Autonomous Robotics

Simultaneous Localization and Mapping

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework #3 released soon.
- Reading assignment for next week (Kinematics) has been posted.
- Midterm in 3 weeks.

Learning Outcomes

After today's lecture, you will be able to:

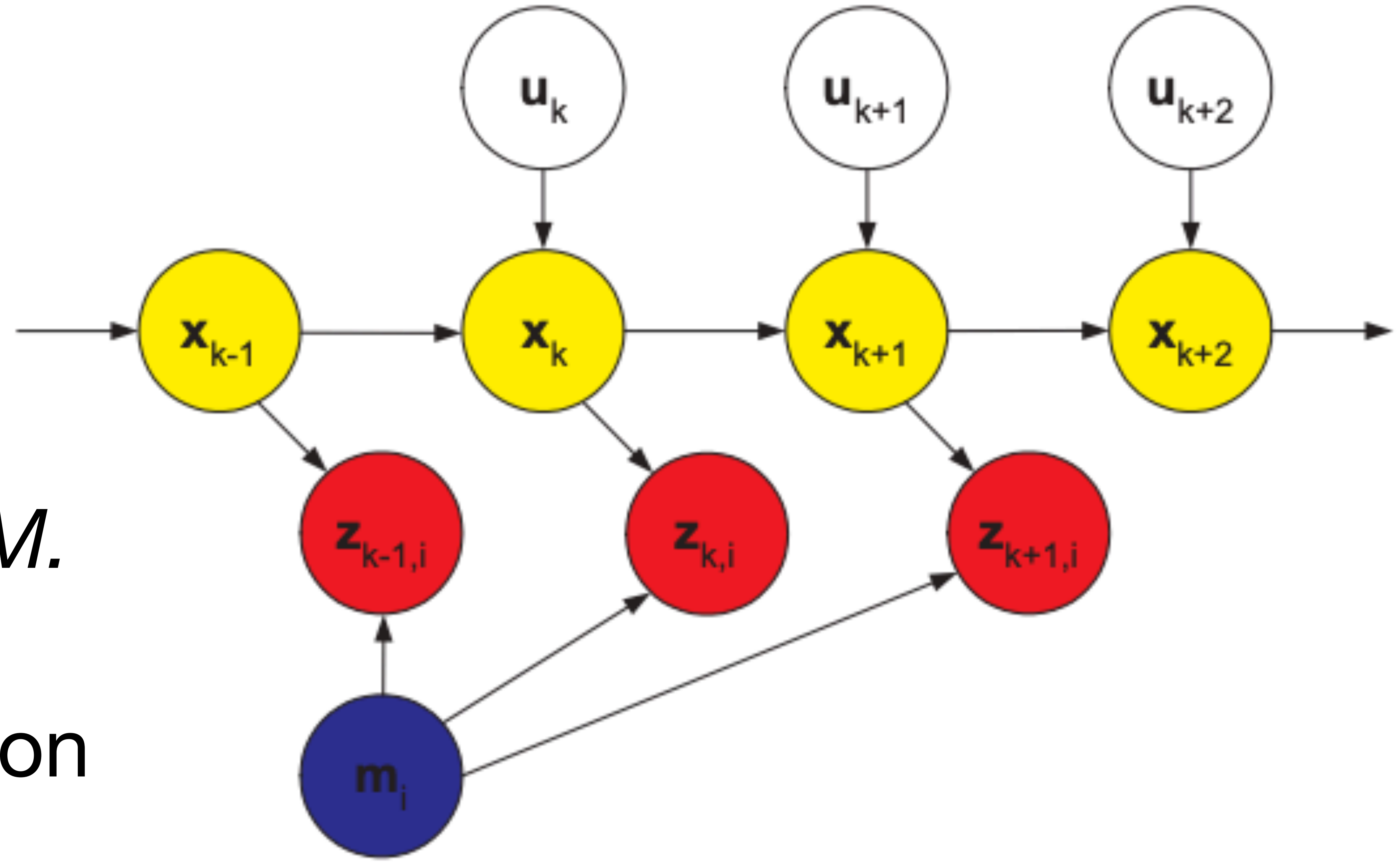
- Explain construction of the SLAM graph for GraphSLAM
- Derive the GraphSLAM objective as the maximum likelihood estimate of the robot's pose and map.
- Identify open questions in SLAM research and applications.

Odometry Data

- Odometry is a measurement of movement.
  - Example: $(\Delta x, \Delta y, \Delta \theta)$.
- Technically, a sensor observation but often treated as the robot's control. Why is this reasonable?
- If we know how far the robot has moved, then why must a robot localize?
  - Dead reckoning
- *Visual odometry*: determine change in position based on change of visual images.

SLAM (simple version)

- Localize and map at the same time.
- Formally, estimate $p(x_t, m \mid z_{1:t}, u_{1:t}, x_0)$
    - Or $p(x_{1:t}, m \mid z_{1:t}, u_{1:t}, x_0)$, i.e., *full SLAM.*
- Assume we have a motion and observation model:
    - $p(x_t \mid x_{t-1}, u_t)$ and $g(z_t \mid x_t, m)$.



Systems view of SLAM

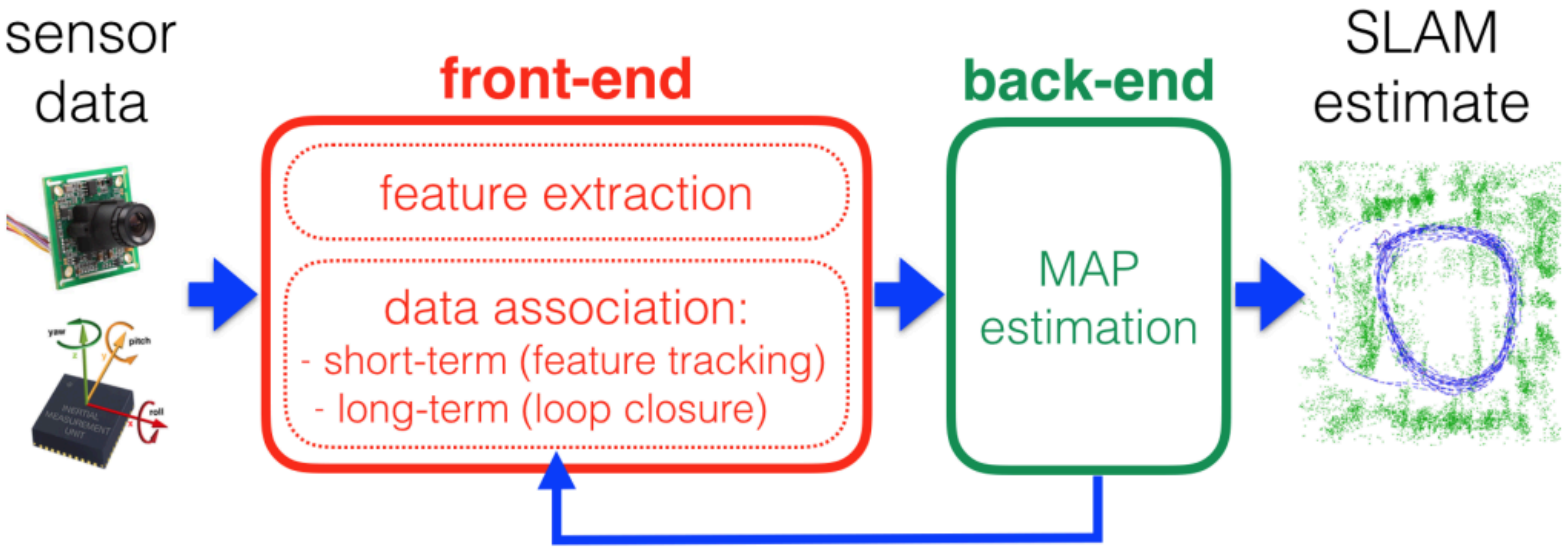


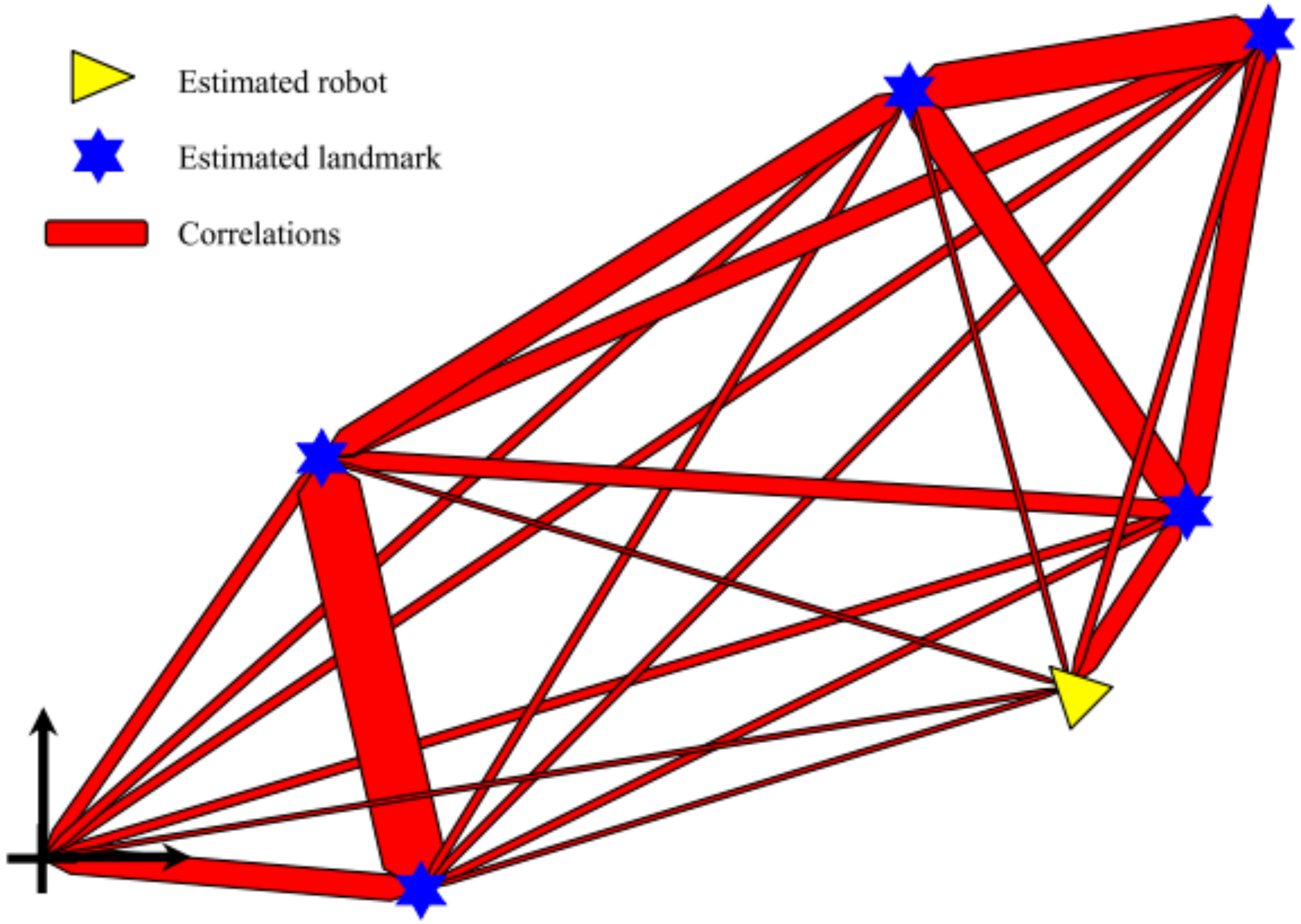
Fig. 2: Front-end and back-end in a typical SLAM system. The back-end can provide feedback to the front-end for loop closure detection and verification.

Back-end: Algorithm Review

- What are the strengths and weaknesses of:
  - EKF-SLAM
  - Particle Filtering SLAM
  - FastSLAM 1.0 and 2.0

The SLAM Graph

- Recall the spring analogy from last week's reading.
- Let's formalize this graph:
  - Nodes represent observed landmarks and robot poses.
  - Motion edges connect nodes for x_t and x_{t+1}.
  - Observation edges connect nodes x_t and z_t.



Graph SLAM

- Edges in SLAM graph create constraints:

Multi-dimensional Case

1-dimensional Case

Motion Constraint $(x_t - f(x_{t-1}, u_t))^\top Q^{-1}(x_t - f(x_{t-1}, u_t))$ $\quad \dfrac{(x_t - f(x_{t-1}, u_t))^2}{\sigma_Q}$

Measurement Constraint $(z_t - h(x_t, m_j))^\top R^{-1}(z_t - h(x_t, m_j))$ $\quad \dfrac{(z_t - h(x_t, m_t))^2}{\sigma_R}$

- Next, "move" nodes around to minimize sum of constraints.

$$J_{\texttt{GraphSLAM}}(x_{1:T}, m) = x_0^\top \Omega x_0 + \sum_{t=1}^{T} (x_t - f(x_{t-1}, u_t))^\top Q^{-1}(x_t - f(x_{t-1}, u_t)) + \sum_{t=1}^{T} \sum_{j} (z_t^j - h(x_t, m_j))^\top R^{-1}(z_t^j - h(x_t, m_j))$$

Moving nodes means optimizing $J_{\texttt{GraphSLAM}}$ with respect to the poses and landmark locations

Graph SLAM Optimization

- In practice, use iterative optimization to find the best position of nodes.
  - Start with a guess $x_{1:t}^0$ and m_0 and then improve guess w.r.t. $J_{\texttt{GraphSLAM}}$ until convergence.
- SLAM graph has special structure that enables fast solving.
  - Sparsity: Each node is only connected to a few other nodes. Why?
- Typically the optimization makes GraphSLAM an offline SLAM algorithm but extensions and fast solvers enable its use for online SLAM (e.g., iSAM).

$$J_{\texttt{GraphSLAM}}(x_{1:T}, m) = x_0^\top \Omega x_0 + \sum_{t=1}^{T} (x_t - f(x_{t-1}, u_t))^\top Q^{-1} (x_t - f(x_{t-1}, u_t)) + \sum_{t=1}^{T} \sum_j (z_t^j - h(x_t, m_j))^\top R^{-1} (z_t^j - h(x_t, m_j))$$

Practice Problem

- A mobile robot moves in its environment for 3 time-steps. At each time-step, it observes two landmarks. How many nodes and edges are there in the SLAM graph?
    - Consider the case when the landmarks are the same ones each time and when they are distinct.

Practice Problem

- A mobile robot moves in its environment for 3 time-steps. At each time-step, it observes two landmarks. How many nodes and edges are there in the SLAM graph?
  - Consider the case when the landmarks are the same ones each time and when they are distinct.

Same: 4 pose nodes, 2 landmark nodes, 3 movement edges, 6 observation edges

Distinct: 4 pose nodes, 6 landmark nodes, 3 movement edges, 6 observation edges

SLAM as Maximum Likelihood Estimation

- Taking a step back. Why is this optimization problem reasonable?
- Goal is to estimate most likely map and set of robot poses:

$$p(x_{1:T}, m \,|\, z_{1:T}, u_{1:T}, x_0) \propto p(z_{1:T} \,|\, x_{1:T}, m)p(x_{1:T} \,|\, x_0, u_{1:T})$$
$$= \prod_{t=1}^{T} p(x_t \,|\, x_{t-1}, u_t) \prod_{t=1}^{T} \prod_{j} g(z_t^j \,|\, x_t, m_j)$$

Bayes rule and Markov assumption

Maximizing likelihood (w.r.t. pose and map) is equivalent to maximizing log likelihood:

$$\sum_{t=1}^{T} \log p(x_t \,|\, x_{t-1}, u_t) + \sum_{t=1}^{T} \sum_{j} \log g(z_t^j \,|\, x_t, m_j)$$

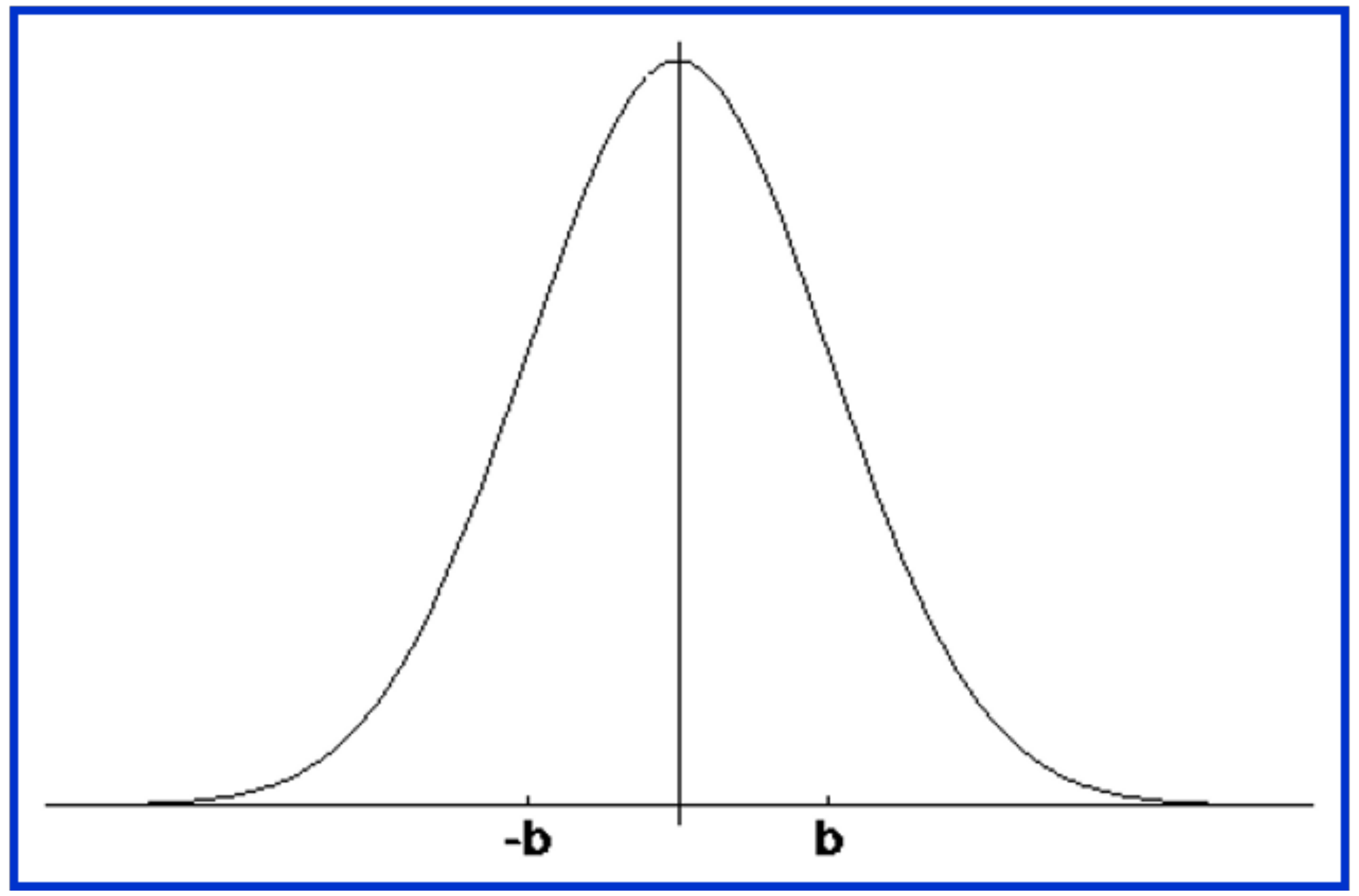
Assume motion and measurement models follow EKF assumptions (see lecture 6 slides):

$$J_{\texttt{GraphSLAM}}(x_{1:T}, m) = x_0^\top \Omega x_0 + \sum_{t=1}^{T} (x_t - f(x_{t-1}, u_t))^\top Q^{-1}(x_t - f(x_{t-1}, u_t)) + \sum_{t=1}^{T} \sum_{j} (z_t^j - h(x_t, m_j))^\top R^{-1}(z_t^j - h(x_t, m_j))$$

Review: Gaussian Distributions

Univariate ($x \in \mathbf{R}$)

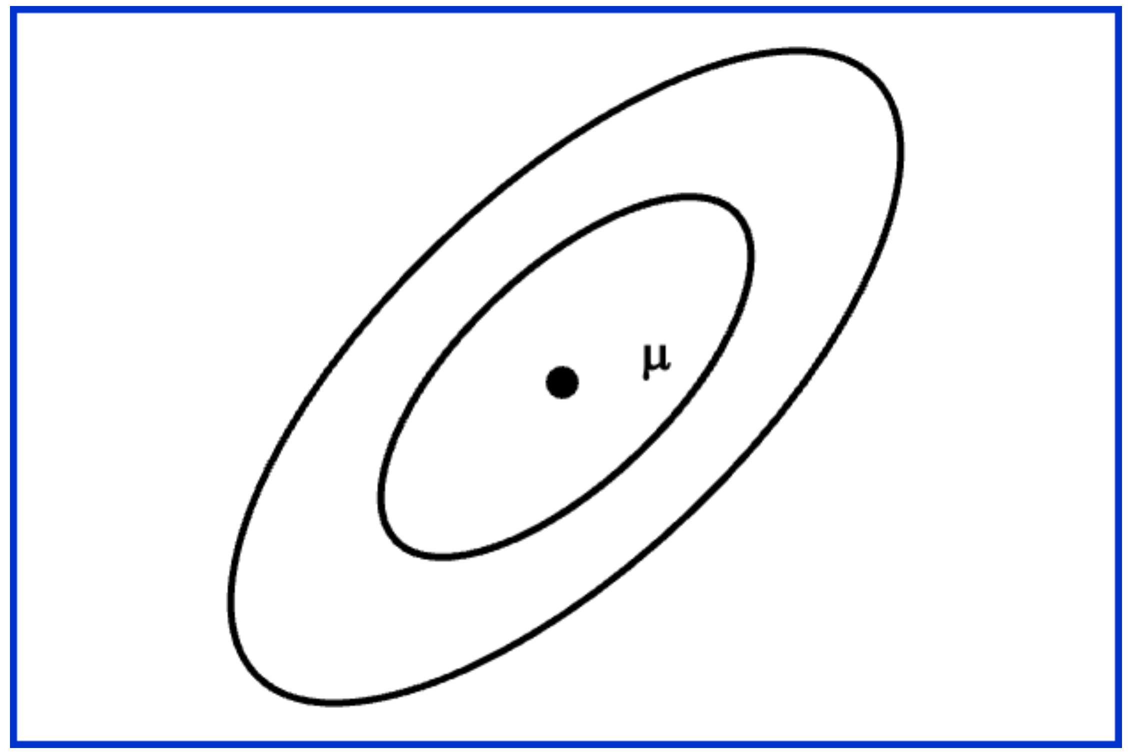
$$x \sim \mathcal{N}(\mu, \sigma^2) \qquad p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



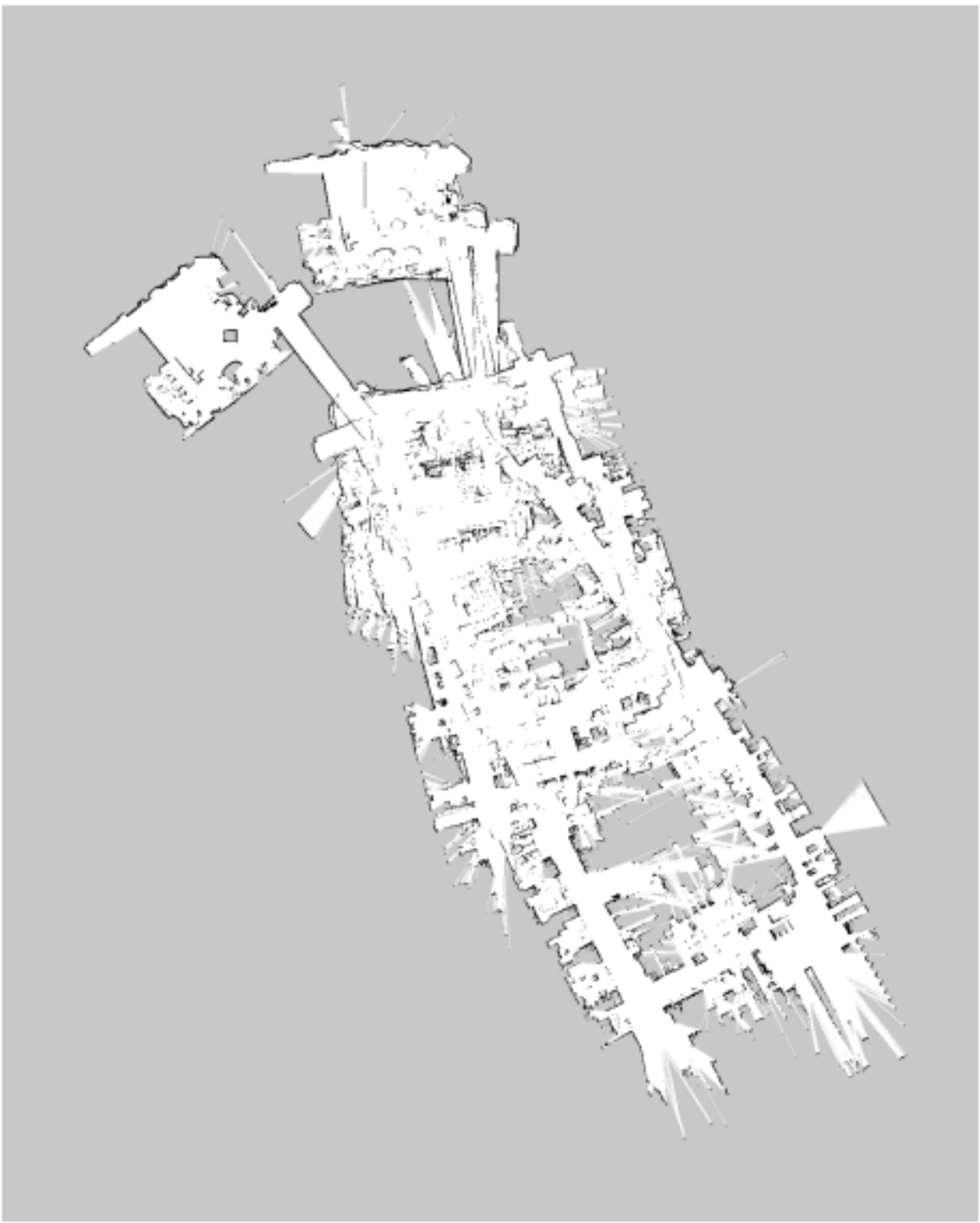
Multivariate ($x \in \mathbf{R}^d$)

$$x \sim \mathcal{N}(\mu, \Sigma)$$

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

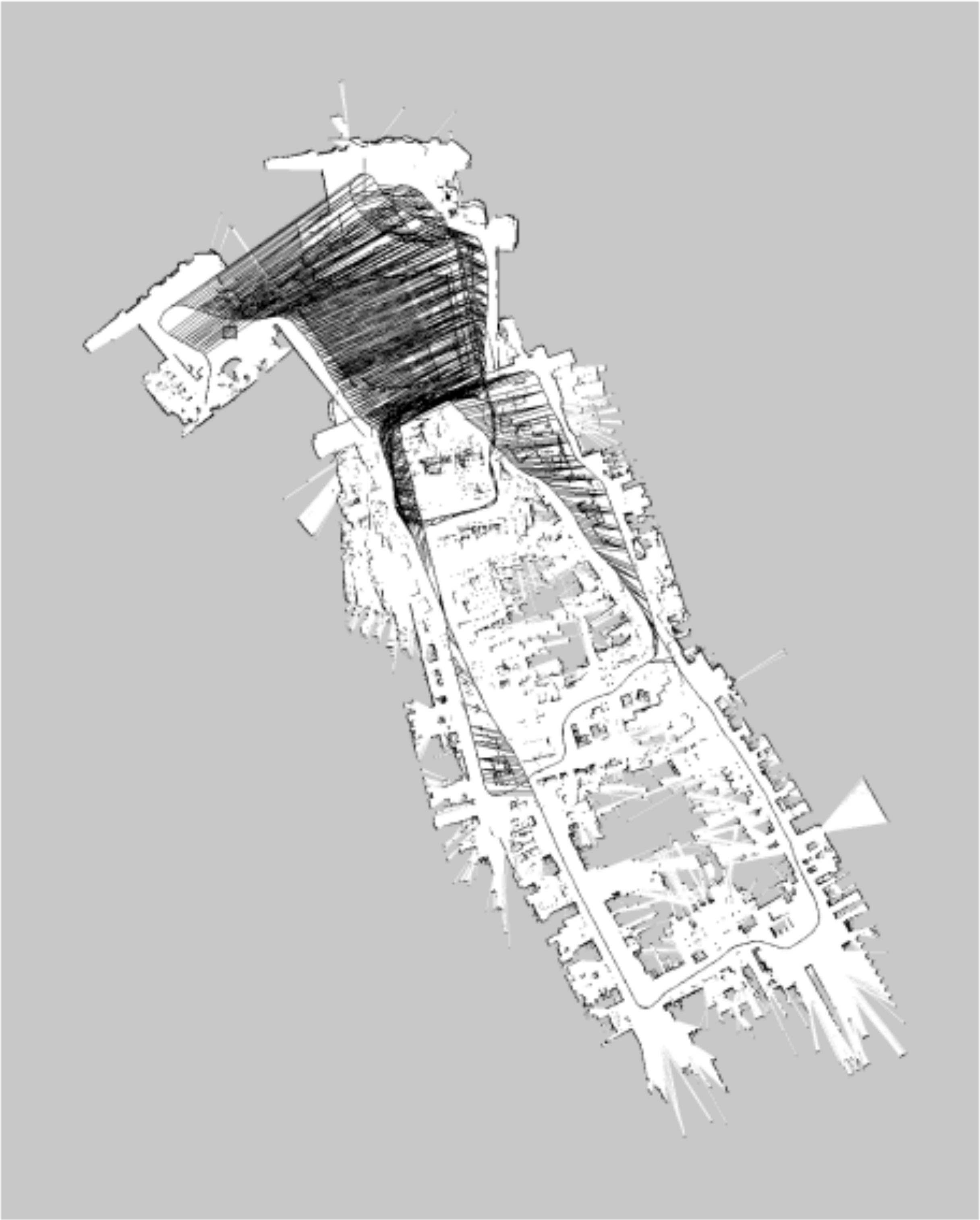


SLAM as Maximum Likelihood Estimation



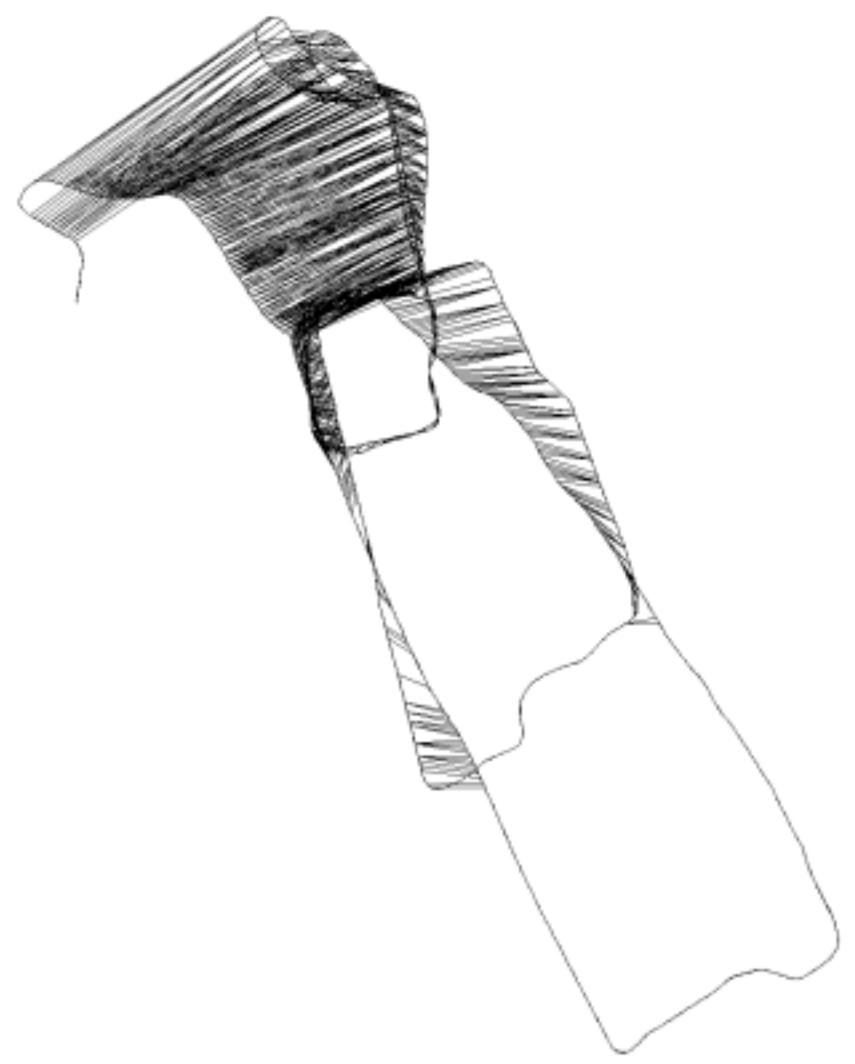
[KUKA Hall 22, courtesy P. Pfaff & G. Grisetti]

SLAM as Maximum Likelihood Estimation



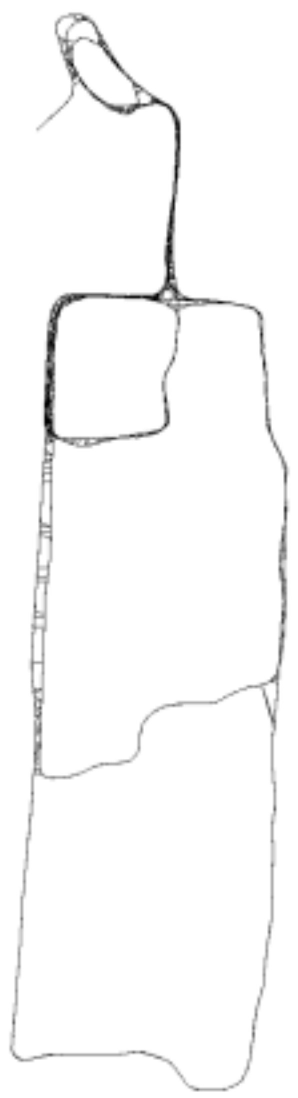
[KUKA Hall 22, courtesy P. Pfaff & G. Grisetti]

SLAM as Maximum Likelihood Estimation



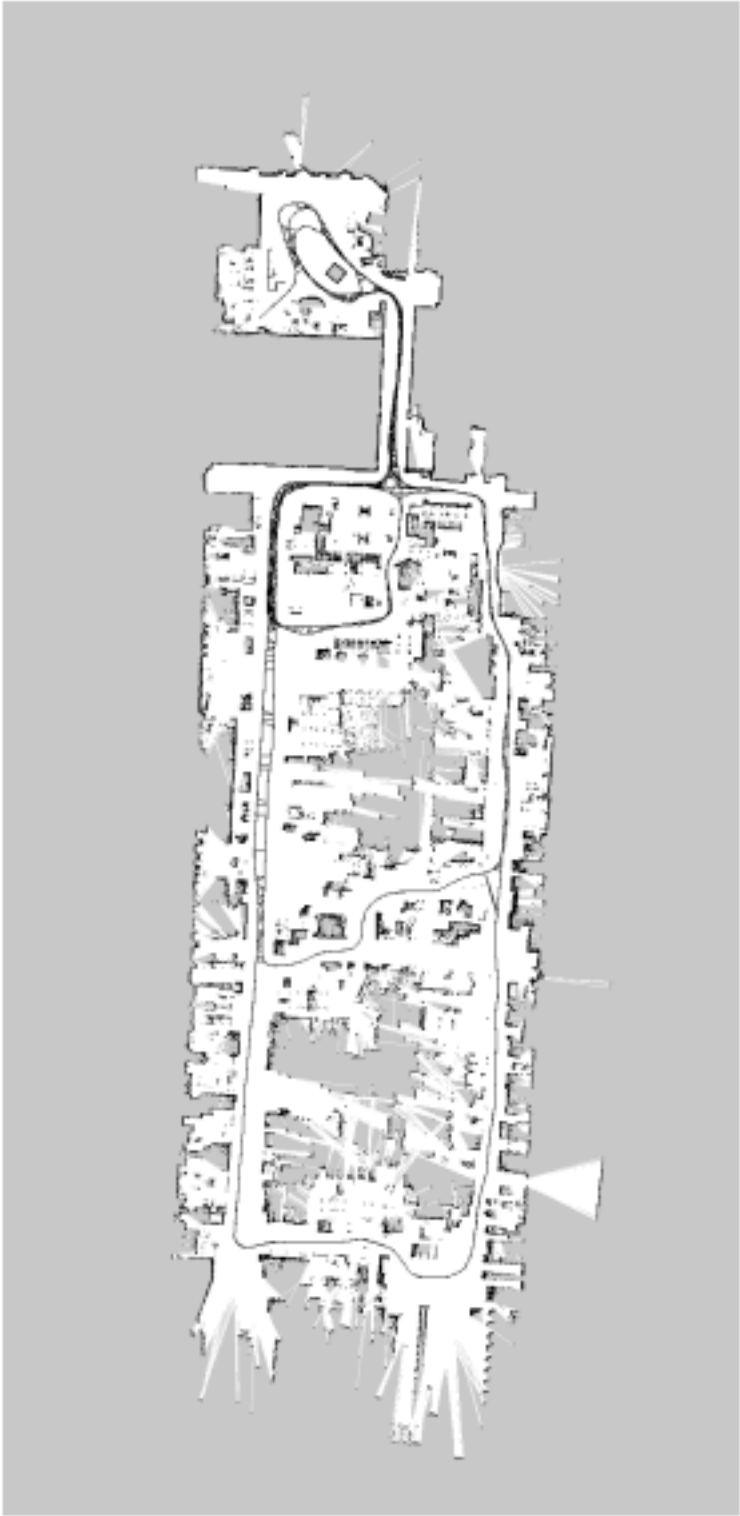
[KUKA Hall 22, courtesy P. Pfaff & G. Grisetti]

SLAM as Maximum Likelihood Estimation



[KUKA Hall 22, courtesy P. Pfaff & G. Grisetti]

SLAM as Maximum Likelihood Estimation



[KUKA Hall 22, courtesy P. Pfaff & G. Grisetti]

Comparing EKF-SLAM to GraphSLAM

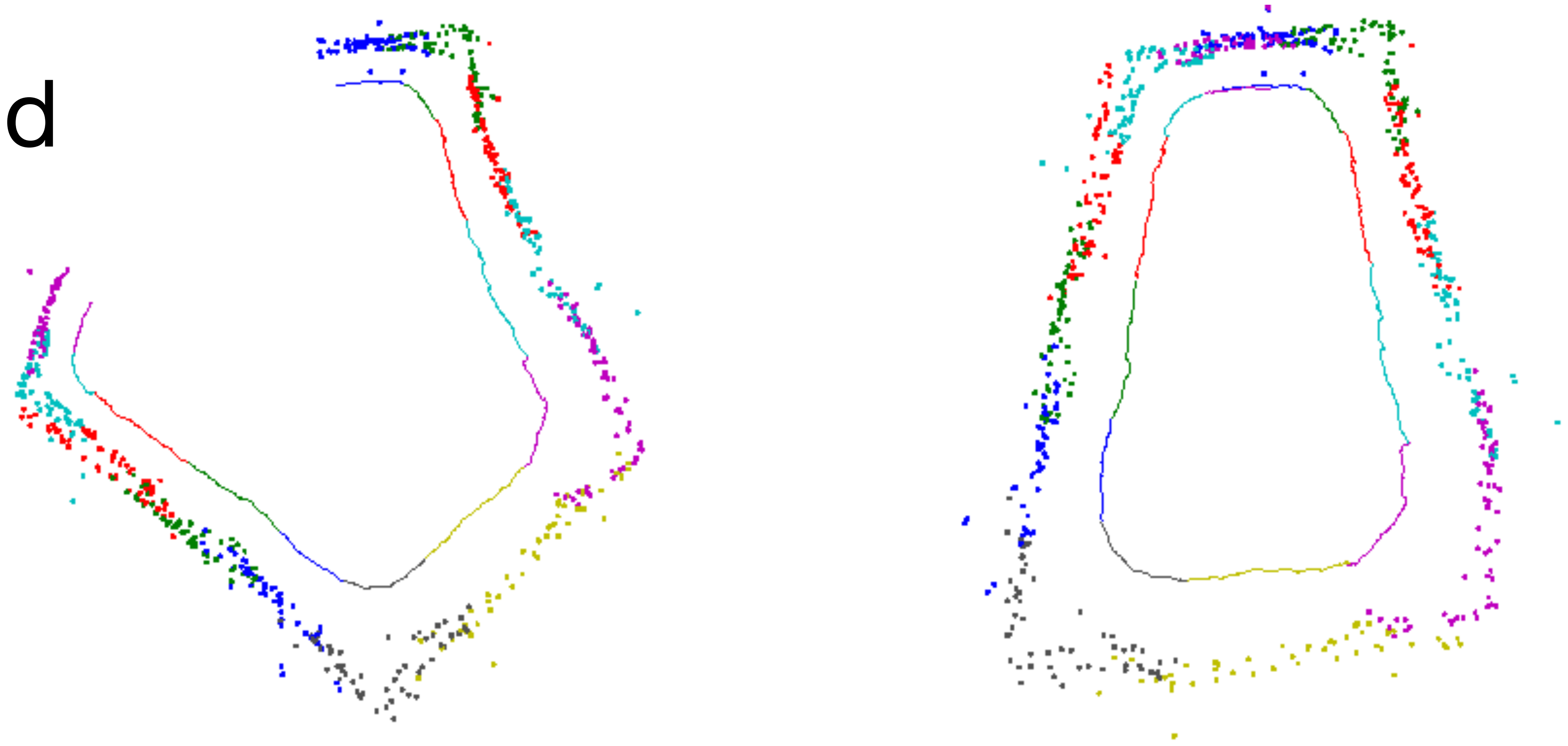
- EKF-SLAM and GraphSLAM make similar assumptions. What are the differences?
  - Filtering vs. optimization (like smoothing) approach.
  - GraphSLAM will have higher memory requirement.
  - GraphSLAM will give an overall more accurate global estimate.

Active SLAM

- So far we have ignored the choice of the robot's controls, but they are critical for optimal mapping.
  - If the robot stays in one area or only visits known areas, then the map will not improve.
- Where to go next? What should a robot consider?
  - Discovering new parts of the map.
  - Maintaining its own belief about where it is.
- How to determine where to go?
  - Optimal design
  - Information gain
  - Reinforcement learning / POMDP planning

Robustness

- Hardware failure
- Robustness to incorrect loop closure
- Dynamic maps
- Integration of the front-end and back-end



Scalability

- Need to forget?
- Handling new types of sensors?
  - E.g., event cameras
- Integrating semantic information
- Handling resource constraints

Summary

- Introduced GraphSLAM
- Discussed extensions and future directions for the SLAM problem.

Action Items

- Kinematics reading for next week; send a reading response by 12 pm on Monday.

- SLAM assignment due in 1 week.