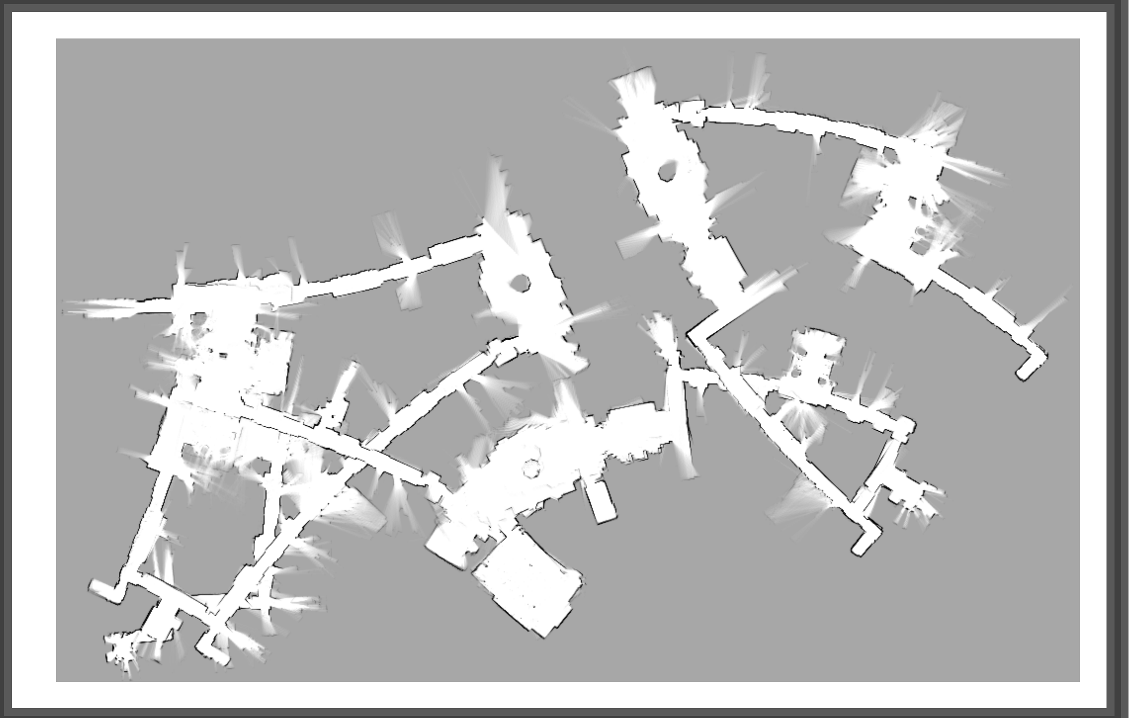
SLAM with sparse sensing

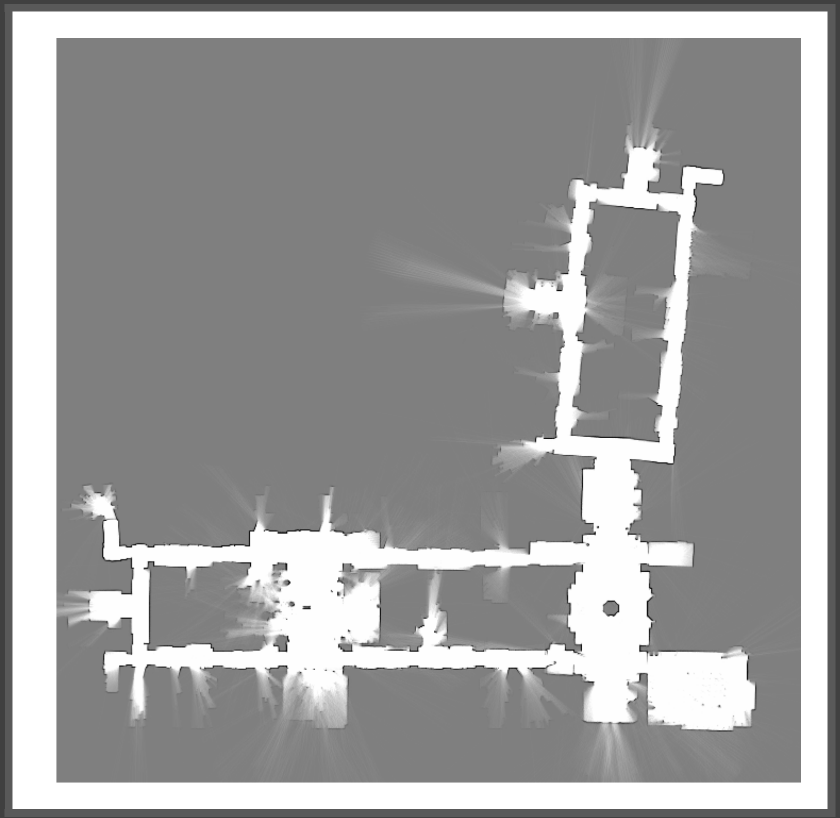
Kris Beevers
Department of Computer Science
Rensselaer Polytechnic Institute
beevek@cs.rpi.edu

October 19, 2005

SLAM



The problem



The goal

(Scan matching SLAM result courtesy of Brian Gerkey)

SLAM

- Build a map while localizing
- Approaches:
  - $\hookrightarrow$ **EKF-driven**: landmark based, scan matching
  - $\hookrightarrow$ Topological
- Landmark based SLAM with EKF:
  - $\hookrightarrow$ Assume Gaussian error models (motion, sensing)
  - $\hookrightarrow$ Move, predict, sense, update, ...
  - $\hookrightarrow$ State: $x(k) = [x_r(k)\ x_f(k)]$, where $x_f(k) = [x_{f_1}\ \dots\ x_{f_n}]$
  - $\hookrightarrow$ Problem: state covariance is large

Particle filters

- Given: an input (motion), motion model, measurement model
- N particles represent posterior (each has own state: pose, map)

```
1: for all particles do
2:     Project pose forward: draw pose from motion model distribution
3:     Extract features, perform data association
4:     Compute innovation (actual - predicted)
5:     Update map, initialize new features
6:     Compute particle weight ~ data association likelihood
7: end for
8: Resample particles w.p. proportional to weights
```

Covariance

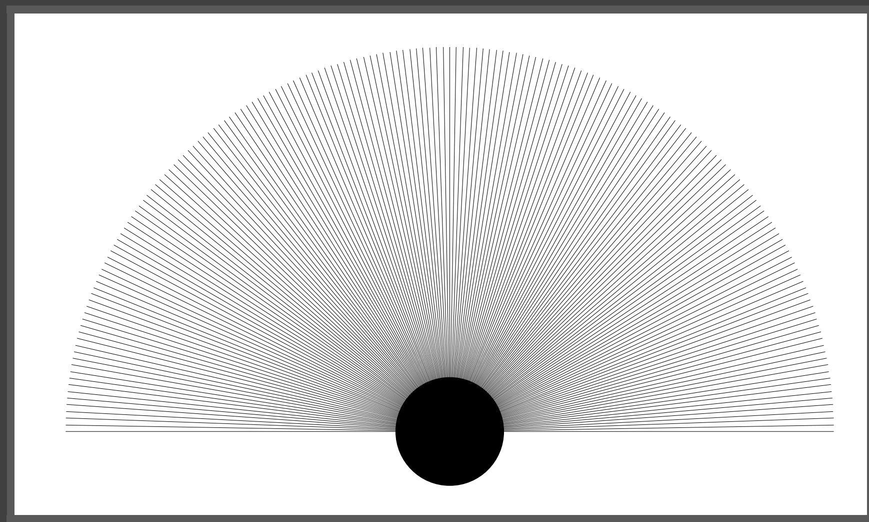
Full covariance: $O(n^2)$

$$P_x = \begin{bmatrix} P_{x_r} & P_{x_r x_f} \\ P_{x_r x_f}^T & P_{x_f} \end{bmatrix} \quad P_{x_f} = \begin{bmatrix} P_{x_{f_1}} & P_{x_{f_1} x_{f_2}} & \dots & P_{x_{f_1} x_{f_n}} \\ P_{x_{f_2} x_{f_1}} & P_{x_{f_2}} & \dots & P_{x_{f_2} x_{f_n}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{x_{f_n} x_{f_1}} & P_{x_{f_n} x_{f_2}} & \dots & P_{x_{f_n}} \end{bmatrix}$$

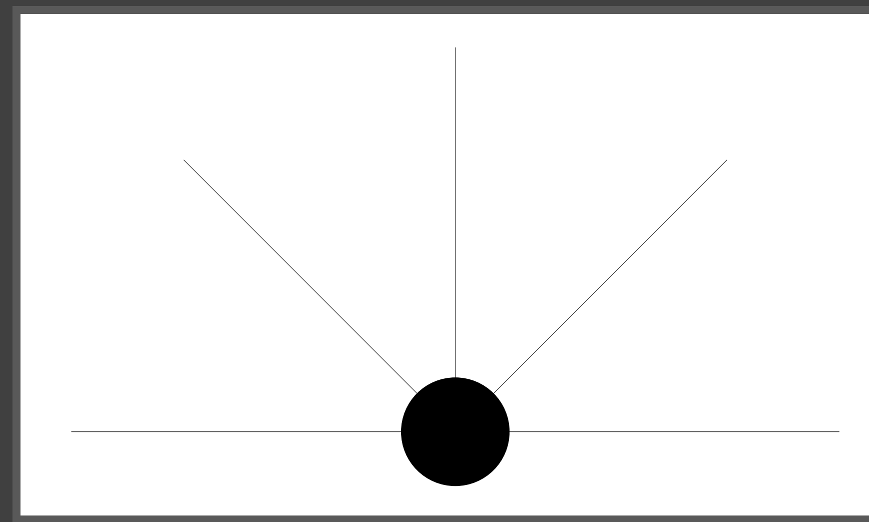
Particle filter: $O(n)$ — each particle has

$$P_x = \left[P_{x_r}\ P_{x_{f_1}}\ P_{x_{f_2}}\ \dots\ P_{x_{f_n}} \right]$$

Sparse sensing



Laser rangefinder



Sparse array

- Cheap (5 infrared rangefinders: $<$ US \$40), low power
- **Problem**: low density

$\hookrightarrow$ extract features by taking scans from multiple poses

SLAM with a pose history

- Do feature extraction on scan data from the last m poses
- Trades off measurement uncertainty and scan density
- J. Leonard, R. Rikoski, P. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *IJRR*, 21(10):943–975, October 2002.

$\hookrightarrow$ Keep most recent m poses in the system state vector:

$$\boxed{x_r(k) = \left[x_{t_k}\ x_{t_{k-1}}\ \ldots\ x_{t_{k-m+1}}\right]}$$

$\hookrightarrow$ Use EKF: high-fidelity, accounts for correlation between poses

$\hookrightarrow$ Massively expensive: $O((m+n)^2)$, plus feature extraction at every time step

Particle filtering with a pose history

for all particles p^i **do**
  Project state forward: draw $x^i_{t_k}$ from motion model distribution centered at $f(x^i_{t_{k-1}}, u(k-1))$, insert $x^i_{t_k}$ into $x^i_r(k)$, discard $x^i_{t_{k-m}}$
  Extract features using last m scans and $x^i_r(k)$, do data association
  Compute innovation, update map, initialize new features
  Compute particle weight $\sim$ data association likelihood
end for
Resample particles w.p. proportional to weights

- Each particle has a unique pose history: particles sample the space of the last m pose histories

$\hookrightarrow$ may require many particles

$\hookrightarrow$ need to extract features for every particle

SLAM with multiscans

- We want:
  - $\hookrightarrow$ Fewer particles
  - $\hookrightarrow$ One feature extraction for each SLAM iteration
- Simplifications we make:
  - $\hookrightarrow$ Group scans from m consecutive poses into a *multiscan*:

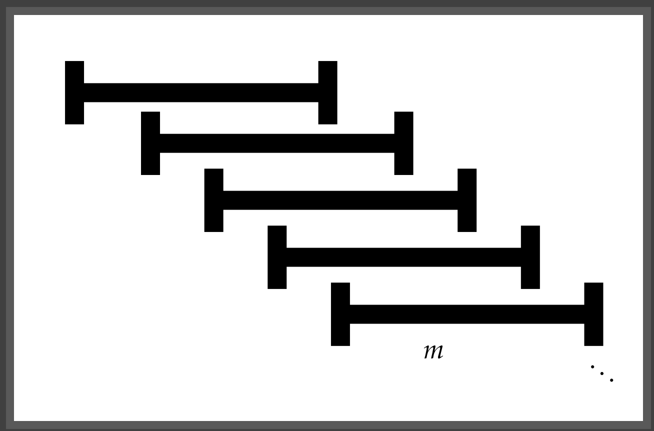
$$\boxed{\mathbf{z}(k) = [z(k)\ z(k-1)\ \ldots\ z(k-m+1)]}$$

  - $\hookrightarrow$ Perform SLAM update only after each m steps
  - $\hookrightarrow$ Extract features using the *expected* pose history

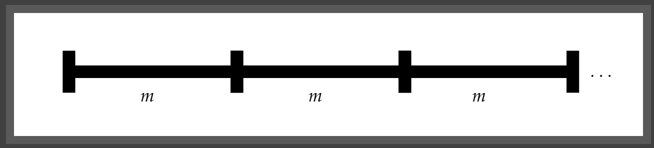
Algorithm

```
For $m$ time steps: move and collect sparse scans
Extract features from multiscan using expected pose history
**for all** particles $p^i$ **do**
    **for** $i = (k - m + 1) \ldots k$ **do**
        Project pose forward by drawing from motion model
    **end for**
    Data association between extracted features and map $x_f^i(k - m)$
    Compute innovation, update map, initialize new features
    Compute particle weight $\sim$ data association likelihood
**end for**
Resample particles w.p. proportional to weights
```

The computational difference



- Feature extraction ($\mathcal{Z}$) and SLAM for every particle at every time step
- Each time step: $O(N \log n) + O(N\mathcal{Z})$



- Feature extraction ($\mathcal{Z}$) once every m time steps
- SLAM for every particle every m time steps
- Every m time steps: $O(N \log n) + O(\mathcal{Z})$
- Smaller N since state doesn't include pose history

Innovation covariance

- $\mathcal{Z}(k) = g(\mathbf{z}(k), \mathbf{x}(k))$: measurement (feature extractor)
- Innovation: $\nu = \mathcal{Z}(k) - Mx(k)$

$\hookrightarrow$ $M \equiv$ selection matrix, result of data association

- We want *innovation covariance*: $S = J_g P_{(\mathbf{z},\mathbf{x})} J_g^T + M P_{x(k)} M^T$
- Problem: what are $J_g, P_{(\mathbf{z},\mathbf{x})}$?

$$\boxed{P_{(\mathbf{z},\mathbf{x})} = \begin{bmatrix} P_{\mathbf{z}} & P_{\mathbf{zx}} \\ P_{\mathbf{zx}}^T & P_{\mathbf{x}} \end{bmatrix}}$$

Maximum likelihood feature extraction

- J_g is hard to compute for complicated feature extractors
- H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–26, January 1982.
  - $\hookrightarrow$ MLE gives good covariance estimates for the parameters being estimated even for an approximately/poorly specified model
- Use MLE as a feature extractor to get a good approximation of S without computing J_g
- We extract line segment features based on this

Results

Data from RADISH

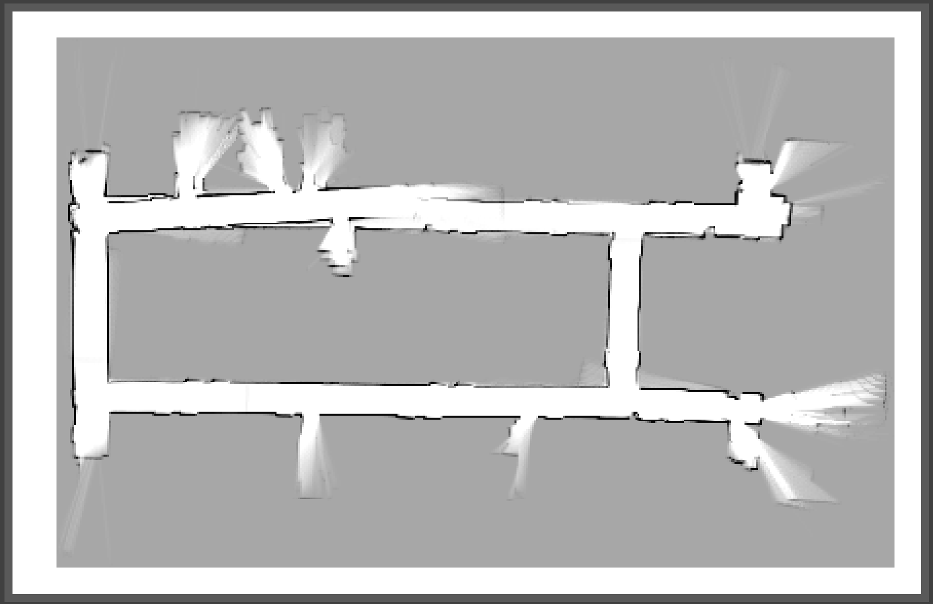
USC SAL Building: Andrew Howard

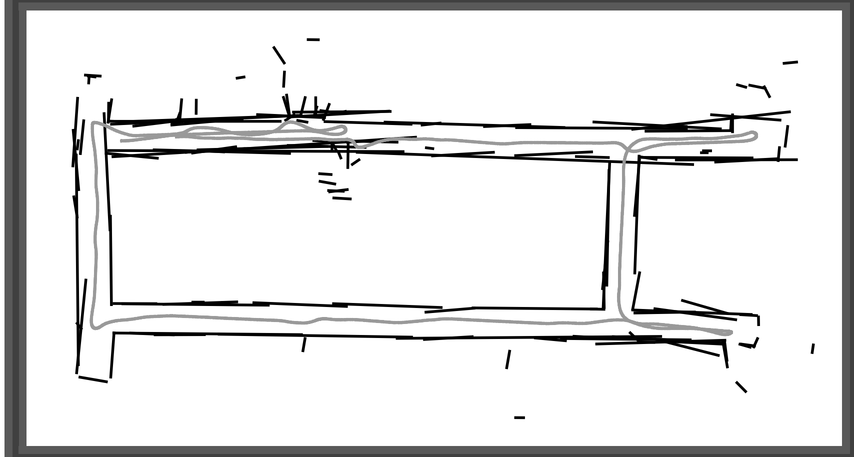
CMU Newell-Simon Hall: Nick Roy

Stanford Gates Building: Brian Gerkey

- Full laser rangefinder datasets
- Keep only the measurements at $0°, 45°, 90°, 135°$, and $180°$
- FastSLAM 1.0, modified for sparse sensing SLAM

Results: USC SAL

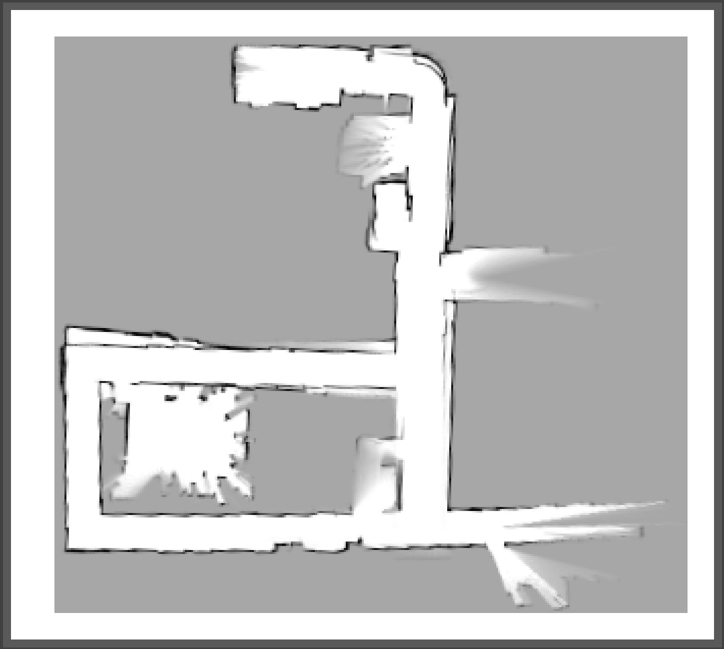


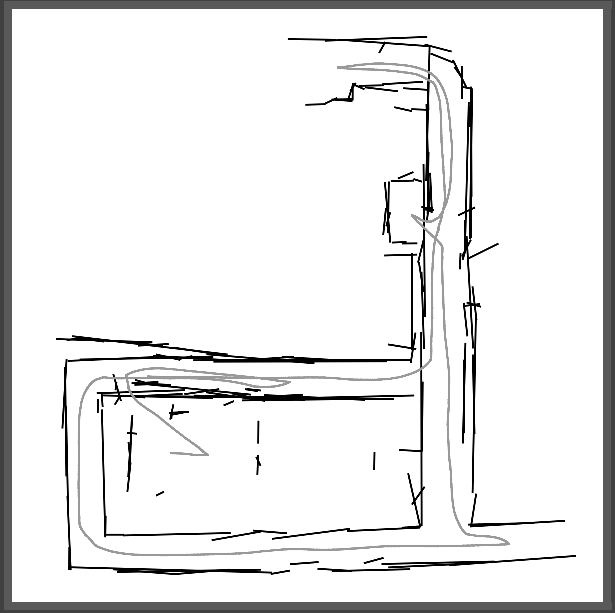


Scans/multiscan	50
Particles	400
Sensing range	5m

Dimensions	39m $\times$ 20m
Trajectory length	122m
Trajectory rotation	338 rad
Landmarks	145

Results: CMU Newell-Simon Hall

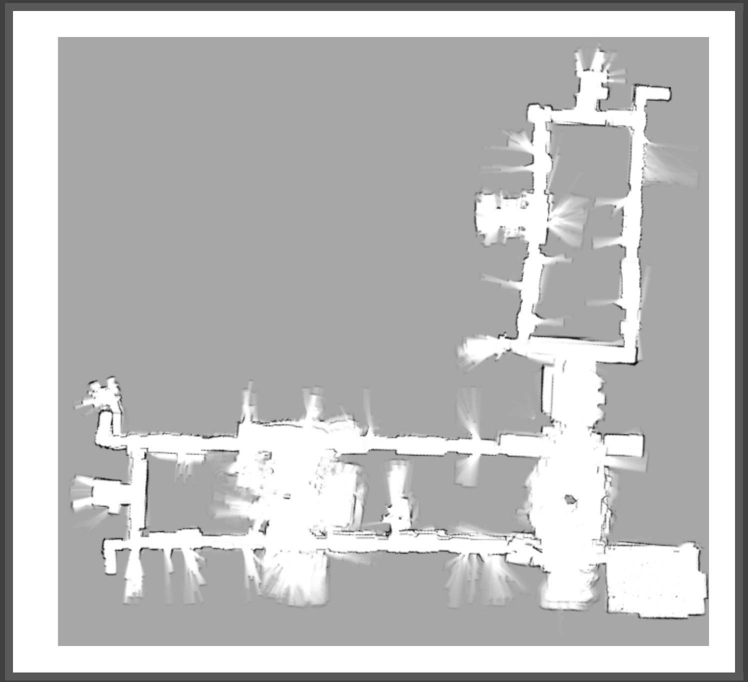


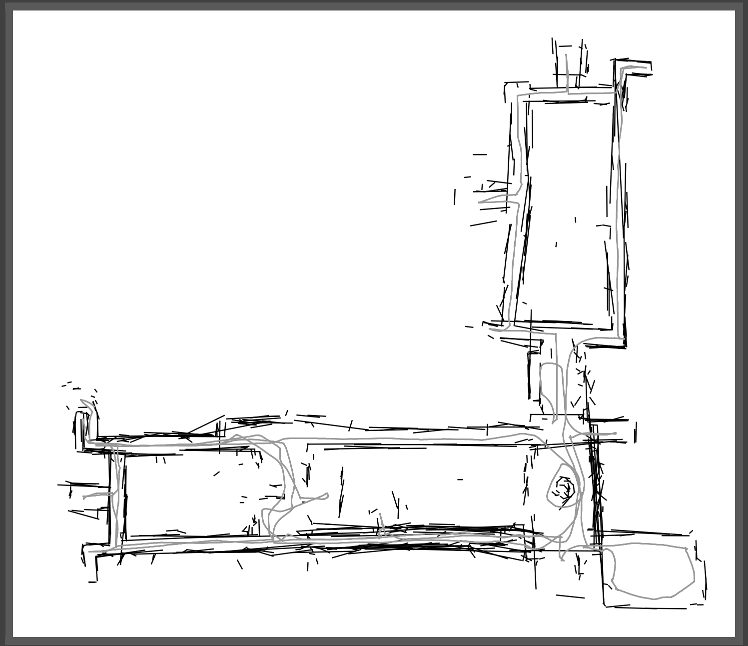


Scans/multiscan	40
Particles	600
Sensing range	3m

Dimensions	25m $\times$ 25m
Trajectory length	114m
Trajectory rotation	133 rad
Landmarks	168

Results: Stanford Gates Building

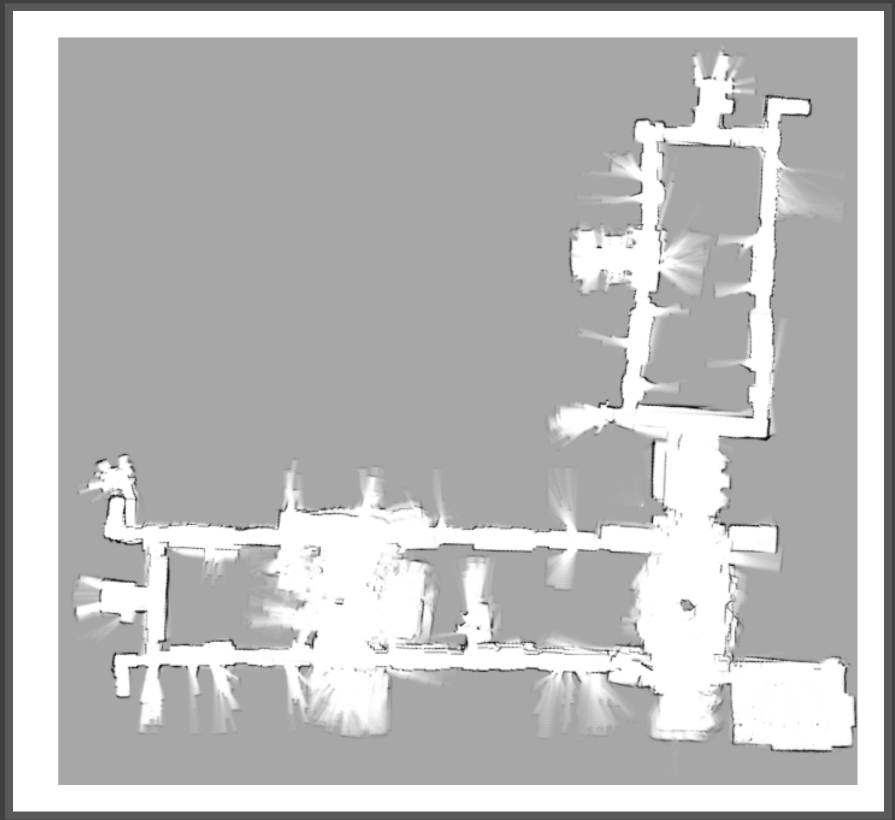




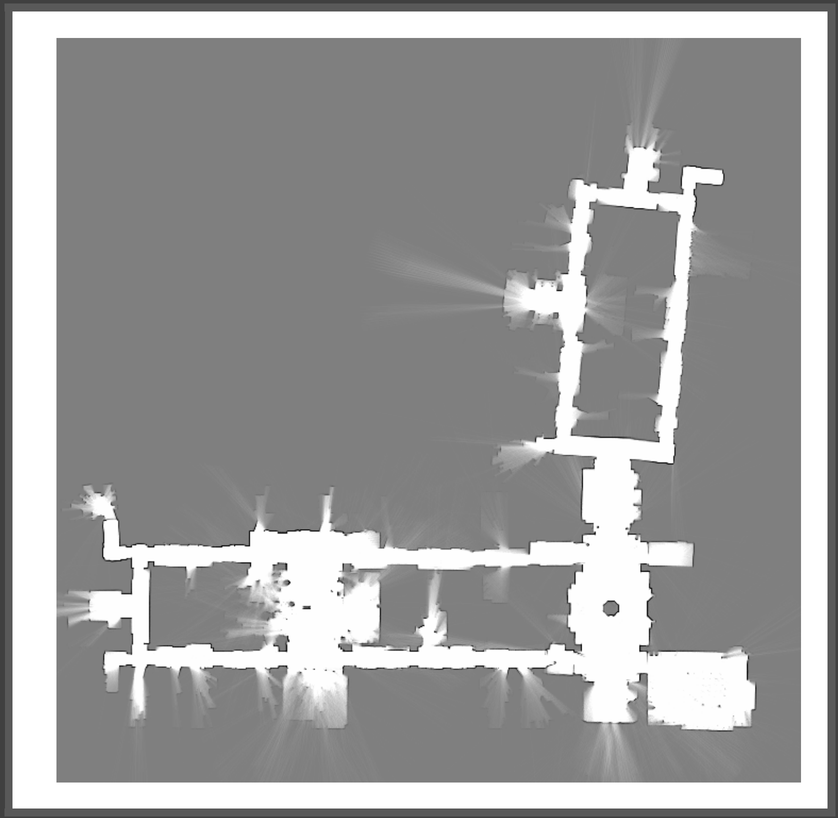
Scans/multiscan	18
Particles	1000
Sensing range	5m

Dimensions	64m $\times$ 56m
Trajectory length	517m
Trajectory rotation	495 rad
Landmarks	750

Results: sparse vs. scan matching



Sparse sensing (5 sensors)



Scan matching (full scan)

Tradeoffs of sparse sensing SLAM

- Using odometry to augment sensing:

$\hookrightarrow$ more uncertainy

$\hookrightarrow$ requires more particles

- Sensitive to amount of uncertainty accumulated in a multiscan

- Scans/multiscan (m) is a “magic number”

- Too many approximations:

$\hookrightarrow$ poor data association

$\hookrightarrow$ spurious landmarks

This is the last slide!

Questions?