CSE 571Robotics

Recap and Discussion

•1

Goal of this course

- Provide an overview of fundamental problems / techniques in robotics
- Understanding of estimation and decision making in dynamical systems
  - Probabilistic modeling and filtering
  - Deterministic and non-deterministic planning
  - Learning for perception and modeling
- Augment model-based understanding with hands-on experience in deep learning

•2

Bayesian Filtering, Models

ESTIMATION

•3

Bayes Filters

z = observation
u = action
x = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

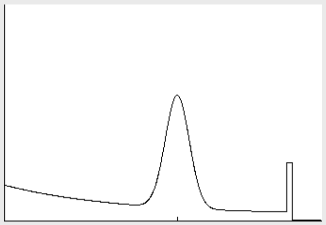
Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

$$\boxed{= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}}$$

•4

Parametric Sensor Model



$$P(z \mid x,m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\max} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x,m) \\ P_{\text{unexp}}(z \mid x,m) \\ P_{\max}(z \mid x,m) \\ P_{\text{rand}}(z \mid x,m) \end{pmatrix}$$

5

•5

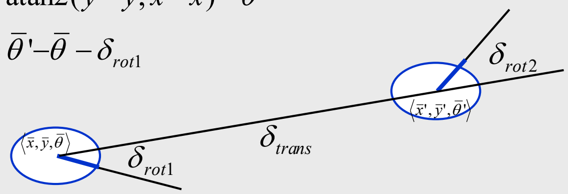
Parametric Kinematics Model

- **Robot moves from** $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ **to** $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$.
- **Odometry information** $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$.

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



6

•6

The Prediction-Correction-Cycle of Kalman Filters

Prediction

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

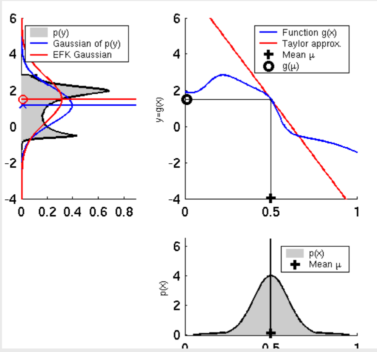
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Correction

7

•7

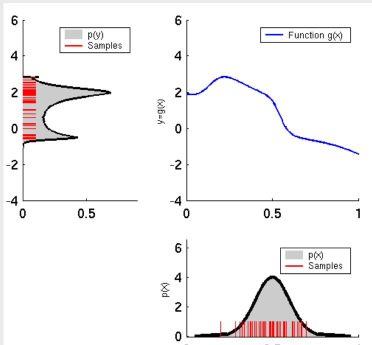
EKF Linearization



8

•8

Particle Filter Projection

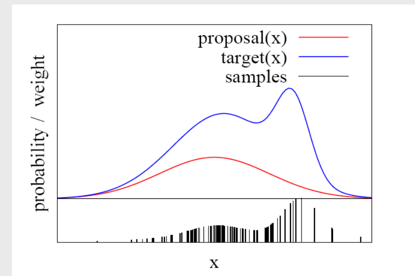


•9

Importance Sampling Principle

- We can use a different distribution g to generate samples from f
- By introducing an importance weight w, we can account for the "differences between g and f"

$$w = f / g$$



10

•10

SLAM

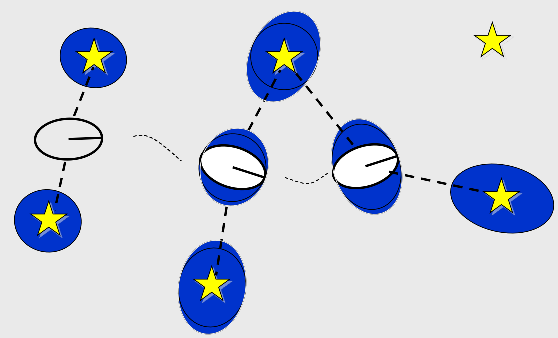
ESTIMATION

11

•11

Why is SLAM a hard problem?

•**SLAM: robot path and map are both unknown**



•**Robot path error correlates errors in the map**

12

•12

EKF-SLAM

- Map with N landmarks:(3+2N)-dimensional Gaussian

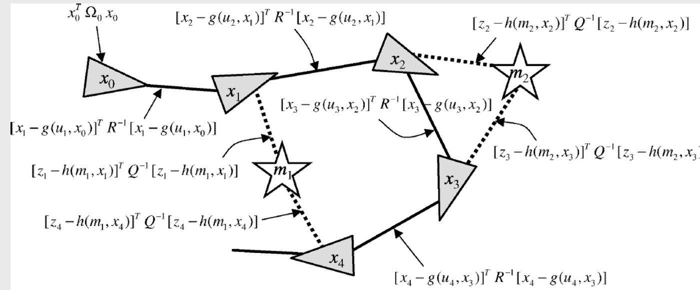
$$
Bel(x_t, m_t) = \left\langle \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N} & \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{pmatrix} \right\rangle
$$

- Can handle hundreds of dimensions

13

•13

Graph-SLAM Idea



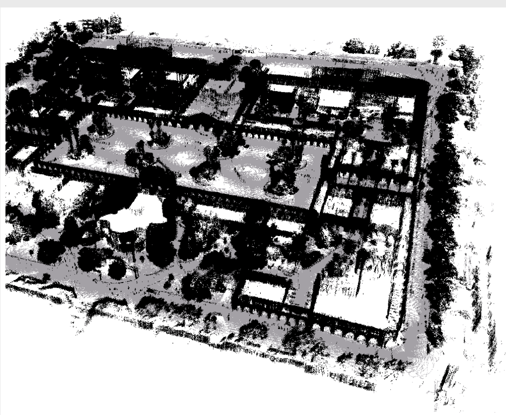
<u>Sum of all constraints:</u>

$$
J_{\text{GraphSLAM}} = x_0^T \Omega_0 x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]^T Q^{-1} [z_t - h(m_{c_t}, x_t)]
$$

14

•14

3D Outdoor Mapping



10^8 features, 10^5 poses, only few secs using cg.

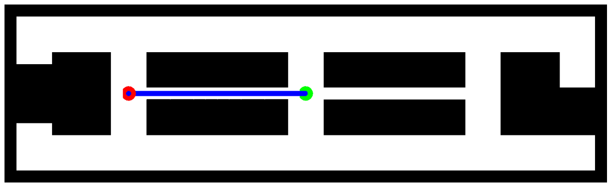
15

•15

PLANNING / CONTROL

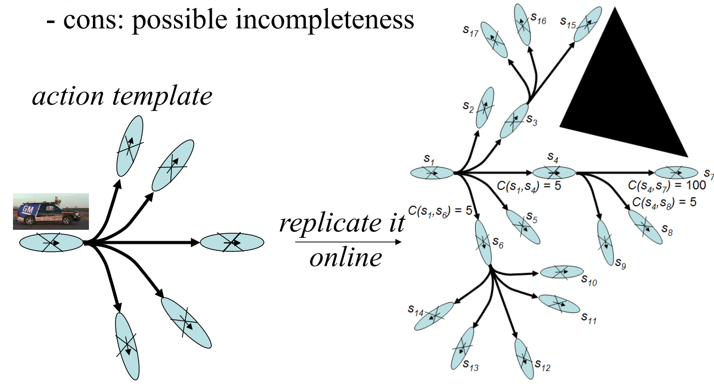
•16

Deterministic, fully observable



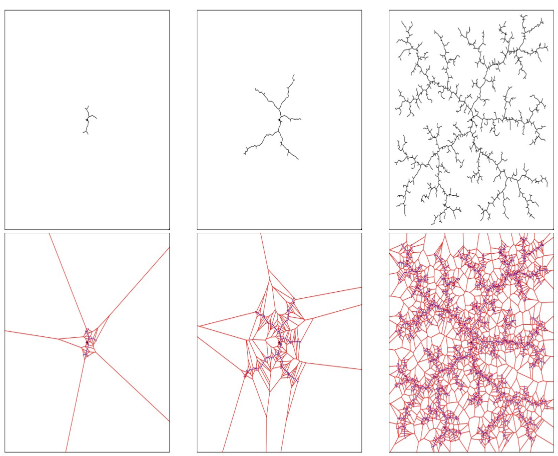
Planning via Cell Decomposition

- Graph construction:
  - lattice graph
  - pros: sparse graph, feasible paths
  - cons: possible incompleteness

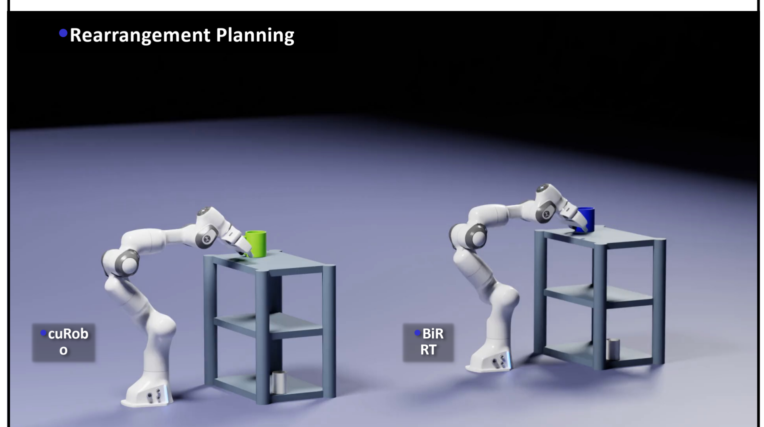


CSE-571: Courtesy of Maxim Likhachev, CMU

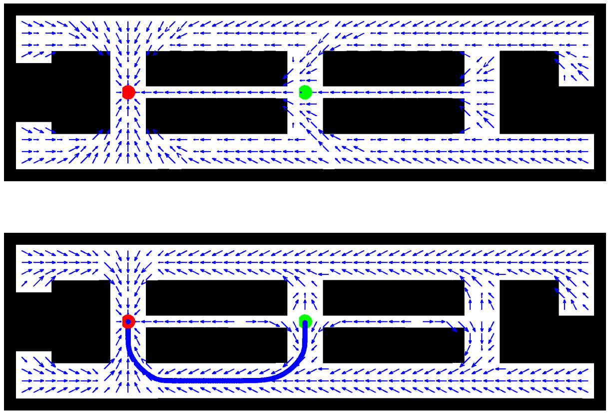
Rapidly exploring Random Tree (RRT)



Source: LaValle and Kuffner 01



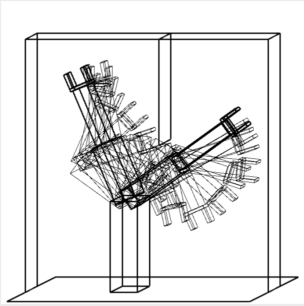
Stochastic, Fully Observable



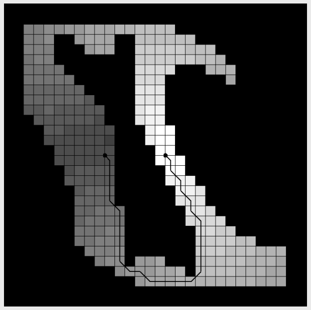
•21

•21

Manipulator Control Path



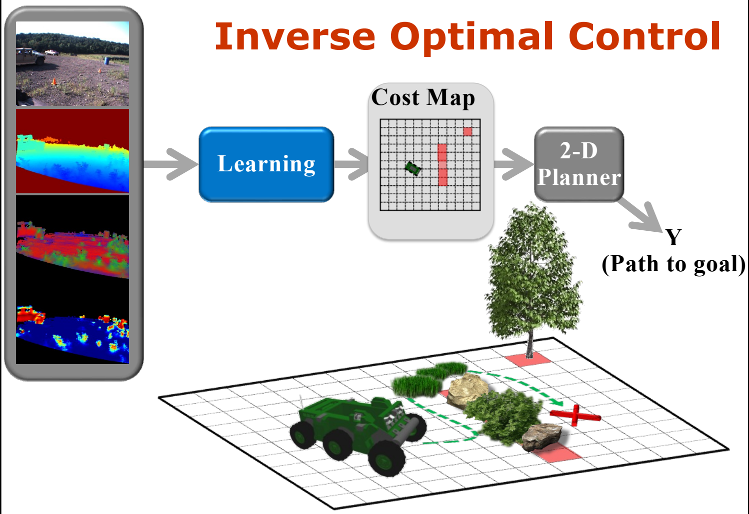
Work space



Configuration space

•22

Inverse Optimal Control



•23

Where is RobotGPT?

Learning at scale enables powerful models for images, videos, and language, but truly capable RobotGPT still has a way to go

Leveraging VLMs for robotics provides exciting initial capabilities but quickly hits ceiling

Robotics must play active role in developing open-world vision-language-action models by

- driving the development of models that have robust understanding of 3D geometric reasoning, physical interaction, planning, and the combination of high-level reasoning and low-level control
- developing benchmarks and test protocols geared toward real world robotics relevance

Simulation has the potential to

- generate the kind of data necessary to train foundation manipulation capabilities (diversity and scaling via Gen-AI-based asset, scene, and task generation; privileged information enables generation of demonstrations via TAMP, control, RL)
- benchmark models and techniques, enabling community to make measurable progress
- broadening community participation due to manageable cost
- **Open issues**: assets, sim2real, real2sim, non-rigid objects and complex interactions

•24