Risk Sensitive Surveillance with Optimal Sensor Quality for Distributed Robotic Systems

Alexander Wallar

University of St Andrews

aw204@st-andrews.ac.uk

July 31, 2014

Table of contents

Goals

1. To provide persistent high quality sensory information for a given region
2. To minimize the risk of capture or damage to the acquiring agent
3. To allow for a dynamic number of agents without significant algorithmic change

Motivation

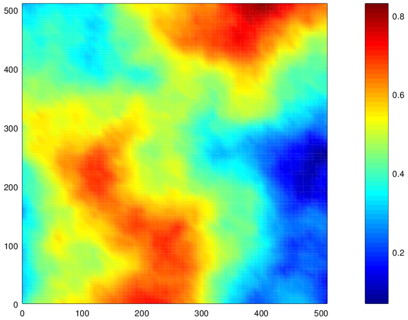
- There are many cases where high quality sensory information is needed from areas where risk of agent damage is high
- *Example*: disaster relief, covert surveillance, safety critical missions, search & rescue. . .

What is Risk?

- Risk is a representation of how dangerous or hostile a given area is
- We should minimize the frequency of data collection in areas or risk
- *Examples*:
  - Detection by a hostile target
  - Damage to the vehicle due to fire
  - Wind speeds
  - Uncertainty of *a priori* regional intelligence

What is Risk?

- More formally 2D risk is defined as a function, $R_0 : \mathbb{R}^2 \to [0, 1]$
- Risk in 3D is the just a exponential decay of the initial risk but also parametrized by the altitude, $R : \mathbb{R}^3 \to [0, 1]$



How to Mitigate Potential Risk?

- Minimize frequency of surveillance for regions of high risk
- Increase the altitude in order to maintain high absolute distance from a ground level risk area
- The policies will change depending on the type of risk

What is Sensor Quality

- In our experiments, the sensor quality was the quality of information being retrieved from a camera
- Many aspects about the state of the robot affect this metric
- We focused on the altitude and motion blur
- However, different measures can be used be for sensor quality

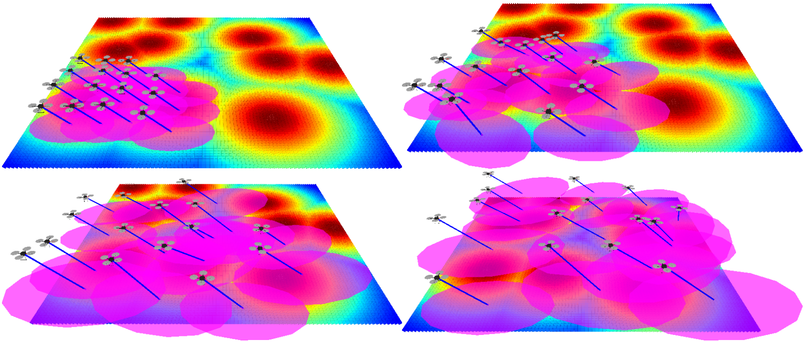
Motion Blur

- Increases as the speed of the vehicle increases
- Can be sensed at each iteration
- Need to determine a optimal velocity that minimizes the motion blur but still does not bring the robot to a standstill

Altitude

- The altitude determines the distance away from the ground that images are being captured from
- Too high, and objects in the image are small to be recognized
- Too low, and objects in the image are indiscernible

Proposed Approach



Proposed Approach

- We break the planning problem into two parts
  1. Plan for 2D velocity
  2. Plan for the altitude
- For 2D planning
  - Move to regions of the map, M, that have a combination largest uncertainty, Υ, and the lowest risk, R
  - Decrease the speed to minimize motion blur whilst maximizing the information capture rate
- For altitude planning
  - Determine the altitude by maximizing the sensor quality whilst minimizing the risk

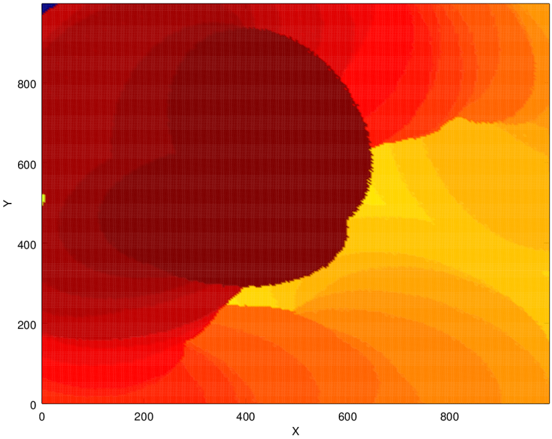
Determining Where to Move

- Our implementation uses a cost surface that is shared between the members in the swarm
- To determine a new heading:
    1. An agent samples costs from the surface around the edge of its sensor foot print,
    2. Moves in the direction of the smallest cost,
    3. And updates the uncertainty measurements in the cost surface for the area that sensor foot print just covered

Cost Surface & Uncertainty

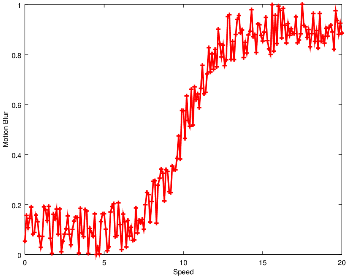
- The cost surface, Γ, is made up of a combination of of the uncertainty measurements and the risk surface of the region
- More formally, $\Gamma(x, y) = \delta(\Upsilon(x, y), R(x, y))$
- Where $\delta : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a user defined function that combines the uncertainty and risk into one metric

Cost Surface & Uncertainty



Determining the Speed

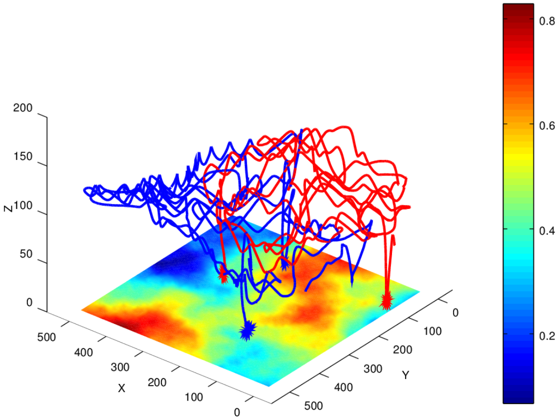
- An optimal speed can be achieved by maximizing the information capture rate, ICR, whilst minimizing the motion blur, MB
- Formally, we maximize $ICR(v) - MB(v)$



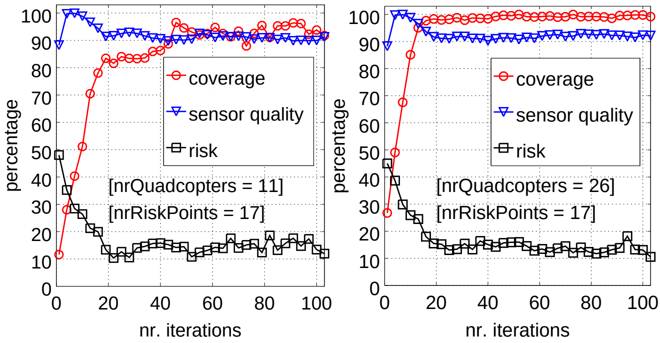
Determining the Altitude

- To determine the desired altitude, we must maximize the sensor quality whilst minimizing the risk with respect to z
- $Alt(x, y) = \arg\max_{z} SQ(z) - R(x, y, z)$
- This means that we are avoiding the risk by increasing the altitude, but are limiting the altitude by the decay in the sensor quality

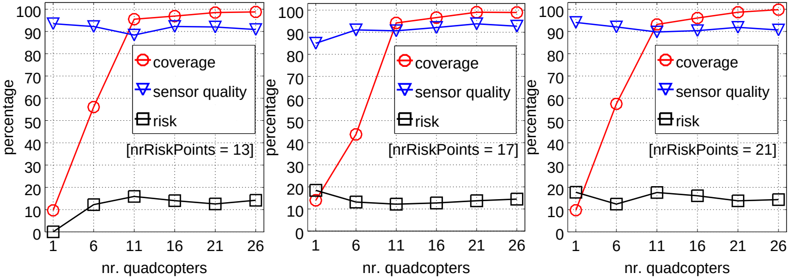
Trajectories



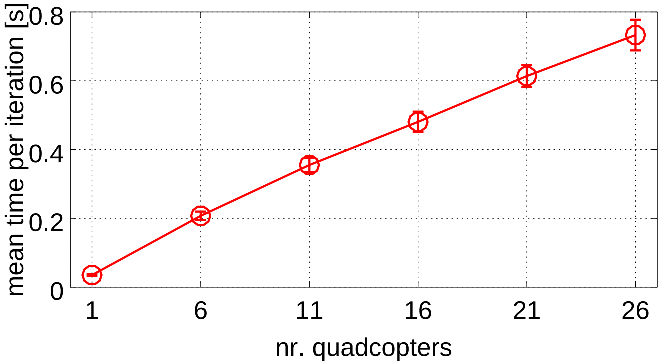
Instance Results



Quadrotors


percentage
100
90
80
70
60
50
40
30
20
10
0
1
6
11
16
21
26
nr. quadcopters
coverage
sensor quality
risk
[nrRiskPoints = 13]
[nrRiskPoints = 17]
[nrRiskPoints = 21]

Timing


mean time per iteration [s]
0.8
0.6
0.4
0.2
0
1
6
11
16
21
26
nr. quadcopters

Conclusions

- The proposed approach minimizes the measured risk whilst maximizing the sensor quality by varying the altitude and reducing the frequency of coverage for risky regions
- The algorithmic time increases linearly with respect to the number of quadrotors
- The quadrotors are quick to reach area coverage convergence