1 Risk Model

Risk is modelled as a function, $\mathbb{R}^3 \rightarrow [0,1]$ which is use to determine the risk at every (x, y, z) location within the search space. In order to determine risk in 3-space, we obtain a 2-D risk model which models the risk at the minimum altitude. This function $R_0 : \mathbb{R}^2 \rightarrow [0,1]$ determines what the risk would be at the minimum altitude. This is assumed to be given to the algorithm *a priori* or can be determined at any time during the iteration of the algorithm. This ground risk, R_0, is modelled as a lookup table rather than a combination of basis functions in order to give a more generic model for risk that can be used in any use case of the algorithm. To determine the 3-D risk, $R(x, y, z)$, we perform an exponential decay on the given ground risk value, $R_0(x, y)$. 3D risk is defined as follows:

$$R(x, y, z) = R_0(x, y) \cdot \exp\left(-\frac{z^2}{K \cdot R_0(x, y)^2}\right)$$

Even though risk in 3D is evaluated and is not simply a lookup table, one can be used instead. This representation for 3D risk is ideal for modelling fires, detection by hostile agents, or any stimuli that would decrease monotonically as the altitude increases.

For the experiments, we have used nine different scenes for the representation of the ground risk, R_0. To generate these scenes, we used the diamond-square algorithm [1] to generate random terrain maps that have values from zero to one. The diamond-square algorithm generates realistic random risk scenes that represent the 2D ground level risk as anticipated. Random terrain maps have also been used in [2] to represent random risk for path planning problems. A heat-map showing a random terrain map generated with the diamond square algorithm is shown in Fig 1.

2 2D Planning

Quadrotors use a very simple, reactive rule to determine where to move that has extremely emergent properties. Given a map, M, quadrotors move to areas with a combination of the largest uncertainty, Υ, and the lowest risk, R. Our implementation uses a cost surface that is shared between members of the swarm. Since the quadrotors update this cost surface as they move around it, the cost surface is used as the mode of communication between the quadrotors. This cost surface removes the need for the agents to have any peer to peer communication and also removes the need for any agent to have perfect information about the swarm.

For a quadrotor to determine a new heading, the agent samples costs from the surface around the edge of its sensor foot print. It then moves in the direction of the smallest cost. One it has moved, the agent updates the measurements for the uncertainty grid, Υ for the area that was just covered in by the sensor foot print. This uncertainty update happens iteratively in the swarm, so the

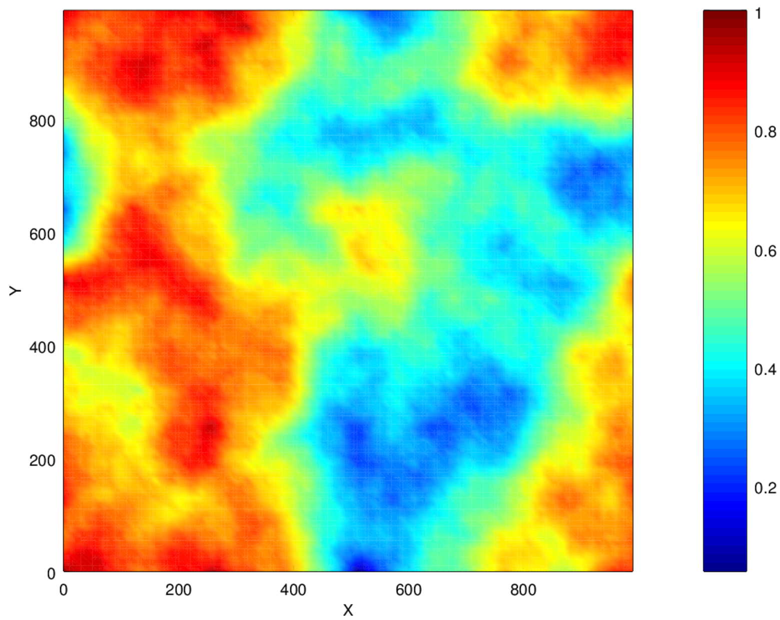

Figure 1: A visualization of the ground risk, R_0

quadrotors left the plan their movements have an updated grid before they determine a new heading.

The cost surface, Γ, is made up of a combination of the uncertainty grid, Υ and the risk surface of the region. More formally, assume that there is a function $\delta : \mathbb{R}^2 \rightarrow \mathbb{R}$ which is used to combine the uncertainty and risk into one metric. This is assumed to be a user defined function which may change from application to application. The cost surface is then defined as,

$$\Gamma(x, y) = \delta(\Upsilon(x, y), R(x, y))$$

For our implementation we defined δ as,

$$\delta(\upsilon, r) = \max \Upsilon - \upsilon + 100 \cdot r$$

An instance of the uncertainty surface, Υ, is shown in Fig 2 where the x, y coordinate represents the 2D position within the workspace and the color represents the level of uncertainty. Notice the two dark blue ellipses with low uncertainty. That is the current sensor footprint of the quadrotors at the time at which the snapshot was taken. Fig. 3 shows a snapshot of the cost surface which is created by combining the risk surface shown in Fig. 1 and the uncertainty surface shown in Fig. 2 using the δ combination function used in our implementation.

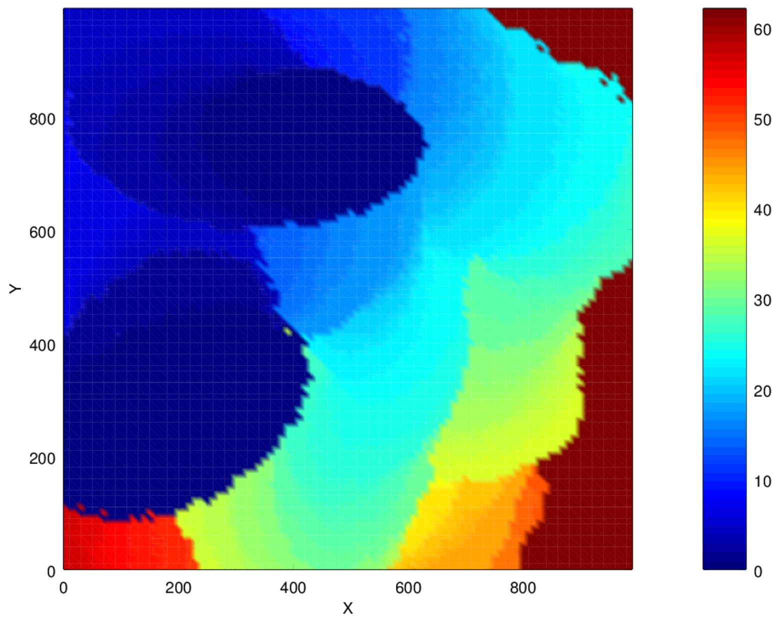

Figure 2: Uncertainty surface where red represents higher uncertainty

References

[1] A. Fournier, D. S. Fussell, and L. C. Carpenter, "Computer rendering of stochastic models," *Commun. ACM*, vol. 25, no. 6, pp. 371–384, 1982.

[2] L. Murphy and P. Newman, "Risky planning: Path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff." in *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011.* IEEE, 2011, pp. 3727–3732. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2011.5980124

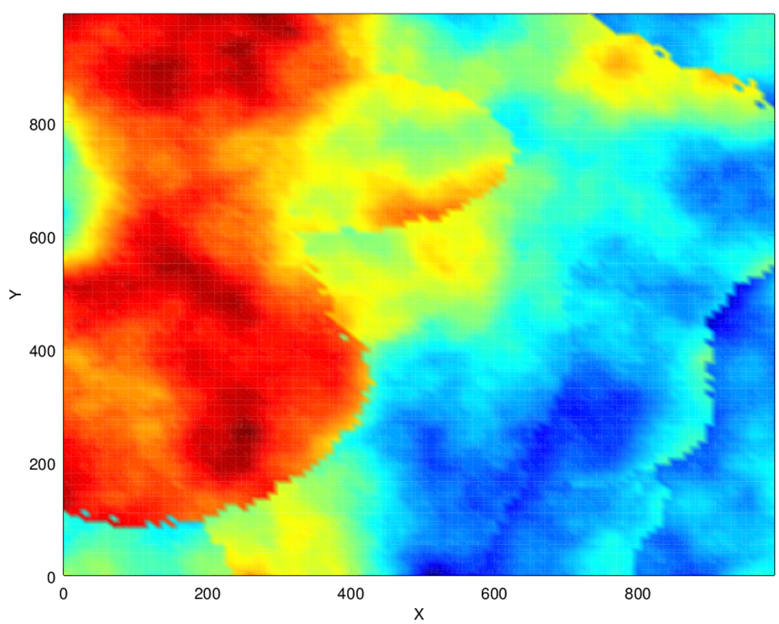

Figure 3: Cost surface which is a combination of the uncertainty surface in Fig. 2 and Fig. 1 where red represents higher cost