Mapping with limited sensing

Kris Beevers

Algorithmic Robotics Laboratory
Department of Computer Science
Rensselaer Polytechnic Institute

October 17, 2006

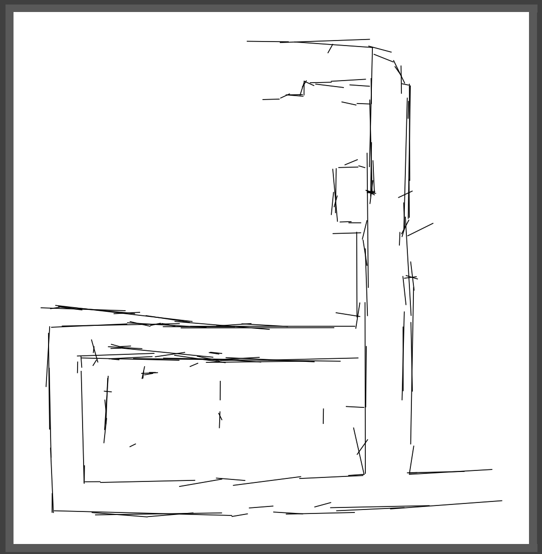
Robot mapping

Basic problem: raw sensing data $\longrightarrow$ useful map of the environment

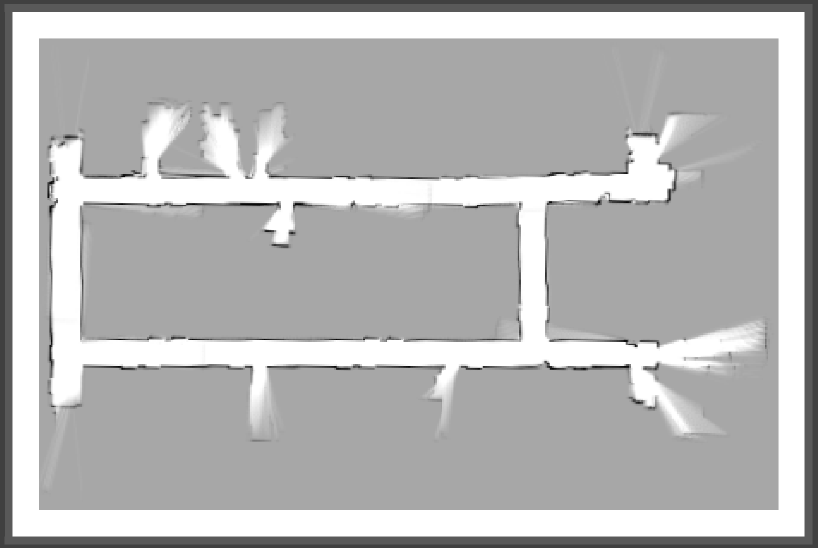
- Useful to whom?
  - Navigation (other robots, people)
  - Search and rescue
  - Reconnaissance, hazmat detection
  - Sensor network localization
  - etc...

- What context?
  - Environment: **structured** or **unstructured**, cluttered, **2D**, 3D
  - Robot: **sensing** and computational capabilities, actuation, odometry uncertainty, etc.

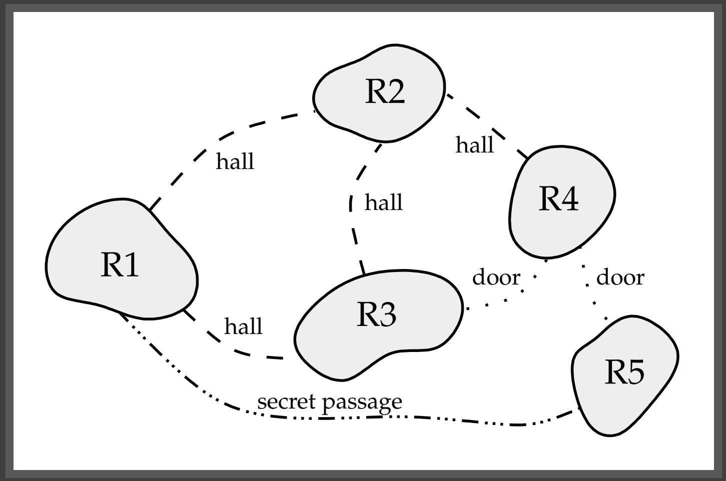
Map representation



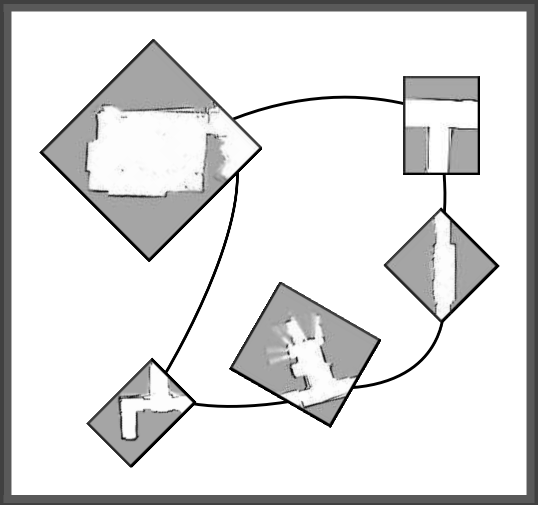
Landmark



Occupancy grid



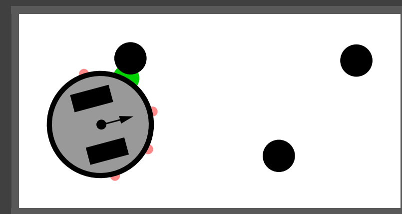
Topological



Hybrid

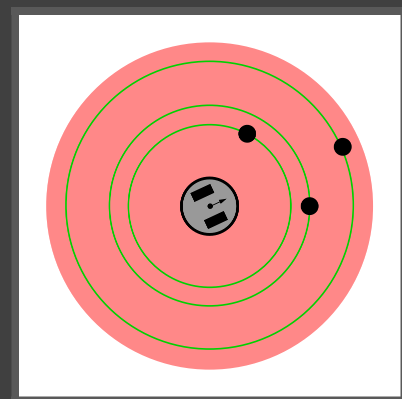
Sensors for mapping

Contact sensor array



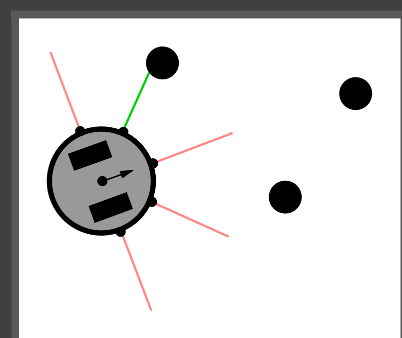
Zero-range, low-res, accurate, cheap

RF signal strength



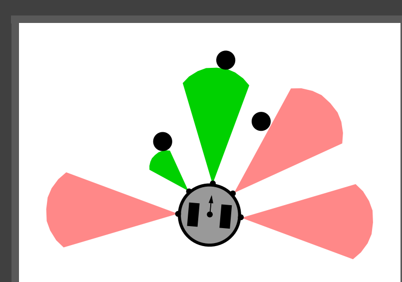
Mid-range, no-res, inaccurate, medium-cost
no bearing information (range only)

Infrared array



Short-range, low-res, accurate, cheap

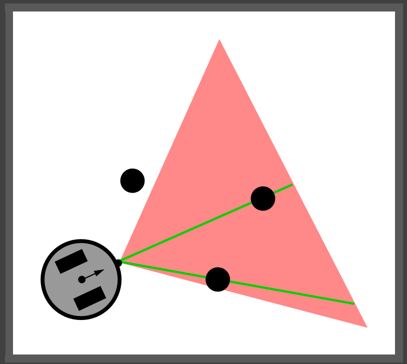
SONAR array



Mid-range, low-res, inaccurate, medium-cost

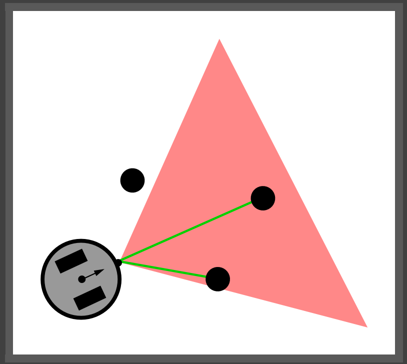
Sensors for mapping (cont.)

Monocular camera



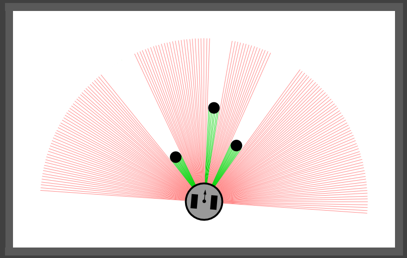
Long-range, high-res, accurate, medium-cost
no range information (bearing only)

Stereo camera



Long-range, high-res, accurate, high-cost

Laser rangefinder

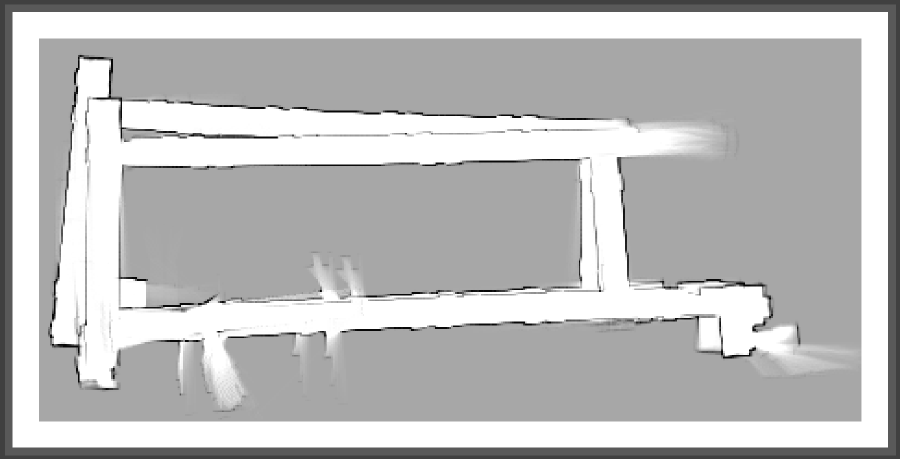


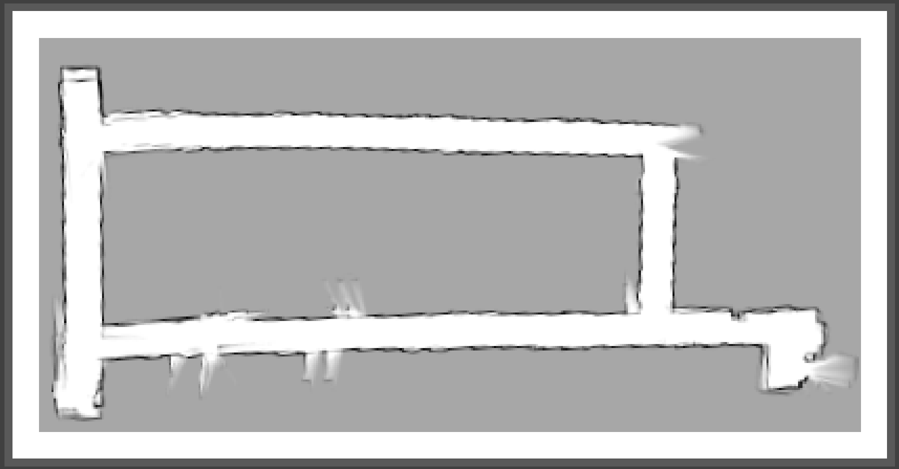
Long-range, high-res, accurate, high-cost

Simultaneous localization and mapping (SLAM)

- **Odometry** is notoriously noisy!
  - Cannot simply build map based on odometry-estimated trajectory
  - GPS is often not available (e.g., indoors)
- SLAM: Alternate mapping and localization steps:

1. Use sensor returns to improve pose estimate **based on current map**
2. Update the map with the sensor returns

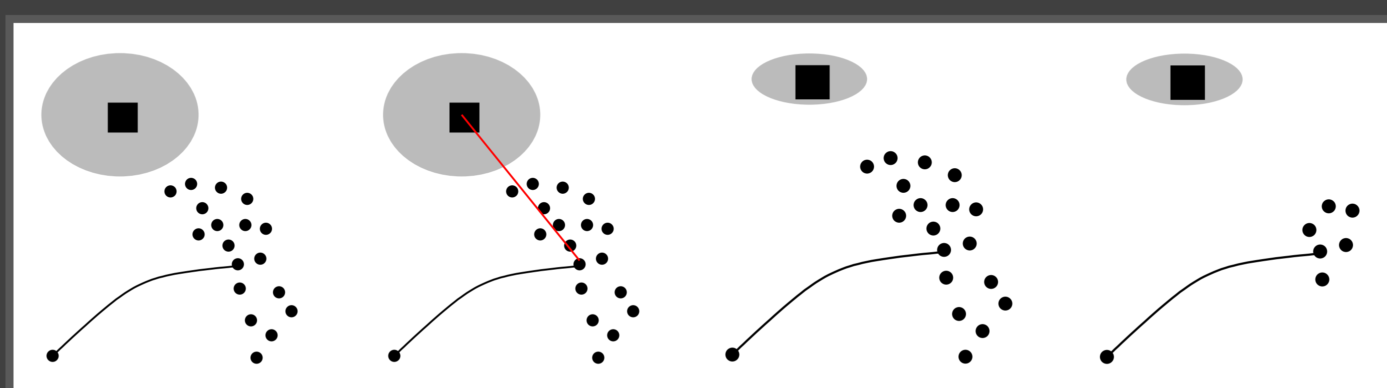


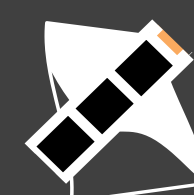


$$\underbrace{p(\mathbf{x}_t|\mathbf{u}_{1:t},\mathbf{z}_{1:t},\mathbf{n}_{1:t})}_{\text{posterior}} = \eta \ \underbrace{p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{n}_t)}_{\text{measurement}} \int \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)}_{\text{motion}} \underbrace{p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1},\mathbf{z}_{1:t-1},\mathbf{n}_{1:t-1})}_{\text{prior}} \, d\mathbf{x}_{t-1}$$

Particle filtering SLAM (sequential Monte Carlo)

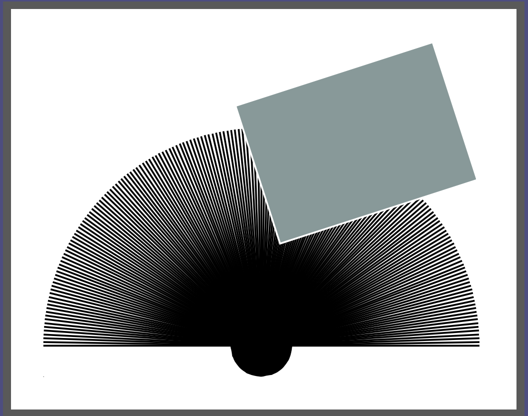
loop
    Move/sense/extract features
    for all particles ϕ^i **do**
        Project forward: $\mathbf{x}_t^{r,i} \sim p(\mathbf{x}_t^r | \mathbf{x}_{t-1}^{r,i}, \mathbf{u}_t)$
        Do data association (compute $\mathbf{n}_t^i$), update map
        Compute weight: $\omega_t^i = \omega_{t-1}^i \times p(\mathbf{z}_t | \mathbf{x}_t^{r,i}, \mathbf{x}^{m,i}, \mathbf{n}_t^i)$
    end for
    Resample (with replacement) according to ω_t^is
end loop

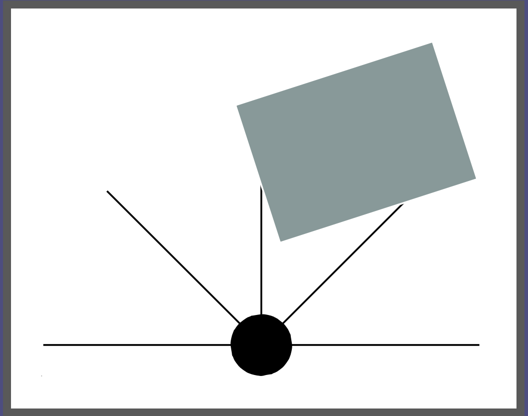


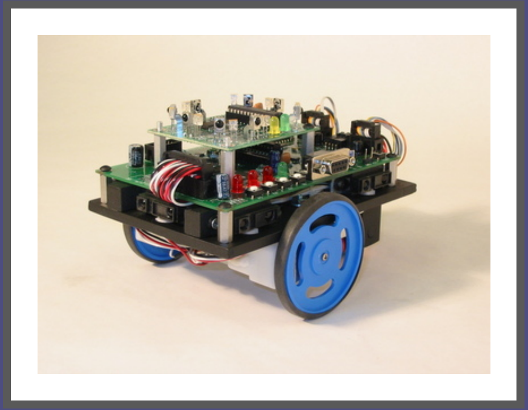


Our research

- Broadly: **mapping with limited sensing**
- Most mapping research assumes:
  - Accurate range/bearing measurements
  - “Dense” data suitable for feature extraction
  - Usually: scanning laser rangefinders
- What about, e.g., arrays of IR sensors?
  - Cheap ($10’s vs. $1000’s)
  - Less power, smaller
  - **But**: short-range, sparse data
- Challenges:
  - Extracting features from data
  - Managing lots of pose/map uncertainty
  - Characterizing map quality in terms of sensors

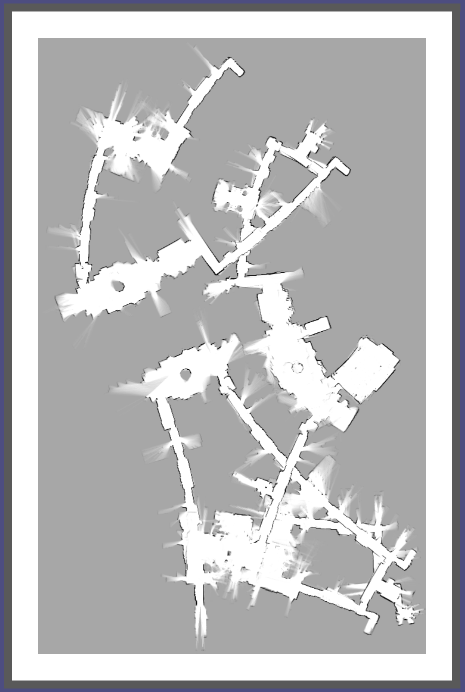




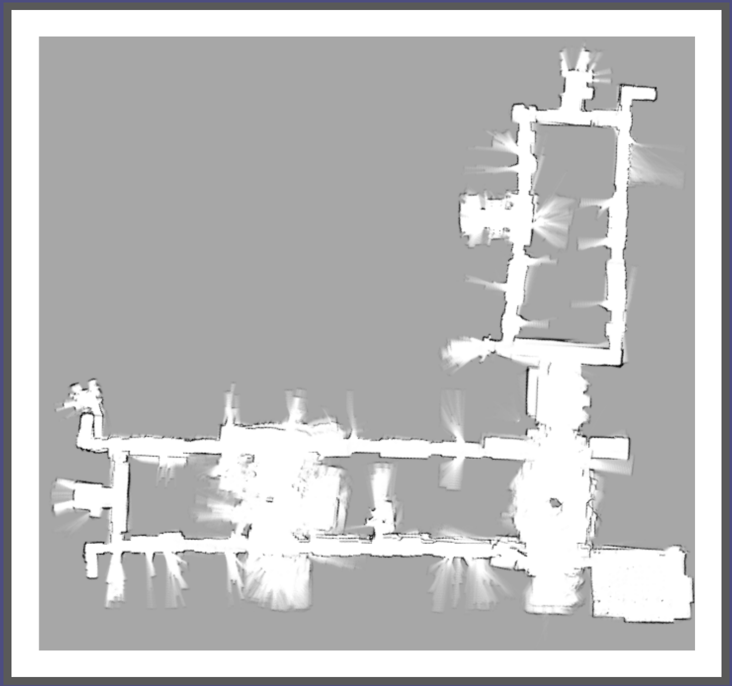


SLAM with sparse sensing

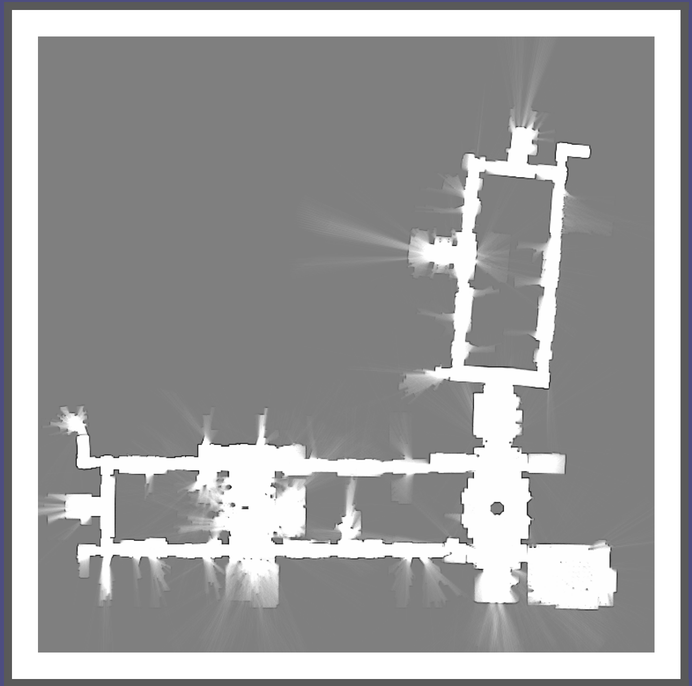
- 5 readings per scan instead of 180 — not enough to extract features
- Basic idea: feature extraction using data from multiple scans
- Challenges:
  - Particle filters need per-particle extraction (conditioned on trajectory)
  - Augmenting exteroceptive sensing with odometry: more uncertainty



Raw odometry



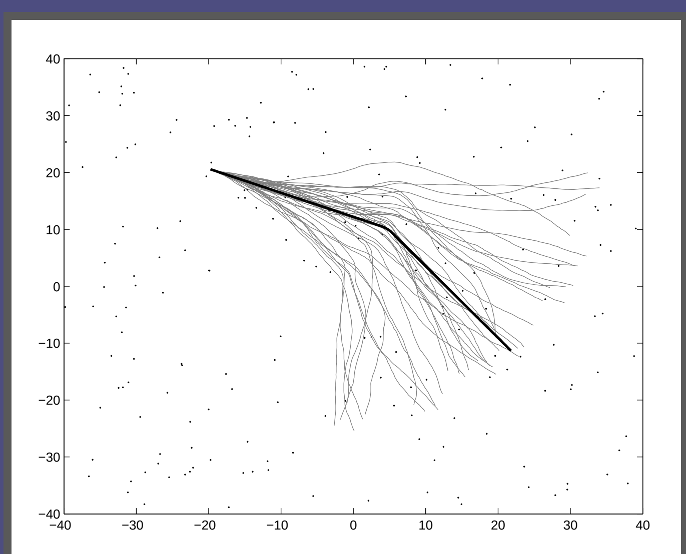
“Multiscan” landmark SLAM



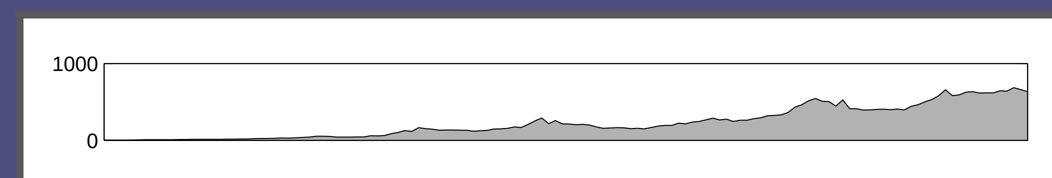
Full laser scan-matching

Improving estimation consistency

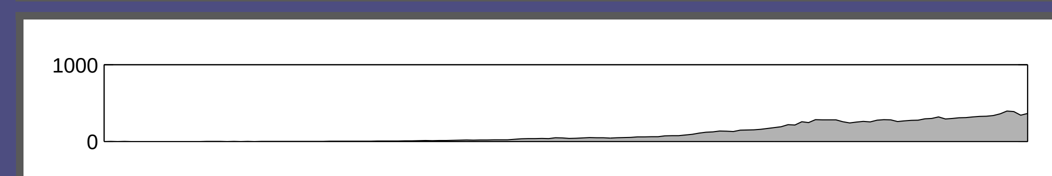
- Most practical SLAM algorithms do not use new information to improve past pose estimates
- Two approaches for doing this inexpensively in particle filtering:
  - **Fixed-lag roughening**: MCMC of particles over a lag time
  - **Block proposal distribution**: “re-draw” poses over lag time from their joint distribution
  - Both techniques: conditioned on the most recent odometry and sensor measurements



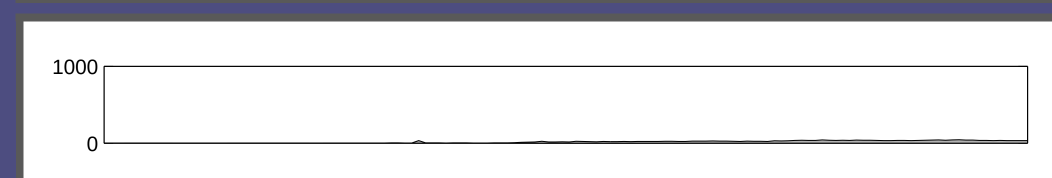
FS2



FLR(10)

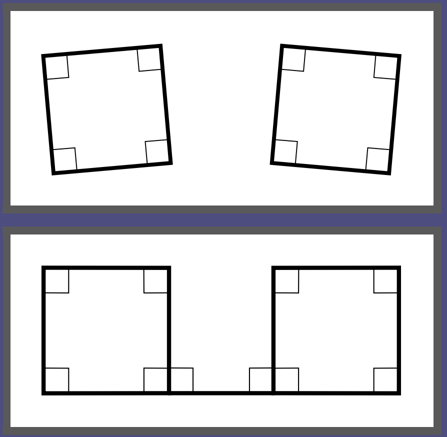


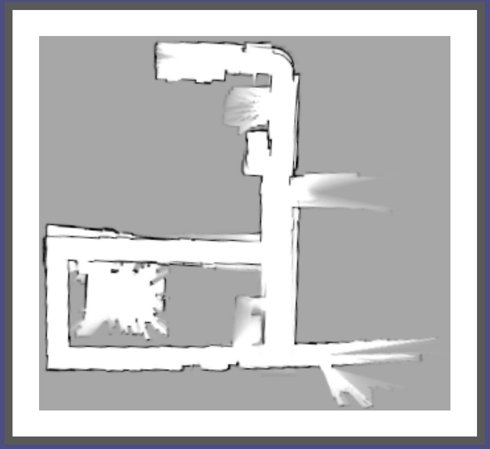
BP(10)



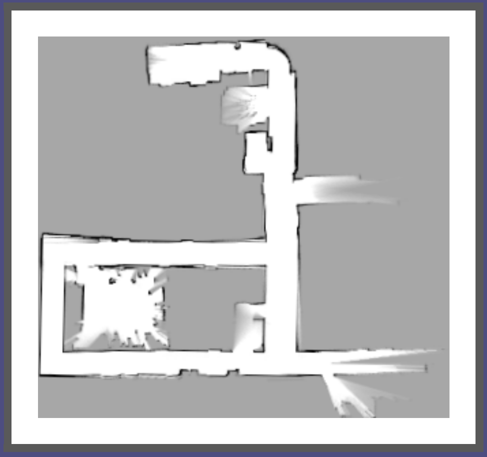
Exploiting prior knowledge: constrained SLAM

- We often know something about the environment **ahead of time**
- Example: indoor environments are “mostly” rectilinear
- Encode prior knowledge as constraints on the map
  - Infer existence of constraints between landmarks
  - Enforce constraints
- Challenge: breaks independence assumptions of particle filter

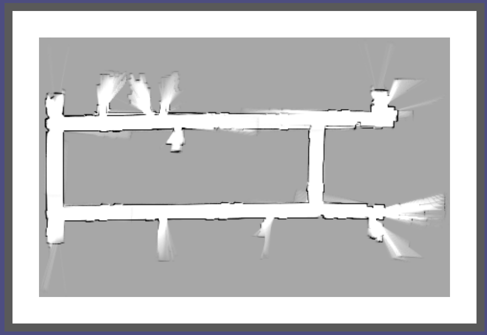




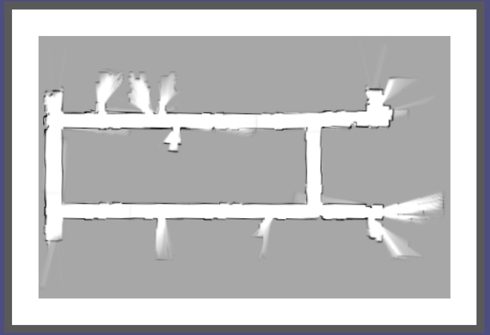
Unconstrained
600 particles



Rectilinearity
40 particles



Unconstrained
100 particles



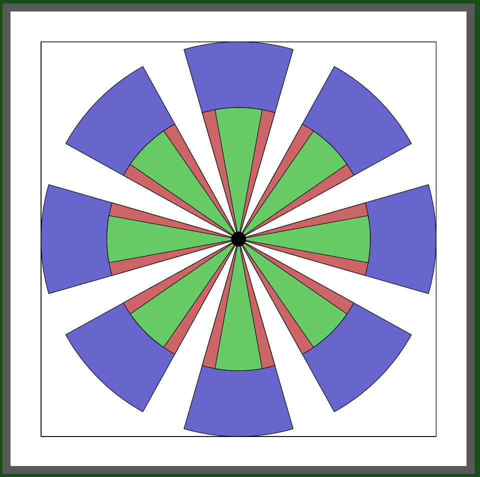
Rectilinearity
20 particles

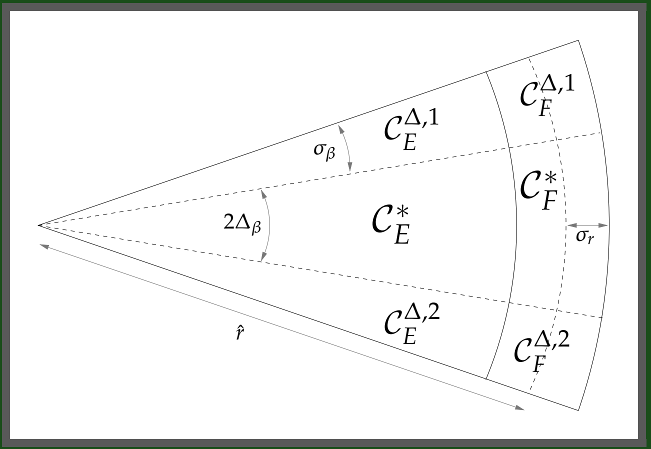
Analytical results on sensing and mapping

- **Question:** how can we relate "sensing capabilities" to map quality?
- Previous work: for every kind of sensor, either **design a specific algorithm or prove no algorithm exists** (localization, O'Kane and LaValle, 2006):
  - Binary characterization (**can or can't** localize)
  - Compass + contact sensor: **can** localize
  - Angular odometer + contact sensor: **can't** localize
- An alternative approach: **fix** the mapping algorithm and define a broad sensor model
  - Encompasses most types of practical mapping sensors
  - Characterize **which sensors can build a map**
  - Give **quality bounds** on the map for a given sensor

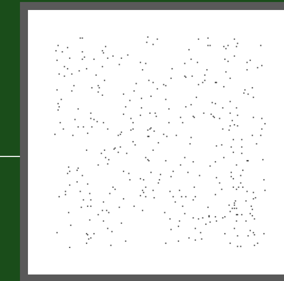
Model

- Environment: $M \times M$ grid of cells m_{ij}; cells occupied ($\mathtt{F}$) at rate d, $\mathtt{E}$ otherwise
- Trajectory: $\mathbf{x}_t^r, t \in [0, T]$; *assumption: poses drawn uniformly at random*
- Sensor:
  - Ring: ρ beams, angles $\beta_i = i\frac{2\pi}{\rho} + U[-\sigma_\beta, \sigma_\beta]$
  - Firing frequency F
  - Beam: goes until *detecting* an occupied cell
  - False negative rate $\varepsilon_{\mathtt{E}}$, false positive rate $\varepsilon_{\mathtt{F}}$
- Mapping: **occupancy grid**; cell measurements depend on "region" in beam
  - $m_{ij} \in \mathcal{C}_{\mathtt{F}}$: bel$(m_{ij} = \mathtt{F})$+=$p_0$
  - $m_{ij} \in \mathcal{C}_{\mathtt{E}}$: bel$(m_{ij} = \mathtt{E})$+=$p_0$

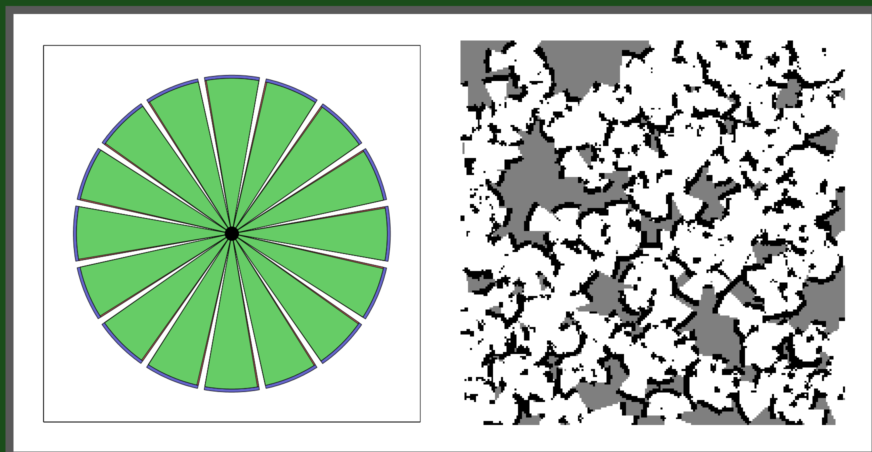




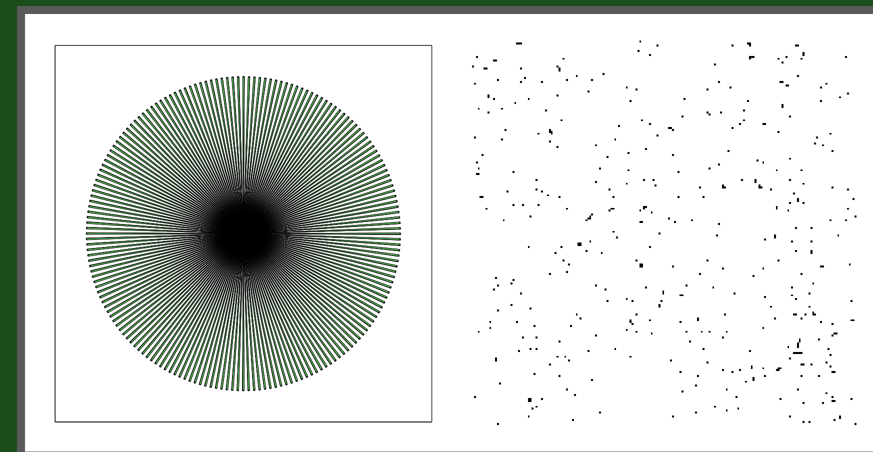
Some (synthetic) examples



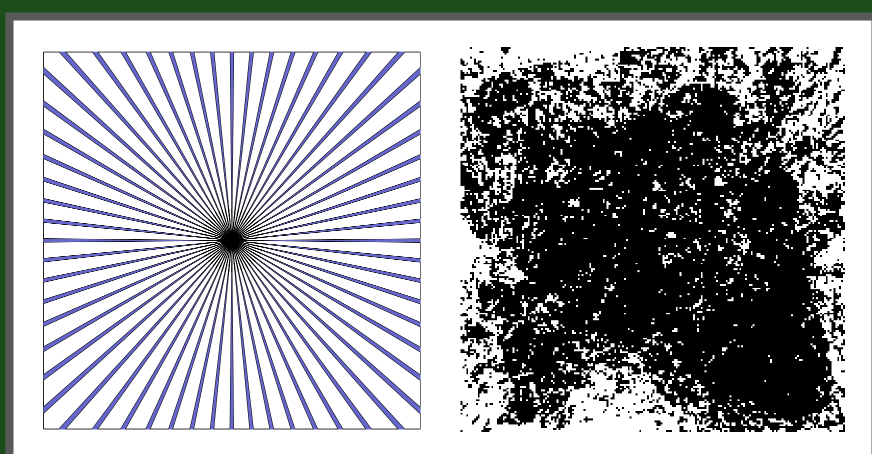
True map, $d = 0.01$



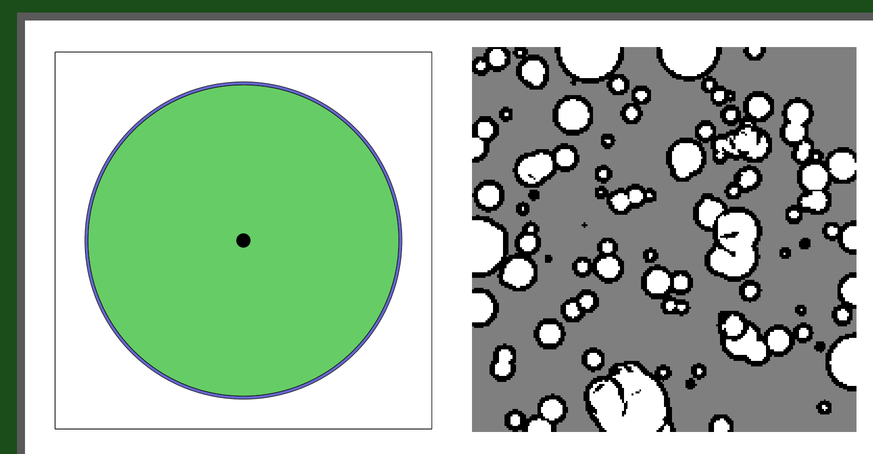
SONAR-like sensor



Laser-like sensor



Bearing-only sensor



Range-only sensor

Obtaining a bound on expected map error

Bound expected # observations of a cell

↓

Compute likelihood that an observation is incorrect

↙ ↘

Conditions for map convergence

Bound expected error in ML map

Bound on expected # observations

Let:

$$\mathcal{E}_{\mathtt{E}} = ((1-d)(1-\varepsilon_{\mathtt{E}}) + d\varepsilon_{\mathtt{F}}) \quad p(\text{some cell in a beam registers as } \mathtt{E})$$
$$\mathcal{E}_{\mathtt{F}} = (d(1-\varepsilon_{\mathtt{F}}) + (1-d)\varepsilon_{\mathtt{E}}) \quad p(\text{some cell in a beam registers as } \mathtt{F})$$

Expected # o_{ab} of times any cell m_{ab} is updated:

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta+\sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot p_{\text{obs}}}$$

where:

$$\boxed{p_{\text{obs}} \geq \begin{cases} \mathcal{E}_{\mathtt{E}}^{\Delta_\beta \tau^2} & \text{if } \tau\delta > \sigma_r \\ 1 & \text{otherwise} \end{cases}}$$

Likelihood of an incorrect observation

Let: $$\boxed{p_{\mathrm{f}} = \min\left\{1, \frac{\Delta_\beta \mathcal{E}_{\mathrm{F}}}{\delta^2}\left((\tau\delta + \sigma_r)^2 - \max\{0, \tau\delta - \sigma_r\}^2\right)\right\}}$$

If cell m_{ij} is unoccupied ($\mathtt{E}$) the likelihood that any update to m_{ij} is incorrect is:

$$\boxed{p(\text{inc}|m_{ij} = \mathtt{E}) \leq \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} p_{\mathrm{obs}} \cdot p_{\mathrm{f}} \cdot \frac{(\tau\delta+\sigma_r)^2 - \max\{0, \tau\delta-\sigma_r\}^2}{(\tau\delta+\sigma_r)^2}}$$

If cell m_{ij} is occupied ($\mathtt{F}$) the likelihood that any update to m_{ij} is incorrect is:

$$\boxed{p(\text{inc}|m_{ij} = \mathtt{F}) \leq \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} p_{\mathrm{obs}} \cdot p_{\mathrm{f}} \cdot \frac{\max\{0, \tau\delta-\sigma_r\}^2}{(\tau\delta+\sigma_r)^2}}$$

Bound on expected ML map error

$\boxed{\text{The map converges if } p_{\text{inc}} < 1/2}$

Let $v = \sum_{ij} v_{ij}$, where $v_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**, and $v_{ij} = 0$ otherwise.

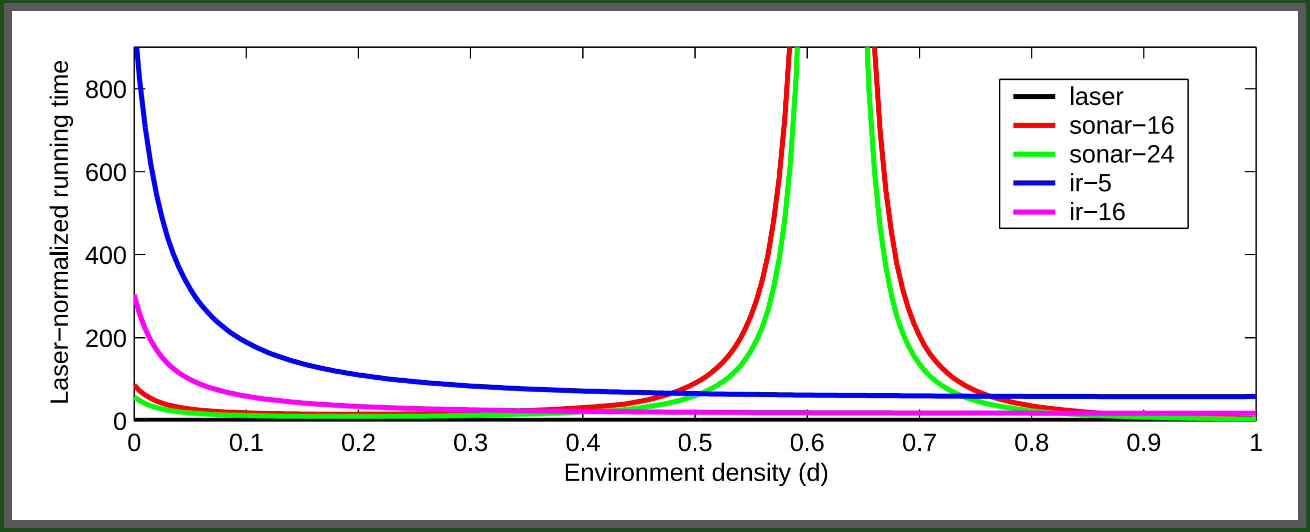
If $p_{\text{inc}} < 1/2$:

$$\boxed{E[v] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\text{inc}}\right)^2\right\}}$$

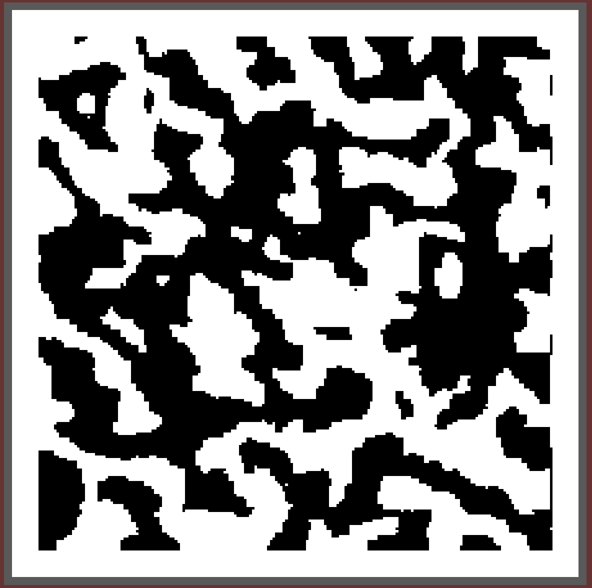
(Chernoff bound)

Application: comparing real sensors

- We obtained model parameters for three real sensors used in mapping:
  - SICK LMS 200-30106 scanning laser rangefinder
  - Polaroid 6500 series SONAR ranging module
  - Sharp GPD12 infrared rangefinder
- **"Laser-normalized" running time**
  - Extra work (time) required for a sensor to build a map of (expected) quality equivalent to that build by the scanning laser rangefinder
  - Depends only on sensor characteristics and environment density



Future directions



- Big gap between our approach and real SLAM:
  - Realistic trajectories
  - Structured environments — MRF model
  - Modeling measurement correspondences
  - Pose uncertainty
  - Beyond simulation — how well does our model match reality?

- Right now, many mapping problems are "solved" if you throw enough $ at them, but:
  - Practical mapping with inexpensive robots: limited **sensing, computing, memory, energy**
  - Sensing capability is a function of the environment
  - What are the *requirements* for a mapping robot?

Thanks for coming!