CSE 571
Probabilistic Robotics

Recap

Bayesian Filtering, Models

ESTIMATION

2

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

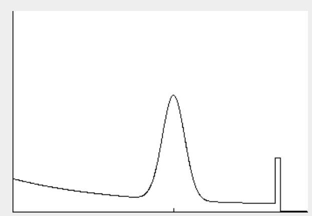
Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

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

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

3

Parametric Sensor Model



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

4

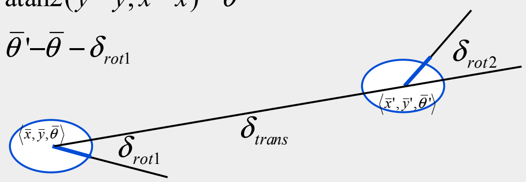
Parametric Kinematics Model

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

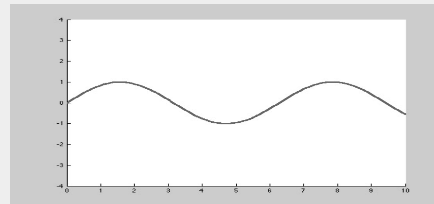
$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

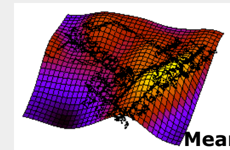
$$\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

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

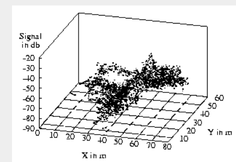


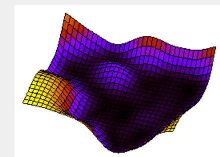
Alternative: Non-Parametric Gaussian Process Models





Mean





Variance

Representations for Bayesian Robot Localization

•Discrete approaches ('95)
- Topological representation ('95)
  - uncertainty handling (POMDPs)
  - occas. global localization, recovery
- Grid-based, metric representation ('96)
  - global localization, recovery

•Kalman filters (late-80s?)
- Gaussians
- approximately linear models
- position tracking

•Particle filters ('99)
- sample-based representation
- global localization, recovery

•Multi-hypothesis ('00)
- multiple Kalman filters
- global localization, recovery

The Prediction-Correction-Cycle of Kalman Filters

Prediction

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

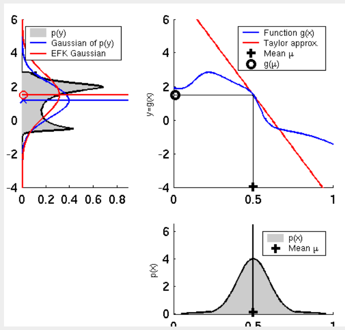
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Correction

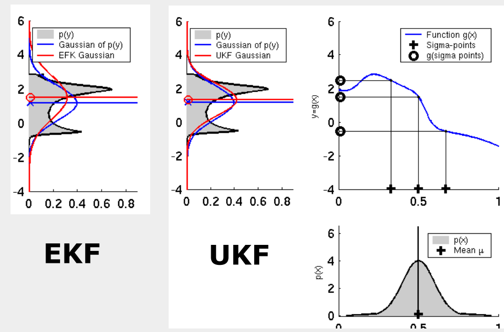
$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t (z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t) \bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

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

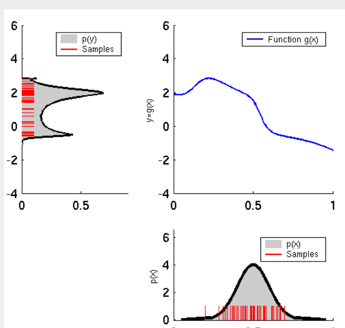
EKF Linearization



UKF Linearization



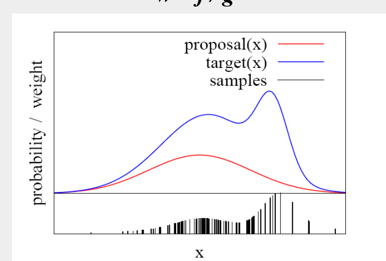
Particle Filter Projection



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"

$$w = f / g$$



SLAM

ESTIMATION

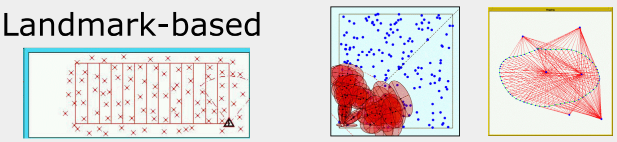
13

Types of SLAM-Problems

Grid maps or scans

[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

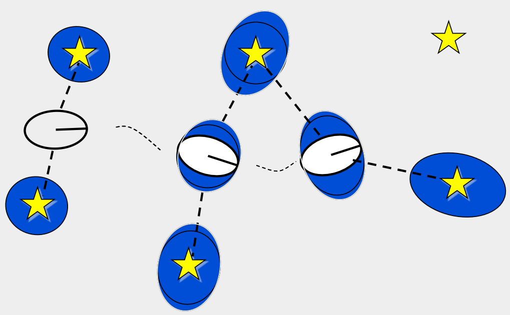
Landmark-based



14

Why is SLAM a hard problem?

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



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

15

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

16

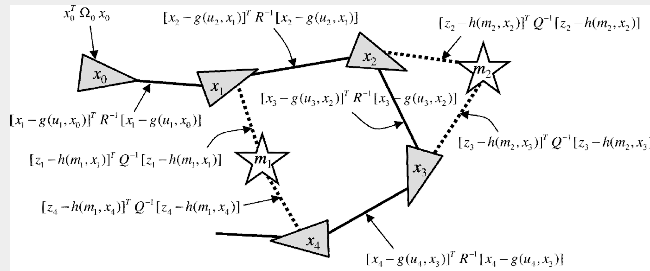
FastSLAM

	Robot Pose	2 x 2 Kalman Filters			
Particle #1	x, y, z	Landmark 1	Landmark 2	…	Landmark N
Particle #2	x, y, z	Landmark 1	Landmark 2	…	Landmark N
Particle #3	x, y, z	Landmark 1	Landmark 2	…	Landmark N
⋮					
Particle M	x, y, z	Landmark 1	Landmark 2	…	Landmark N

17

[Courtesy of Mike Montemerlo]

Graph-SLAM Idea

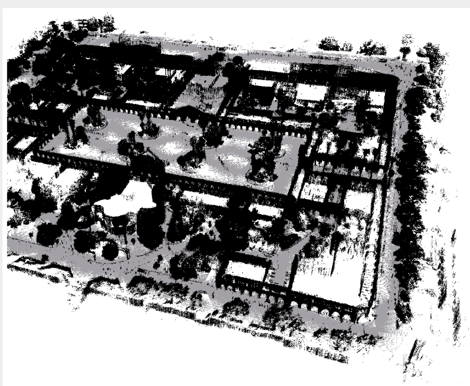


Sum of all constraints:

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

18

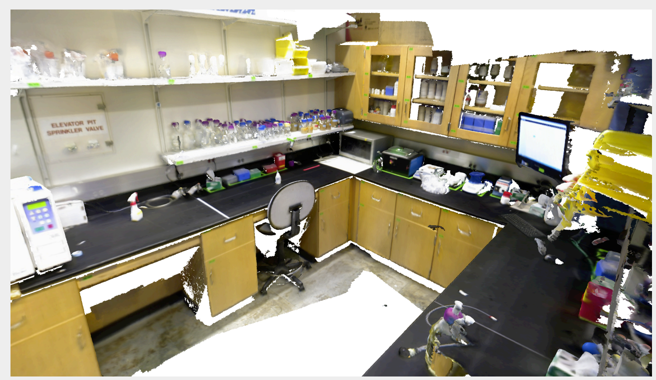
3D Outdoor Mapping



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

19

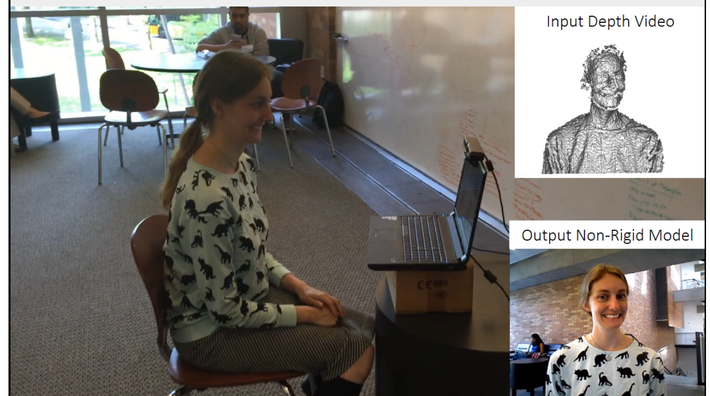
RGB-D Mapping



20

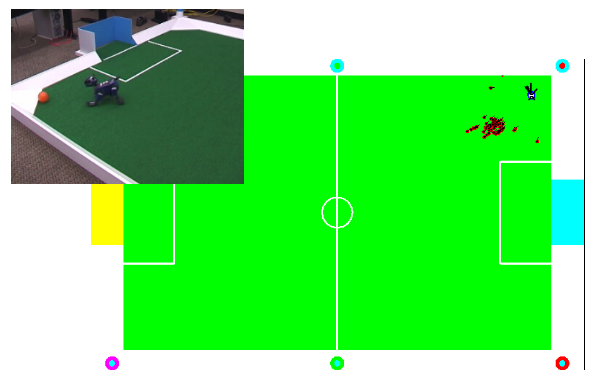
RGB-D Mapping

21

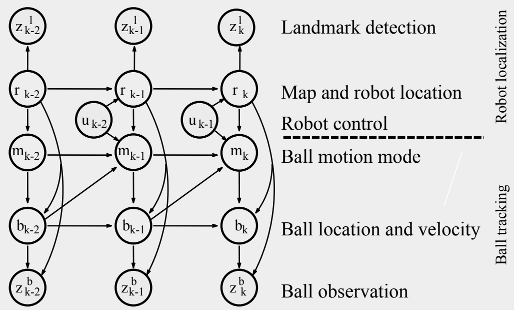


MORE COMPLEX ESTIMATION

Ball-Environment Interaction

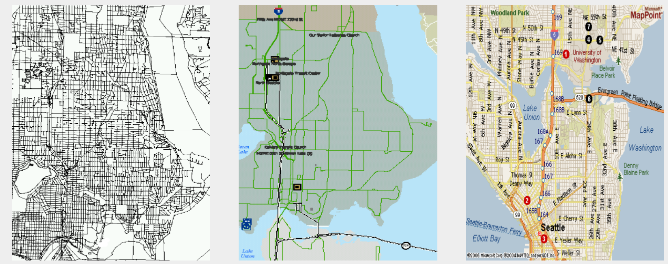


Inference: Posterior Estimation

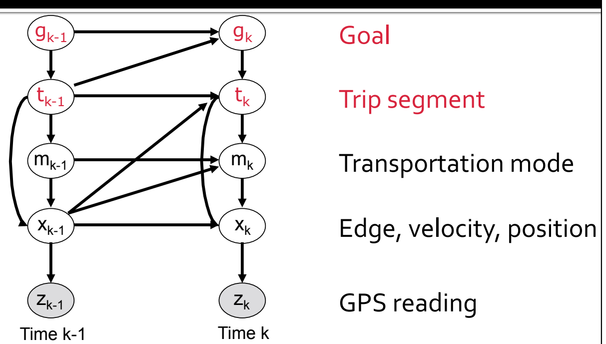


$$\boxed{p(b_k, m_k, r_k \mid z^b_{1:k}, z^l_{1:k}, u_{1:k-1})}$$

Geographic Information Systems



Hierarchical Model



Particles: $s^{(i)} = \left\langle \langle g,t \rangle^{(i)}, m^{(i)}, e^{(i)}, v^{(i)}, \theta^{(i)}, N^{(i)}(\mu,\sigma^2) \right\rangle$

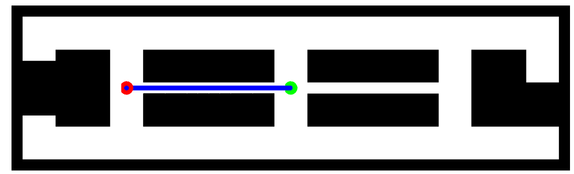
PLANNING / CONTROL

Coordinated exploration with three robots
from unknown start locations

The robots are fully autonomous.
All computation is performed on–board.

Shown is the perspective of one robot

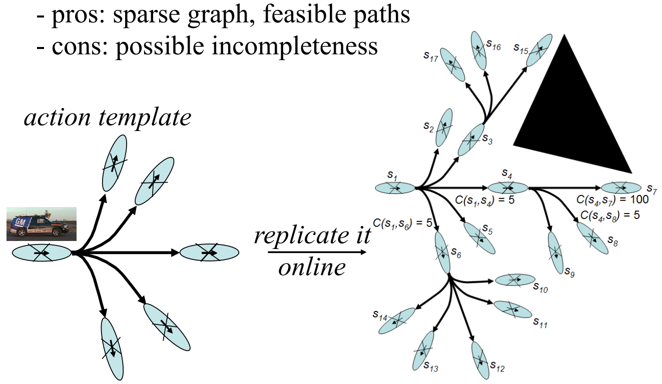
Deterministic, fully observable



•30

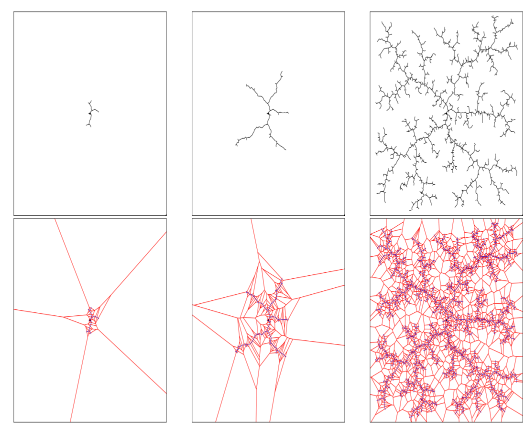
Planning via Cell Decomposition

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



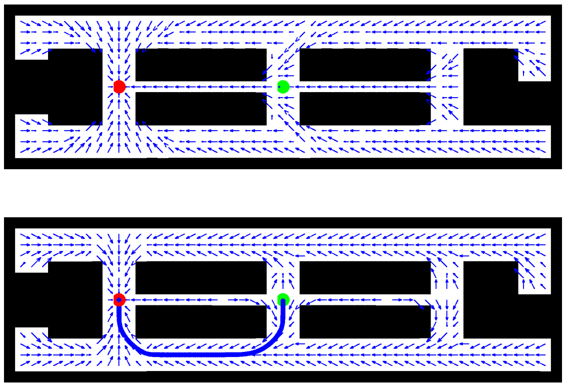
CSE-571: Courtesy of Maxim Likhachev, CMU

Rapidly exploring Random Tree (RRT)



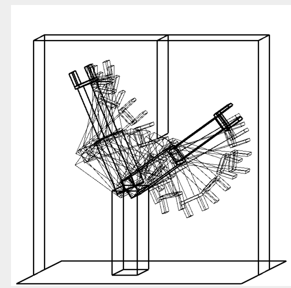
Source: LaValle and Kuffner 01

Stochastic, Fully Observable

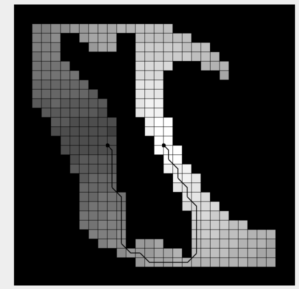


•33

Manipulator Control Path

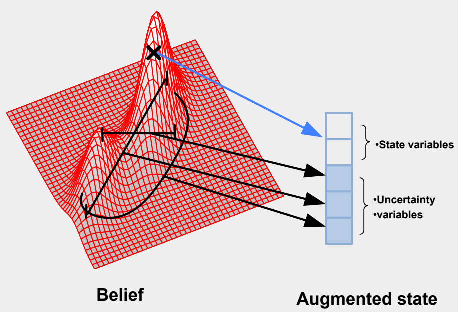


State space

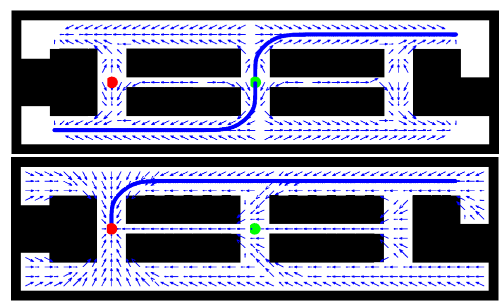


Configuration space

RL in Uncertain Environments: Converting Beliefs to Augmented States

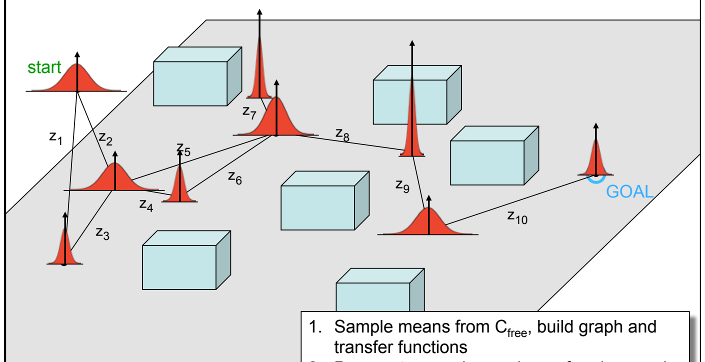


Stochastic, Partially Observable



•36

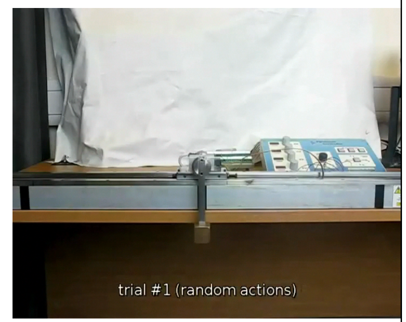
The Belief Roadmap Algorithm



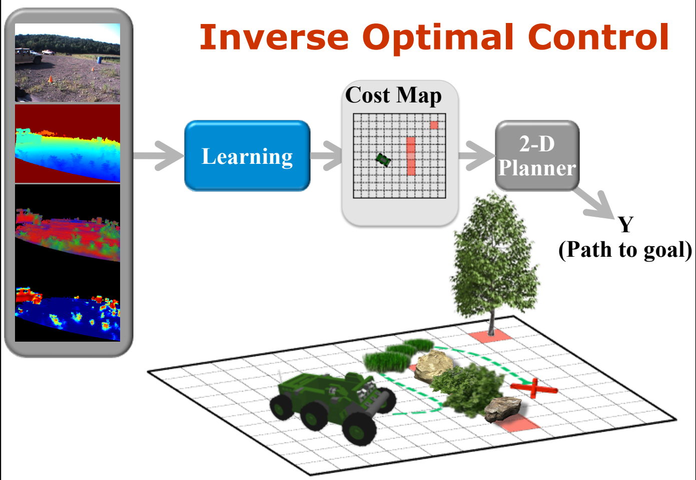
1. Sample means from C_{free}, build graph and transfer functions
2. Propagate covariances by performing graph search

PILCO: GP Model-Based Learning

- Swing pendulum up and balance in inverted position
- Learn nonlinear control from scratch
- 4D state space, 300 controll parameters
- **7 trials/17.5 sec experience**
- Control freq.: 10 Hz



Inverse Optimal Control



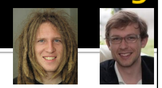
Further Examples and Discussion

40

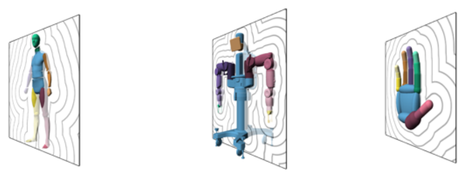
[Schmidt-Newcombe-F: RSS-14]

DART: Dense Articulated Real-Time Tracking

Using Articulated Signed-Distance Functions



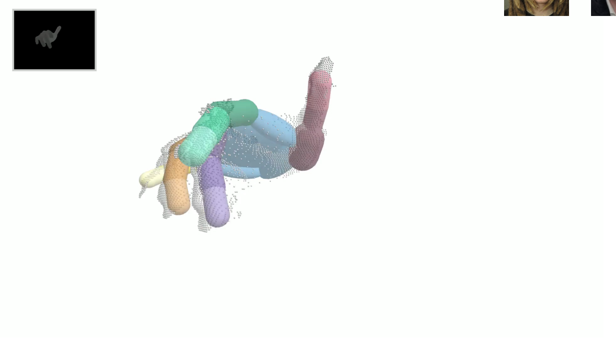
- Goal: General tool for real-time tracking of arbitrary articulated objects
- Input: Shape models of parts along with joint structure
- Insight: Efficient optimization via articulated signed distance functions



[Schmidt-Newcombe-F: RSS-14, AR-15]

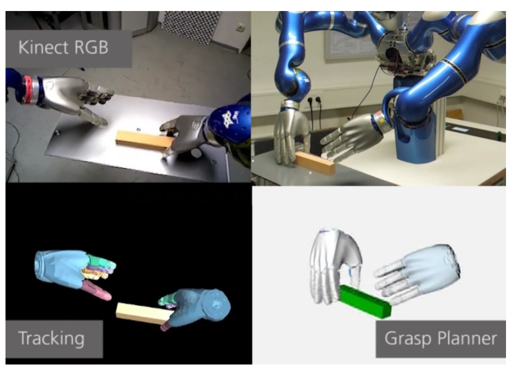
DART: Model-Based Tracking

Hand (27 DoF) and Human Body (42 DoF

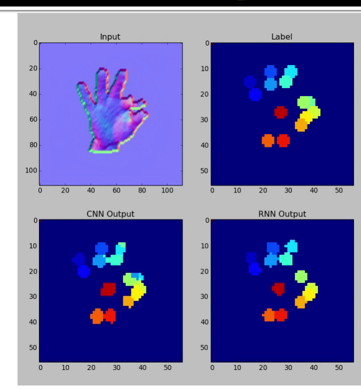


[Schmidt-etal: ICRA-15]

Fine-Grained Manipulation



Detection-Based Approach to Articulated Tracking



Inspired by [Tompson-etal: SIGGRAPH-14]

Model-Based / Detection-Based

	Model-based	Detection-based
Generality	Minimal assumptions, broad applicability	Only in trained regime
Robustness	Requires initialization and good model; Can detect failures	Robust in trained regime; Failure detection more difficult
Training	Minimal training, model building	Major training effort ; Model-based for supervision
Physics, contacts	Explicitly modeled	Must be learned in data driven way
Efficiency	Efficient for local tracking; Initialization extremely hard	Highly efficient once trained

[Bollini-Rus]

BakeBot

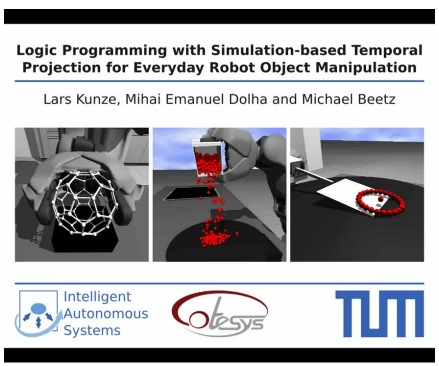
BakeBot:
Motion Planning for Cooking

Mario Bollini and Daniela Rus
CSAIL, MIT

CSAIL

[Kunze-Dolha-Beetz]

Physics Simulation for Prediction



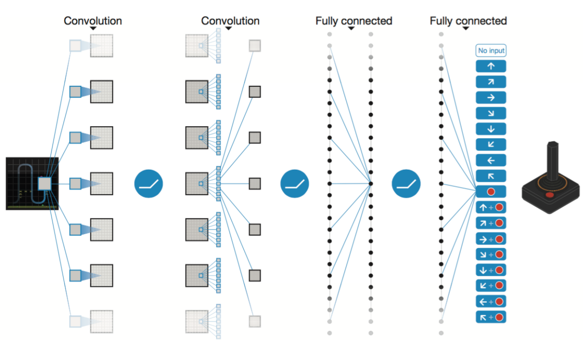
How Julia Does it

Gravity and Onions

Deep RL

Human-level control through deep reinforcement learning

Q-Network



Learning Models from Raw Perception

[Finn-Tan-Duan-Darrell-Levine-Abbeel]

Deep Spatial Autoencoders for Visuomotor Learning

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, Pieter Abbeel

Summary

- Probabilistic robotics
  - Great framework for approaching robotics problems
  - Not always possible, or the most appropriate approach
- On models and learning
  - Models are great but never perfect
  - Especially when reasoning about messy stuff
  - Learning of residuals, shortcuts, everything?
  - Combination is key