Using entropy to drive search in occupancy grids

Zeyn Saigol

IR Lab, School of Computer Science

University of Birmingham

2nd February 2010

Outline

1. Introduction
2. Problem model and occupancy grids
3. Entropy
4. Infotaxis
5. Robotic mapping
6. Frat-house algorithms ($\Sigma\Delta H$)
7. Conclusions

Project Overview - Vents

- Hydrothermal vent
- Ocean-floor outgassing of mineral-rich water, 300+°C
- Rare and interesting!

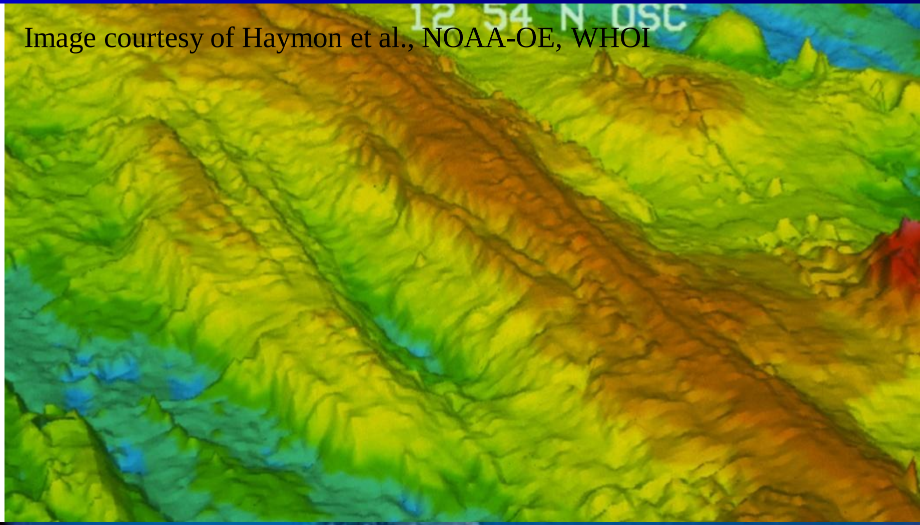


Image courtesy of Haymon et al., NOAA-OE, WHOI

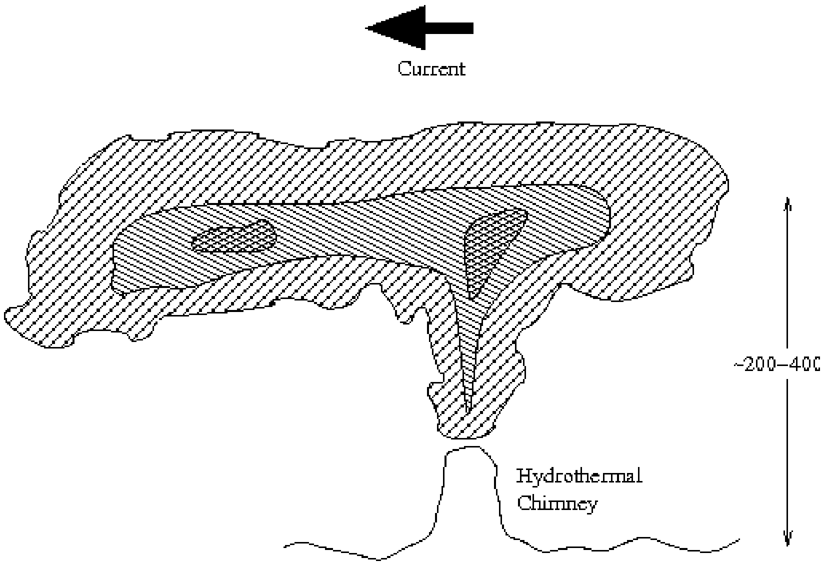
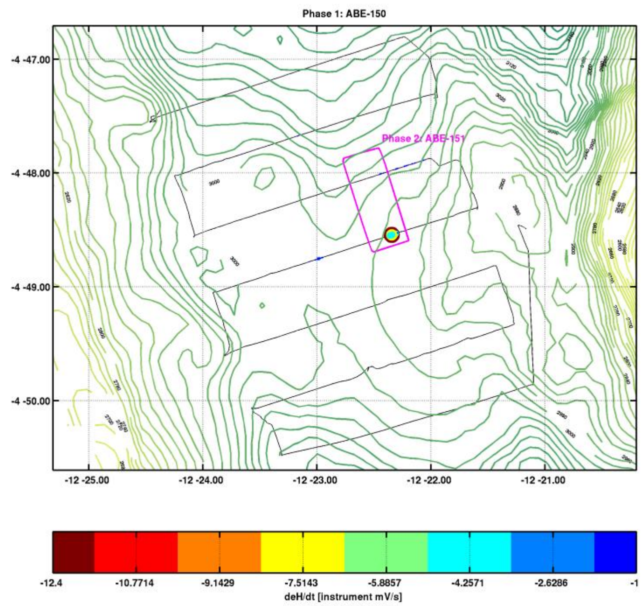


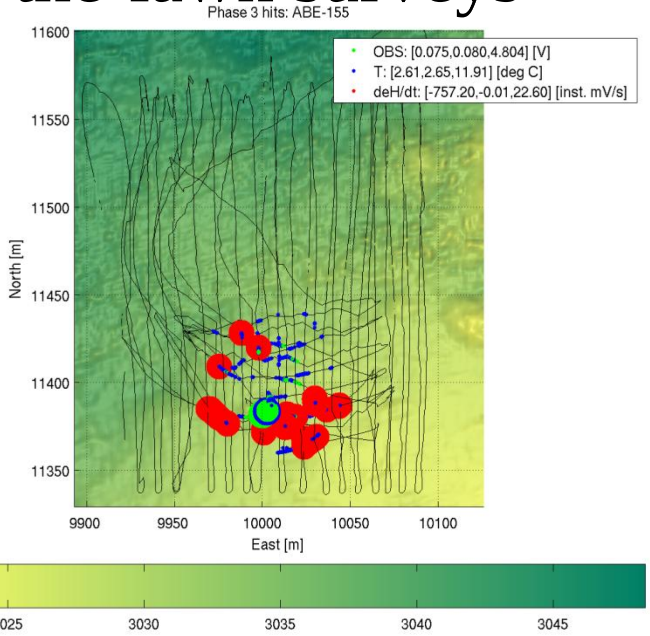

Photo by Meg Tivey, WHOI

Project Overview – Vent Prospecting

a) Ship-based observations

b) AUV in decreasing-size mow-the-lawn surveys





Project Overview – Prospecting with AUVs

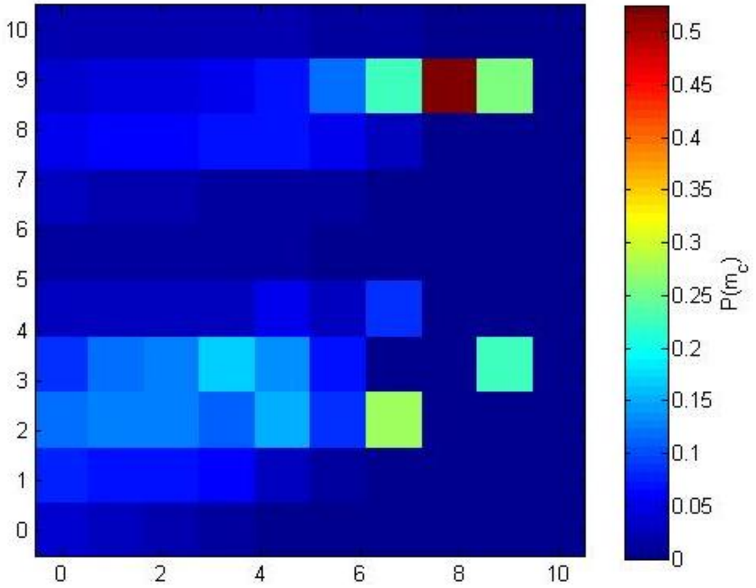
- Aim is to optimise AUV missions by performing on-board data analysis and path planning.
- Target for missions is to find as many vents as possible, given time/battery life available.

- Can decompose into:
  - Mapping – create a map of likely vent locations from sensor data.
  - Plan best path, given map.

Model - Occupancy Grids

- Adopted mapping algorithm due to Mike Jakuba (2007), producing an occupancy grid (OG).
- Developed to create hydrothermal vent maps from AUV data.

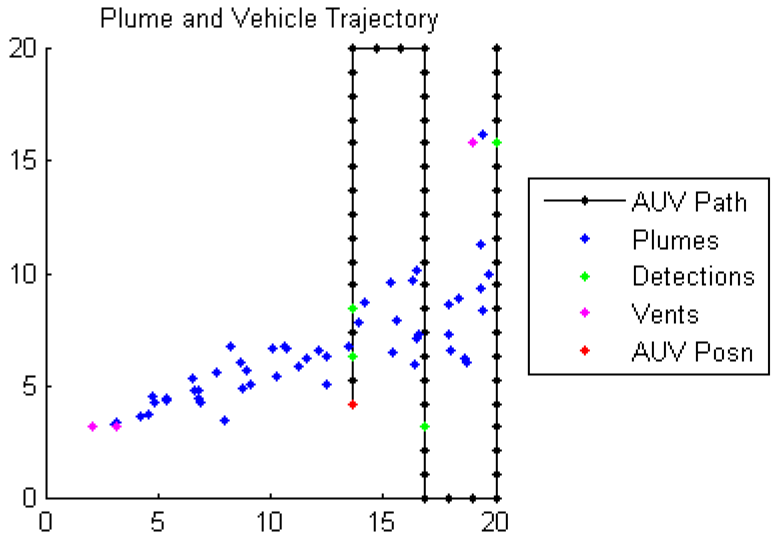
Photo credit Tom Kleindinst, WHOI



- Divide search area into grid of cells that can either be occupied (contain a vent) or empty.
- OG specifies probability of occupancy of each cell, $P(m_c)$

Environment Simulation

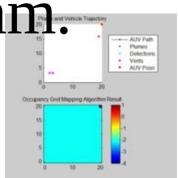
- Purely 2-D environment.
- Agent constrained to move one cell/timestep, N/E/S/W.
- Each vent emits 1 plume particle/timestep, dispersed by current plus Gaussian noise.



- Possible observations are:

$$z = \begin{cases} l, \text{locate a vent} \\ p, \text{detect a plume} \\ n, \text{detect nothing} \end{cases}$$

- The observation z (plus vehicle position, current, and the previous OG) provide the input to the mapping algorithm.
- MTL provides the baseline performance comparison

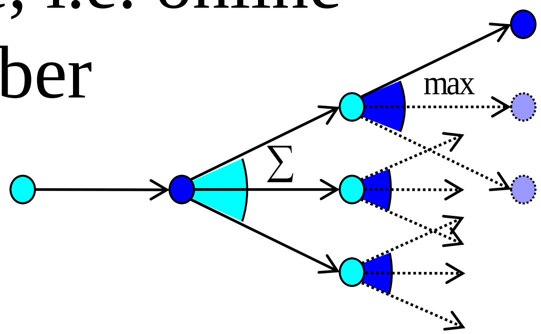


Observation Model

- Can calculate probability of a detection from a given vent using the environment simulation.
- From this we developed a model of $P(z|\boldsymbol{O})$, the probability of each of the observations $\{l,p,n\}$ given the current occupancy grid $\boldsymbol{O}$.
- (details omitted).

Story So Far…

- Previously, set up problem as a POMDP:
  - States, actions, rewards (=1 for each vent found)
  - Observation model, and belief state *b*={OG,AUV-posn}
- Information-Lookahead algorithm: solve by forward search through belief space, i.e. online POMDP method, for a limited number of steps.

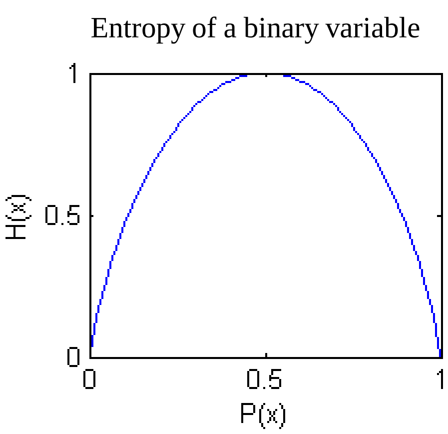


- New approach desired!
  - 25 mins run-time for 4 steps (IL-4)
  - 4 steps, discount factor 0.9 => only accounting for 35% of the potential value function

Entropy of Maps

- Entropy is a measure of the information content of a random variable – higher entropy means the variable carries less information.
- For discrete variable, entropy is defined by

$$H(X) = -\sum_x P(x)\log_2 P(x)$$



- If we encode the location of a source as a 2-D Gaussian, the entropy of the distribution represents how certain we are of the source location – as we gain information, the entropy will decrease.

Infotaxis

- Infotaxis (Vergassola et al (2007)) is an algorithm for locating a chemical source in a turbulent current, i.e. without a consistent chemical gradient leading to the source.
- Use a very similar environment model to ours, but maintain a continuous distribution for the location of a single source.
- Choose adjacent location *r* to move to so as to maximise reduction in entropy:

$$\Delta S(r) = P(r)[-S] + [1 - P(r)][\rho_0(r)\Delta S_0 + \rho_1(r)\Delta S_1 + \ldots]$$

where *P(r)* is the probability of the source being at $r, \rho_X(r)$ is the probability of *X* detections at r, and ΔS_X is the corresponding change in entropy.

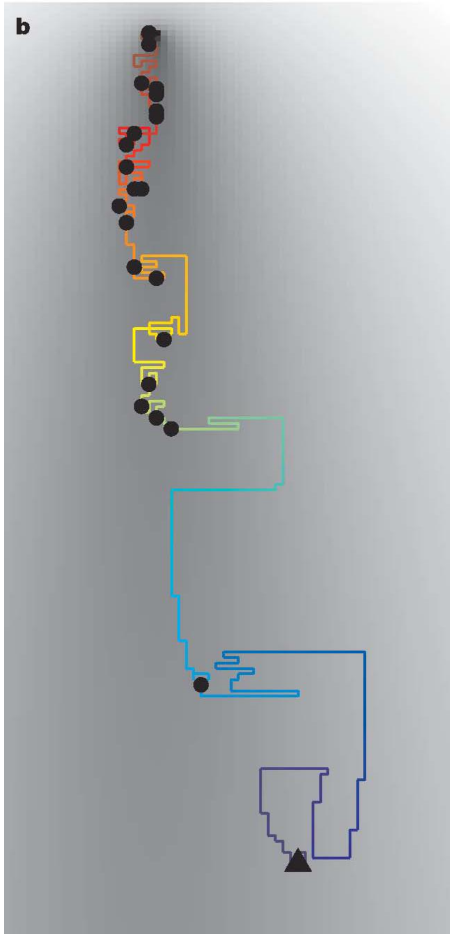


Image from Vergassola et al (2007)

Infotaxis For OGs

- Firstly, we have an occupancy grid rather than a source location distribution.
- The entropy of each cell is clearly defined as

$$H_c = -P(m_c)\log_2 P(m_c) - (1 - P(m_c))\log_2(1 - P(m_c))$$

- As the occupancy probability is independent for all cells, we assume that we can calculate a useful proxy for total map entropy by summing the entropy of all cells.
- Secondly, our observation model is different – we have 3 possible observations {*l*,*p*,*n*} instead of $\{l, \rho_0, \rho_1, \ldots\}$
- However we can still calculate the expected change in entropy:

$$E[\Delta \mathrm{H}] = \sum_c \mathrm{H}_c - \left(\sum_z P(z) \sum_c \mathrm{H}_{c|z} \right)$$

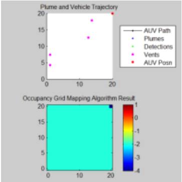
and choose the action that maximises this.

Experimental Setup

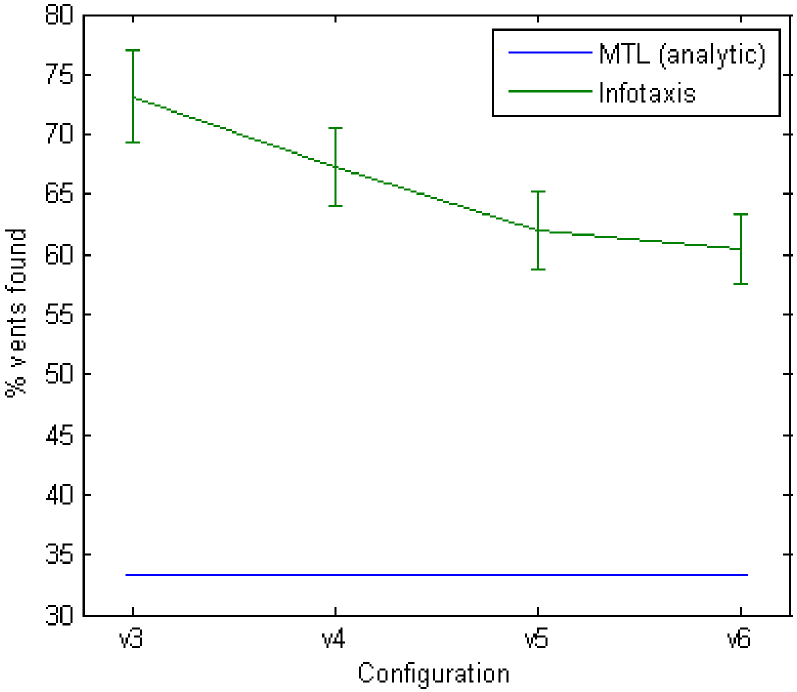
- 150 trials with different random seeds
  - Different vent locations
  - Different dispersion patterns for plumes
- Repeated with 3,4,5 and 6 vents (total 600 trials)
- Fixed vehicle starting location (down-current corner of grid)
- 20x20 grid
- Current varies according to deterministic sine-wave pattern
- Each trial is 133 time-steps (AUV covers 1/3 of grid)

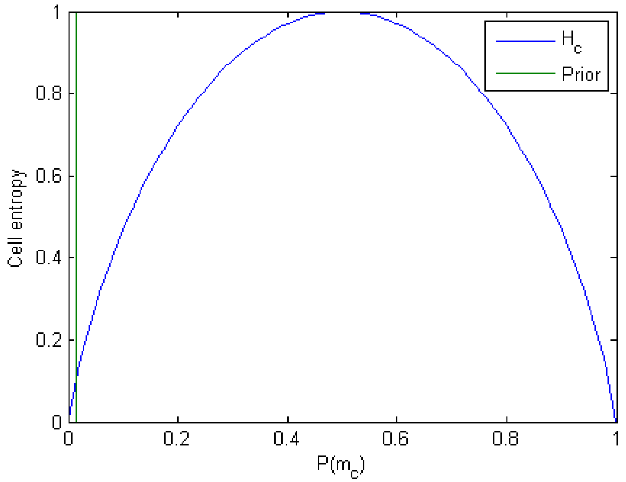
Infotaxis Results

- Results much better than MTL
- Example trial



- Behaviour slightly myopic





- Prior issue – entropy will **increase** when get a useful detection.

Robotic Mapping

- Using an indoor mobile robot to create a map has been studied widely, including work on planning a path to build an OG floor plan of a building efficiently.
- Use range-finding sensors (e.g. sonar, laser), which provide a direct (if noisy) estimate of whether cells within a “sensor cone” are occupied or empty.
- Canonical technique is described in Thrun et al (2005), based on calculating the information gain for each cell.
- Assume sensor measures occupancy correctly with probability p_{true} , then measure occupied with probability:

$$p^{+} = p_{true} p(m_c) + (1 - p_{true})(1 - p(m_c))$$

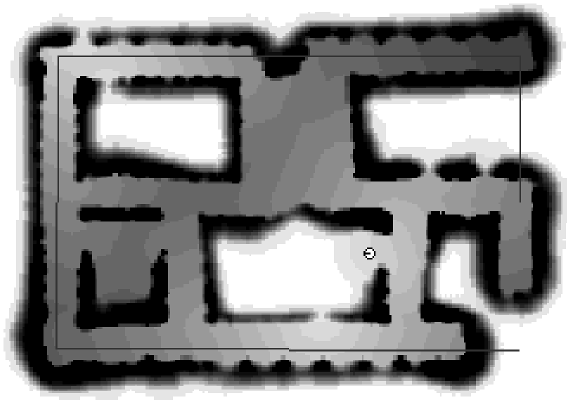
- Then expected new entropy is: $\mathrm{E}\left[\mathrm{H}_c^{'}\right] = p^{+}\mathrm{H}_c^{+} + p^{-}\mathrm{H}_c^{-}$

Robotic Mapping cont'd

- However, find that in practice it works as well to assume a binary information gain: 0 for cells that have already been measured, and 1 for un-measured cells.
- Then this information gain is propagated using a value-iteration-like algorithm:

$$V'(m_i) = \begin{cases} \max_j \left[c(m_i, m_j) + V'(m_j) \right] & \text{if } i \text{ measured} \\ 1 & \text{otherwise} \end{cases}$$

where $c(m_i,m_j)$ is the cost of moving from i to j



Frat House Algorithms

- Need to solve several problems:
  - We have long-range sensors that measure a large proportion of the search area.
  - Sensors do not provide an occupied/empty estimate.
  - Due to the small prior, entropy will often increase when we get a useful detection.
- Basic idea: maximise the change in entropy, regardless of whether it's a reduction or increase.
- So:
  - Calculate $|\Delta H_c|$ for each cell
  - Sum this value over all cells to allow for distal sensors

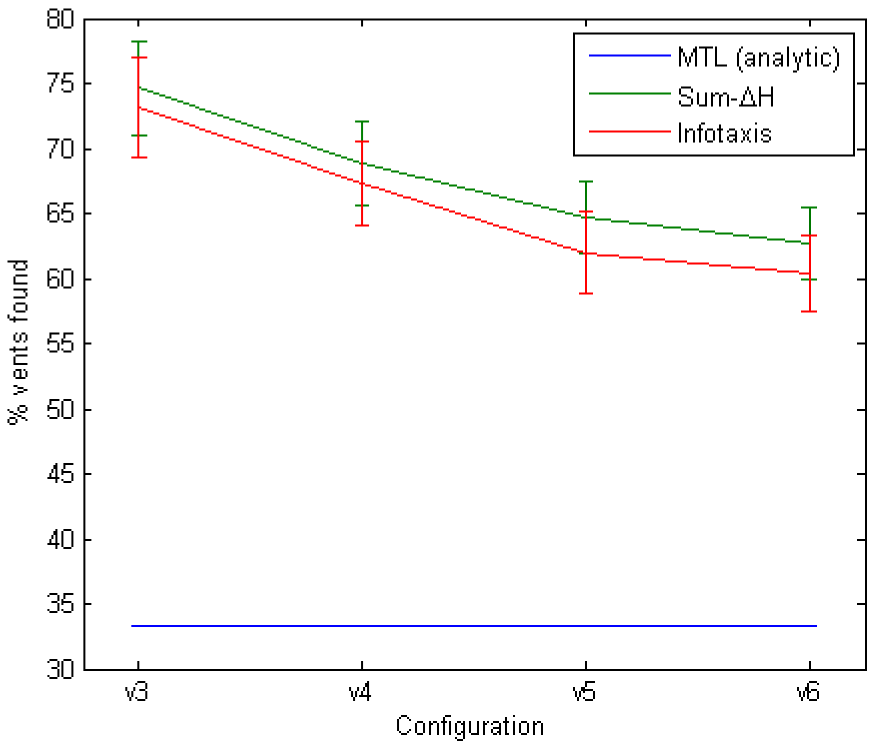
Frat House - Sum-Delta-H

- Sensors do not provide occupied/empty estimates, but we can still take an expectation over the possible observations:

$$\begin{aligned} \Sigma\Delta\mathrm{H} &= \mathrm{E}\left[\sum_c |\Delta\mathrm{H}_c|\right] \\ &= \sum_z P(z) \sum_c \left|\mathrm{H}_c - \mathrm{H}_{c|z}\right| \end{aligned}$$

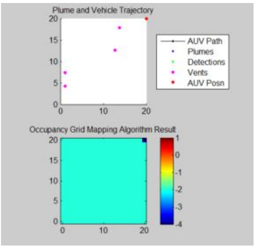
- The algorithm calculates $\Sigma\Delta\mathrm{H}$ for all available actions, and selects the action with the highest value.
- Differs from Thrun et al (2005) in that we "look ahead" one step, and take an expectation over observations; they do not do this because they select an action first, using the current map, then receive an observation and update the map, and finally perform the action selected. Made more sense to us to choose an action after receiving an observation.

Sum-Delta-H Results



- Useful improvement over Infotaxis!
- But not a completely fair comparison:
  - Prior not an issue for Vergassola et al.
  - Not calculating entropy of a *distribution*.

- $\Sigma\Delta H$ example trial



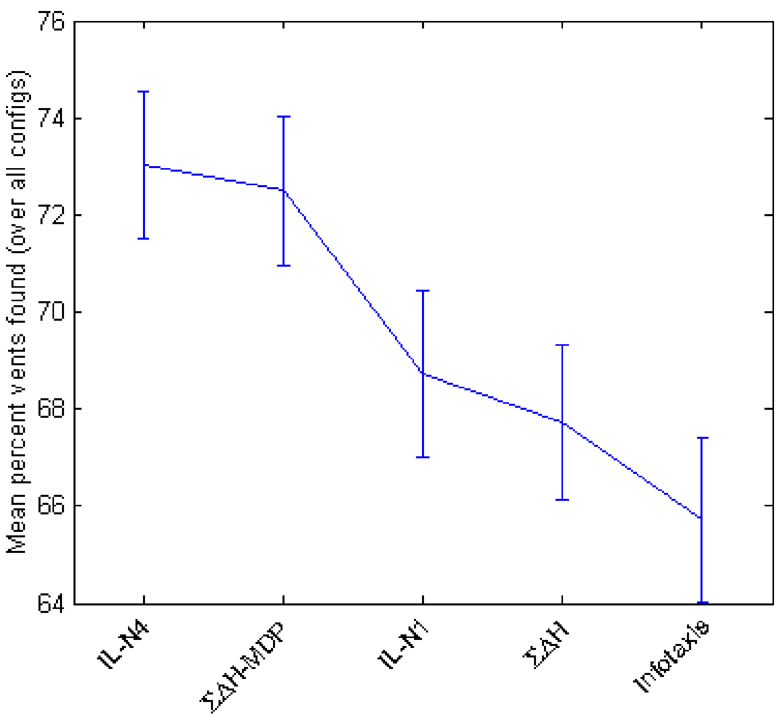
Sum-Delta-H MDP

- Wanted to avoid myopic behaviour of $\Sigma\Delta\mathrm{H}$ - it ignores potential reward more than one step away from the agent's location.
- Solve this by assuming the agent can jump to any location on the grid, and a calculate $\Sigma\Delta\mathrm{H}$ value for making an observation from there.
- Then, to allow for the cost of moving to distant cells (plus expected observations getting there), we treat the $\Sigma\Delta\mathrm{H}$ values as rewards in an MDP.
- Value iteration, using the formula

$$V'(s) = \max_a \left[\Sigma\Delta\mathrm{H}_{s'} + \gamma V(s')\right]$$

Results and Discussion

- Comparison with information-lookahead (IL): IL-N1 is statistically equivalent to ΣΔH, IL-N4 is statistically equivalent to ΣΔH-MDP



- However, ΣΔH-MDP is usefully faster than IL-N4: 5 mins instead of 25 mins per trial.
- IL scales slightly better with grid size, but scales very badly with number of steps of lookahead.

Summary

- We have created a novel algorithm to guide exploration in occupancy grids.
- This adapts existing entropy-based techniques to deal with:
  - Low prior occupancy probabilities.
  - Uncertain, long-range sensors.
- The $\Sigma\Delta H$-MDP algorithm we created outperforms traditional methods such as MTL, and performs at least as well as online POMDP methods.

References

Jakuba, M. (2007). *Stochastic Mapping for Chemical Plume Source Localization with Application to Autonomous Hydrothermal Vent Discovery*. PhD thesis, MIT and WHOI Joint Program.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'Infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409.