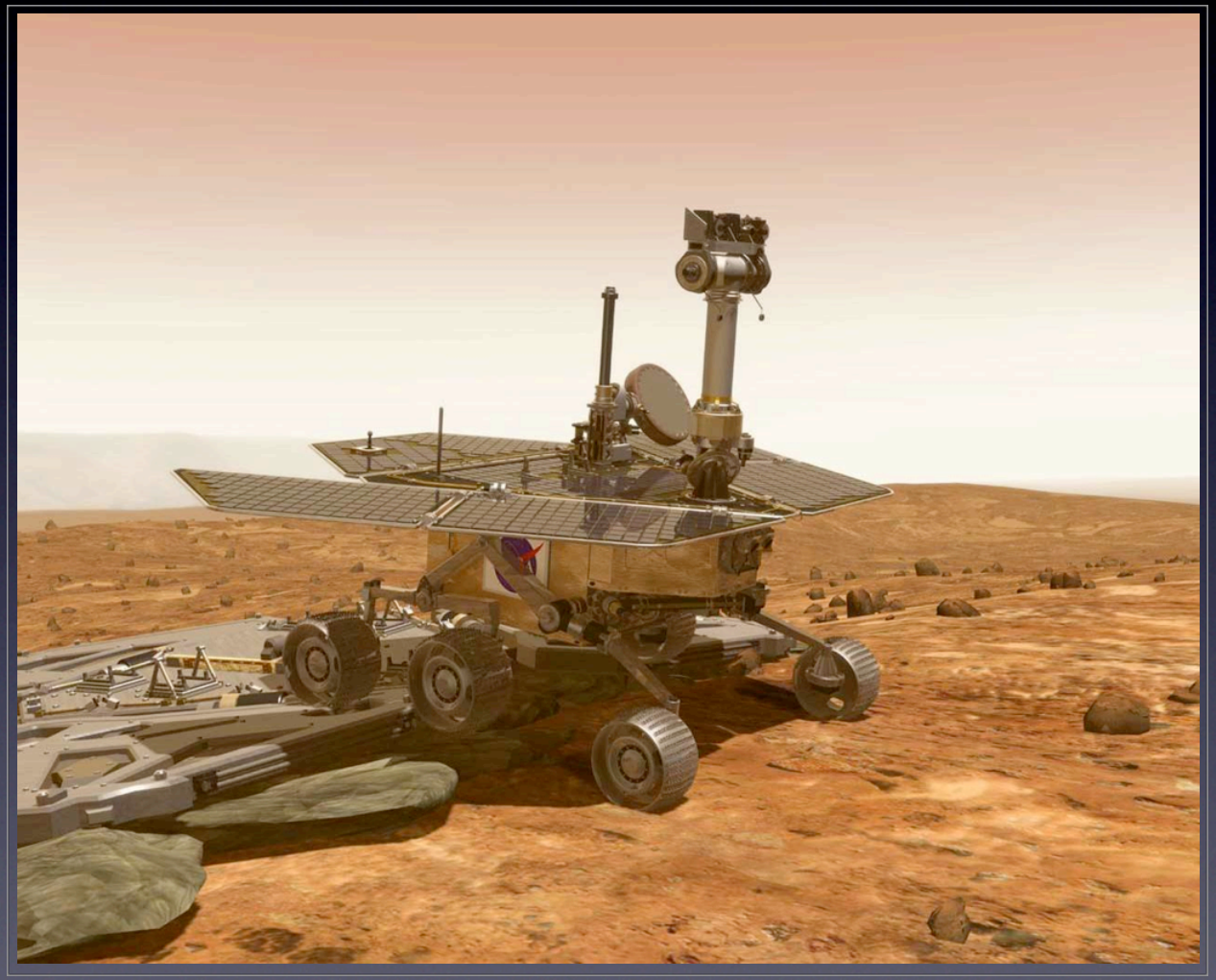
Global A-Optimal Robot Exploration in SLAM

ICRA 2005

Robert Sim and Nicholas Roy

Presented by Andy Matuschak

The Problem



The Problem

- How do we interpret our sensors' data?
- How can we avoid obstacles and hazards?
- What's the best path to explore the terrain?
- What do we do in the event of a Martian attack on our instruments?

The Problem

- How do we interpret our sensors' data?
- How can we avoid obstacles and hazards?
- What's the best path to explore the terrain?
- What do we do in the event of a Martian attack on our instruments?

The Problem

- How do we interpret our sensors' data?
- How can we avoid obstacles and hazards?
- What's the best path to explore the terrain?
  - And what does "best" mean?
- What do we do in the event of a Martian attack on our instruments?

What is "best"? Formulation

In general, a state is represented by:

$$\xi = (x, y, \theta, x_1, y_1, x_2, y_2, \ldots, x_n, y_n).$$

Where the features in the environment are at:

$$(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$$

And the robot's position is represented by:

$$(x, y, \theta)$$

What is "best"? Formulation

The robot takes some action at every step:

$$u_t = (\Delta d, \Delta\theta)$$

And takes range and bearing measurements:

$$z_t = (r_1, b_1, r_2, b_2, \ldots r_n, b_n)$$

But the measurements are noisy! We need:

$$p(\xi | z_t, u_t, z_{t-1}, u_{t-1}, \ldots, z_0, u_0)$$

What is "best"? Using SLAM

- SLAM gives the posterior state distribution
  - ("**S**imultaneous **L**ocation **a**nd **M**apping")
- Assumes noise is Gaussian
- Makes increasingly better state estimates
- Produces: $p(\xi) = N(\mu, \Psi)$

What is "best"?
Entropic Analysis

- We need something to minimize!
- How much does each move help?
- We can try analyzing the system's entropy.

What is "best"? Entropic Analysis

- Entropy of a distribution: $H(p(\xi)) = \int_{\Xi} p(\xi) \log p(\xi)$
- Relative entropy after taking some action:

$$\Delta I_{t+1|t} = (E_z \left[H(p(\xi|a,z)) \right] - H(p(\xi)))$$

- Now, if we have *d* prior components:

$$\begin{aligned} p(\xi) &= k \exp\left\{ -\frac{1}{2} (\Delta\xi)^T \Psi^{-1} (\Delta\xi) \right\} \\ \Rightarrow H(p(\xi)) &= \frac{d}{2}(1 + \log 2\pi) + \frac{1}{2} \log\det(\Psi) \\ \Rightarrow \Delta I_{t+1|t} &\propto E_z \left[\log\det(\Psi_{t+1}) \right] - \log\det(\Psi_t) \end{aligned}$$

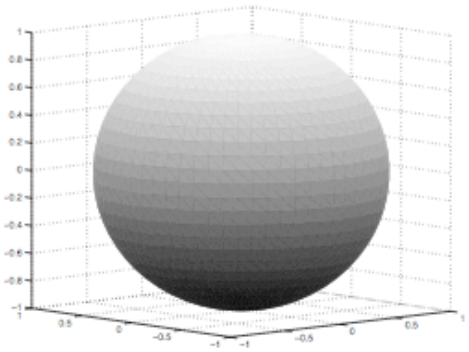
- Where $k = (2\pi)^{-d/2} \det(\Psi)^{-1/2}$ and $\Delta\xi = \xi - \mu.$

What is "best"? D-Optimal

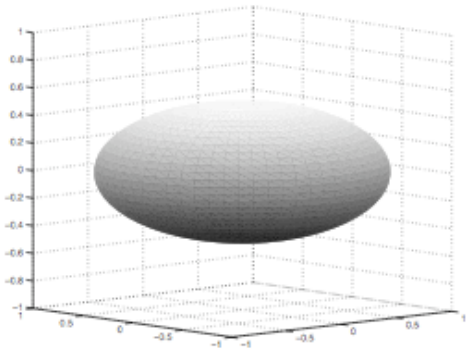
- We want to maximize information gain,
- So we want to minimize entropy.
- Given $\Delta I_{t+1|t} \propto E_z\left[\log\det(\Psi_{t+1})\right] - \log\det(\Psi_t)$, we minimize the covariance determinant.
- This is called "d-optimal" minimization.

What is "best"? D-Optimal

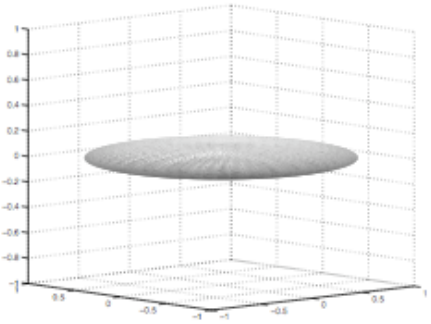
- Determinant is proportional to the volume of a hyperellipsoid where each dimension's diameter is an eigenvalue of the covariance.
- Can send to zero by minimizing one dimension.



(a) All eigenvalues at 1
$\det \Psi = 1$
$\mathrm{tr}(\Psi) = 3$



(b) One eigenvalue at 0.5
$\det \Psi = 0.5$
$\mathrm{tr}(\Psi) = 2.5$



(c) One eigenvalue at 0.1
$\det \Psi = 0.1$
$\mathrm{tr}(\Psi) = 2.1$

What is "best"? A-Optimal

- Idea: minimize *mean* error instead of overall variance
- Minimize trace instead of determinant:

$$\Delta I_{t+1|t} = E_z\left[\mathrm{tr}(\Psi_{t+1})\right] - \mathrm{tr}(\Psi_t)$$

- That's proportional to the mean for a constant feature count.

What is "best"? A-Optimal

- The change may seem arbitrary, but:

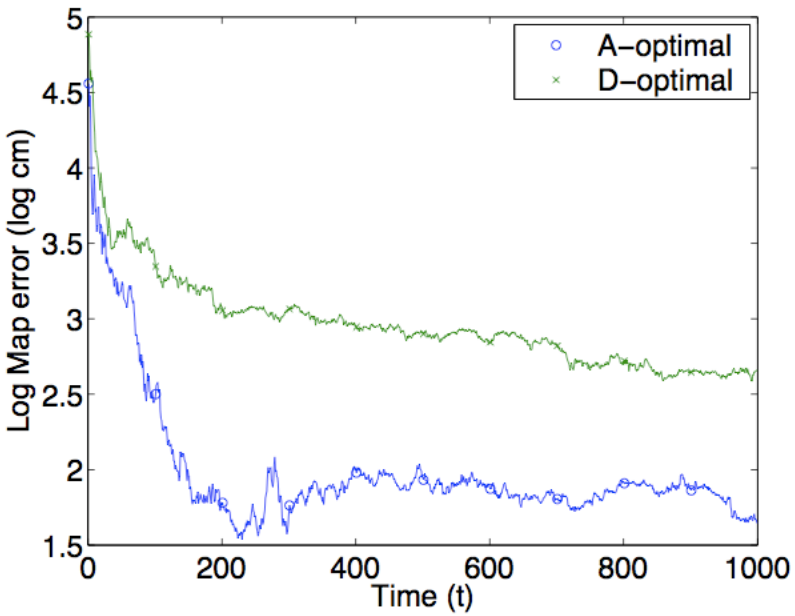


Figure 2. Map accuracy as a function of time for strategies optimizing an *a-optimal* and a *d-optimal* measure of uncertainty using a closed-loop, greedy strategy..

And now, for exploration…

Greedy Exploration

- At every step, pick the single “best” action.
- Fast!
- Simple!
- Not very effective!

Global Exploration

- Idea: pick the "best" *sequence* of actions.
- Optimally accurate!
- Clearly intractable without manipulation.

Pruning the Search

- Discretize environment into grid.
- Robot can move to 8 connected neighbors.
- Find best path *which doesn't cross itself.*
- Repeat until uncertainty is low enough.

Algorithmic Idea

- Because paths don't cross, each point has one *best* covariance trace.
- For each point, we store the best trace and the last point visited in the best path to it.
- Only update these if the trace along some other path is lower.
- Use a priority queue for the states to speed up convergence.

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
    a) Pop $\{x, y, \xi, \Psi\}$
    b) For each neighbor (x', y') of (x, y):
        i) Compute ΔI
        ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
            A) $I(x', y') = I(x, y) + \Delta I$
            B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
            C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)}\, I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
    a) $(x'y') = (x, y)$
    b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
    a) Pop $\{x, y, \xi, \Psi\}$
    b) For each neighbor (x', y') of (x, y):
        i) Compute ΔI
        ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
            A) $I(x', y') = I(x, y) + \Delta I$
            B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
            C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)}\, I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
    a) $(x'y') = (x, y)$
    b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
   a) Pop $\{x, y, \xi, \Psi\}$
   b) For each neighbor (x', y') of (x, y):
      i) Compute ΔI
      ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
         A) $I(x', y') = I(x, y) + \Delta I$
         B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
         C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)} I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
   a) $(x'y') = (x, y)$
   b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
    a) Pop $\{x, y, \xi, \Psi\}$
    b) For each neighbor (x', y') of (x, y):
        i) Compute ΔI
        ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
            A) $I(x', y') = I(x, y) + \Delta I$
            B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
            C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)} I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
    a) $(x'y') = (x, y)$
    b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
    a) Pop $\{x, y, \xi, \Psi\}$
    b) For each neighbor (x', y') of (x, y):
        i) Compute ΔI
        ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
            A) $I(x', y') = I(x, y) + \Delta I$
            B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
            C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)}\, I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
    a) $(x'y') = (x, y)$
    b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
    a) Pop $\{x, y, \xi, \Psi\}$
    b) For each neighbor (x', y') of (x, y):
        i) Compute ΔI
        ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
            A) $I(x', y') = I(x, y) + \Delta I$
            B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
            C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)} I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
    a) $(x'y') = (x, y)$
    b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

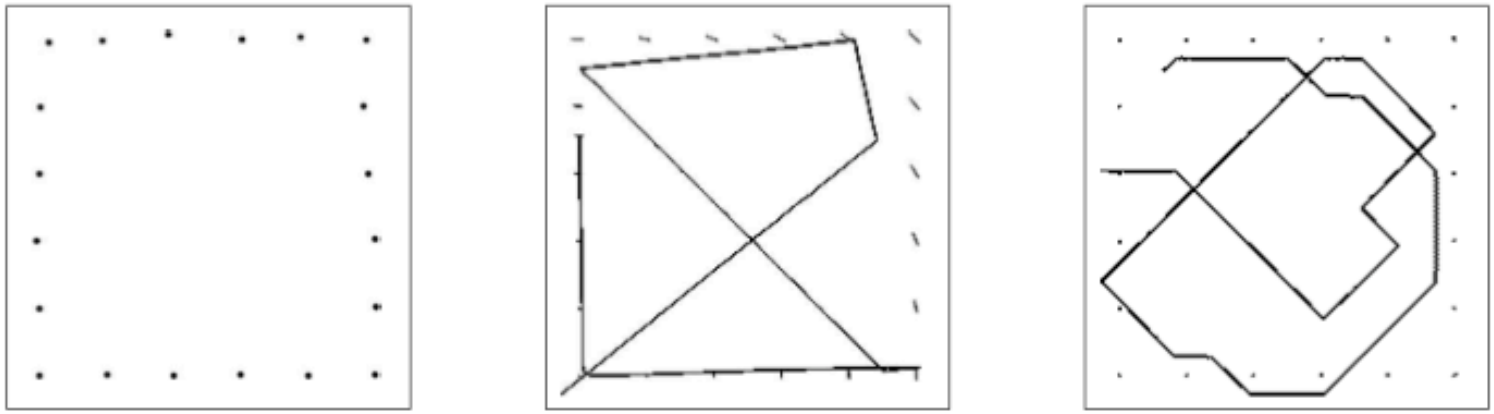
Convergence

- No state can be repeated on any trajectory.
- Entropy goes down with every measurement, since we prune bad paths.
- Therefore, the algorithm converges.

Time Analysis

- Assume a priority queue with linear search.
- For *s* positions, we have $O(s^2)$ updates.
- Checking the *m*-length "parent" list at each step is $O(m)$.
- But the list is bounded by *s* (no repeats!), so the total running time is $O(s^3)$.
- This can be much better with faster queues.

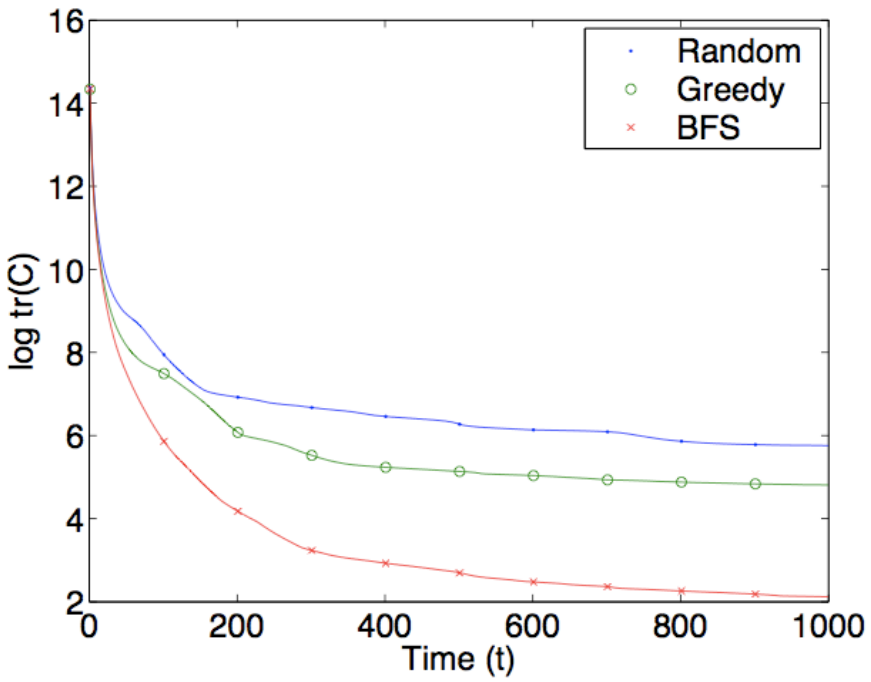
Performance



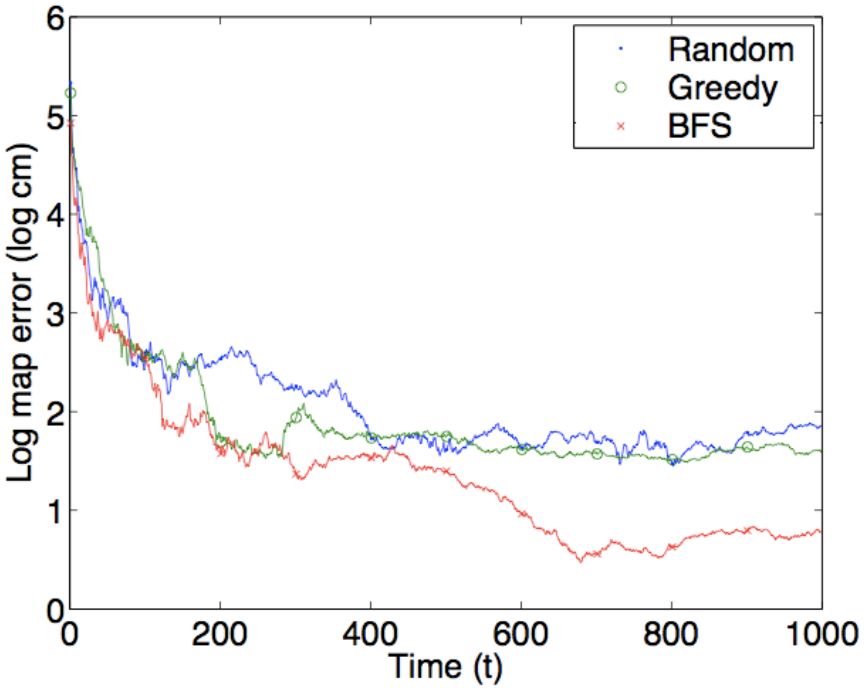
(a) Initial Map (b) Greedy Trajectory (c) Global Trajectory

Figure 3. (a) An exploration problem, with a set of features arranged in a ring around the edge of the environment. (b) The trajectory from a greedy exploration algorithm, computing the single position in the environment with maximum expected information gain, and moving to that position. (c) The trajectory from our global planning algorithm; notice the deliberate loop-closing.

Performance



(a) Trace of Posterior Covariance



(b) Map Accuracy

My Thoughts

- Changing the meaning of "best" had huge impact. What other "best"s are good?
- How much do we lose by not allowing loops in our paths?
- What if making an observation is expensive?

Questions?