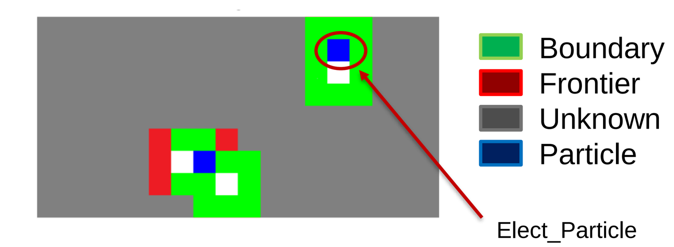
Closest_Frontier finds the shortest path from all particles to a frontier



4 moves required

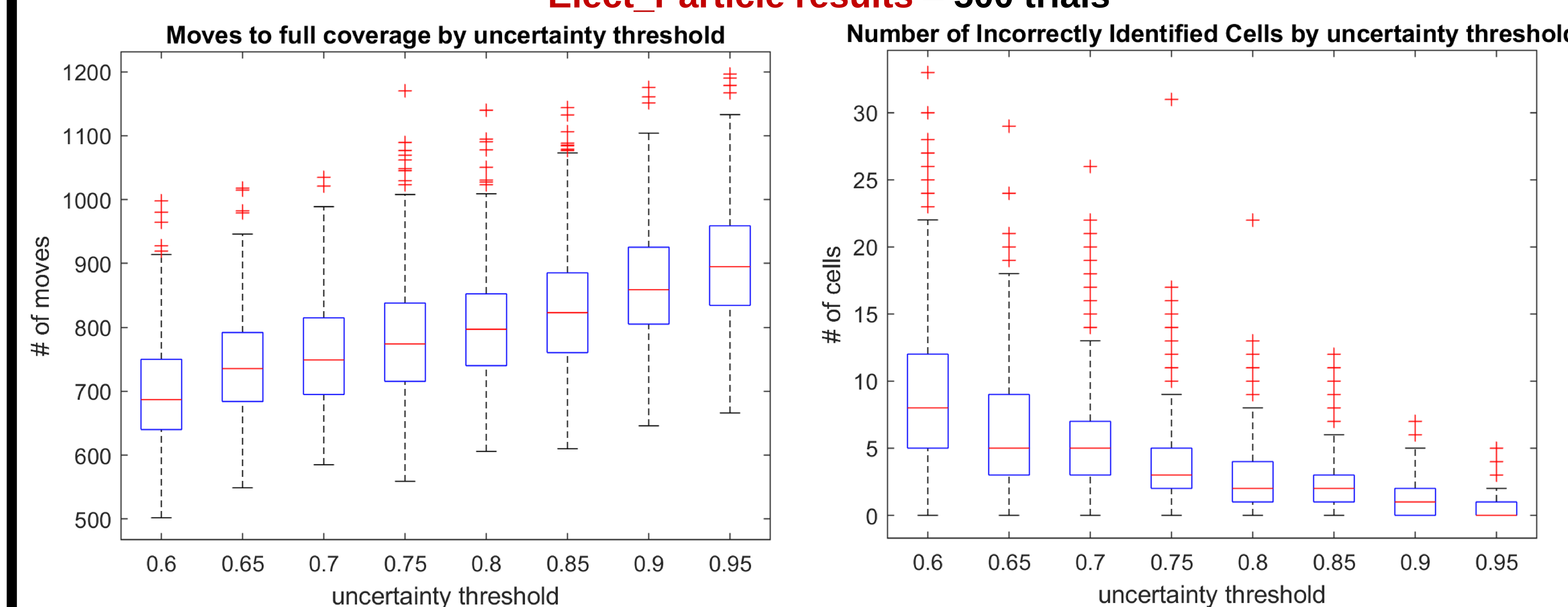
2 moves required

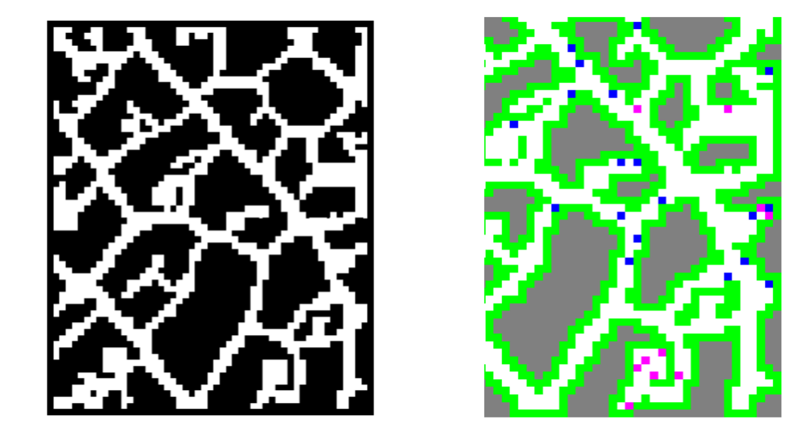
The ElectParticle algorithm had a *high tendency to fail at low uncertainty thresholds*, so the elected particle was switched when there were no frontiers near the current elected particle



The circled particle has become "trapped" and has nowhere to explore. At this point, the elected particle is randomly selected from the remaining particles

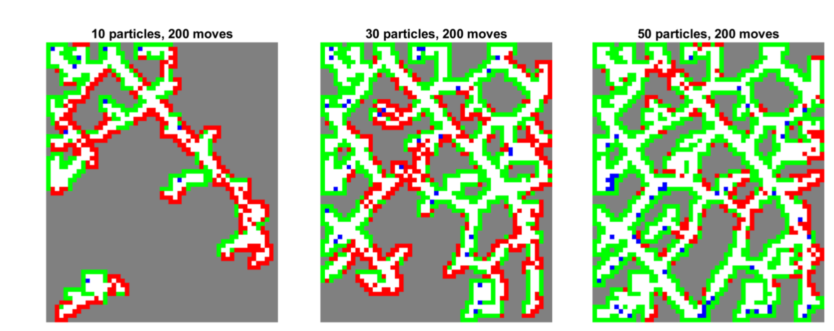
Elect_Particle results – 500 trials





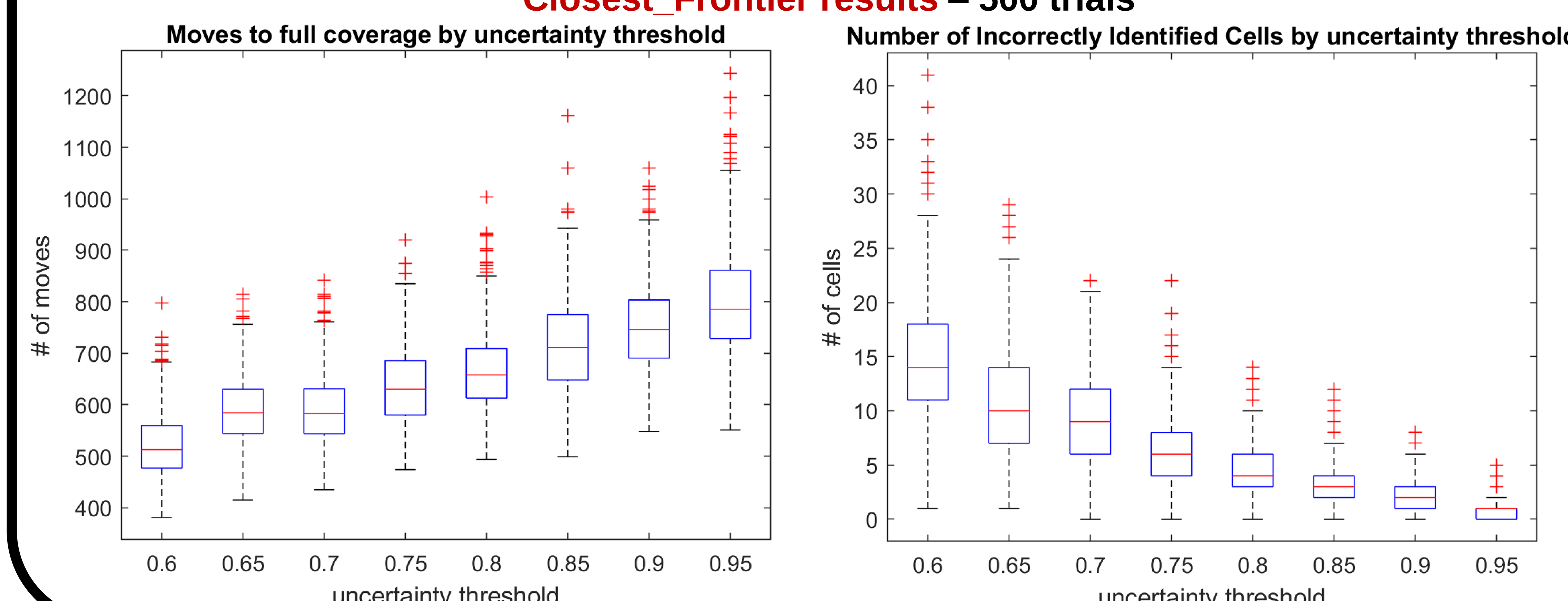
Leaf vein map used to test algorithms

Cells in purple are misclassified: 0.6 uncertainty threshold



As swarm population grows, moves required to full coverage decrease – our simulations used a population of 30 particles

Closest_Frontier results – 500 trials



- Closest_Frontier was faster than Elect_Particle for all uncertainty thresholds
- Closest_Frontier slowed down at a faster rate than Elect_Particle as uncertainty threshold increased
- Closest_Frontier had more errors than Elect_Particle for nearly all uncertainty thresholds

Conclusions and Next Steps

- Bayesian mapping using a uniformly controlled swarm can be expedited with algorithmic inputs, but efficiency remains inconsistent
- Accuracy of both algorithms improved as the certainty threshold was increased
- ClosestFrontier was more consistent than ElectParticle at low certainty thresholds and had faster times to full coverage overall but had a greater number of errors than ElectParticle at most uncertainty thresholds
- Motion models need to be verified experimentally using a magnetic setup
- Continuous workspace simulations require more mathematical analysis of Brownian motion in the presence of obstacles

Acknowledgments

University of Houston Office of Undergraduate Research
This work was supported by National Science Foundation Grant 1646607

References

A. Mahadev, D. Krupke, S. P. Fekete, and A. T. Becker, "Mapping, foraging, and coverage with a particle swarm controlled by uniform inputs," International Conference on Intelligent Robots and Systems, 2017.