Announcements

- PS2: due Friday 23:59pm.
- Final project: 45% of the grade, 10% presentation, 35% write-up
  - Presentations: in lecture Dec 1 and 3 --- schedule:

CS 287: Advanced Robotics
Fall 2009

Lecture 24:
SLAM

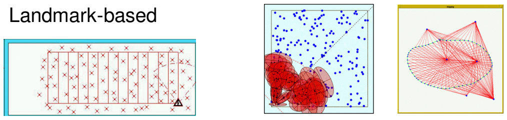
Pieter Abbeel
UC Berkeley EECS

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



[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;...

Recap Landmark based SLAM

- State variables:
  - Robot pose
  - Coordinates of each of the landmarks
- Robot dynamics model: P(x_{t+1} | x_t, u_t)
- Sensor model: P(z_{t+1} | x_t, m)
  - Probability of landmark observations given the state
- Can run EKF, SEIF, various other approaches
- → Result: path of robot, location of landmarks

KF-type approaches are a good fit b/c they can keep track of correlations between landmarks

Note: Could then use path of robot + sensor log and build a map assuming known robot poses

Grid-based SLAM

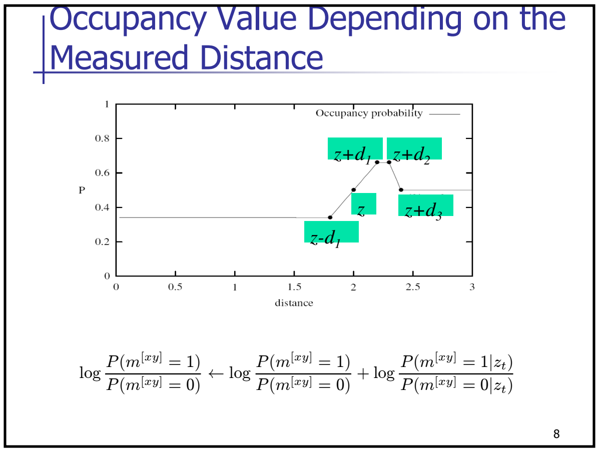
- Can we solve the SLAM problem if no pre-defined landmarks are available?
- As with landmarks, the map depends on the poses of the robot during data acquisition
- If the poses are known, grid-based mapping is easy ("mapping with known poses")

Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid.
- Estimate the probability that a location is occupied by an obstacle.
- Key assumptions
  - Occupancy of individual cells (m[xy]) is independent

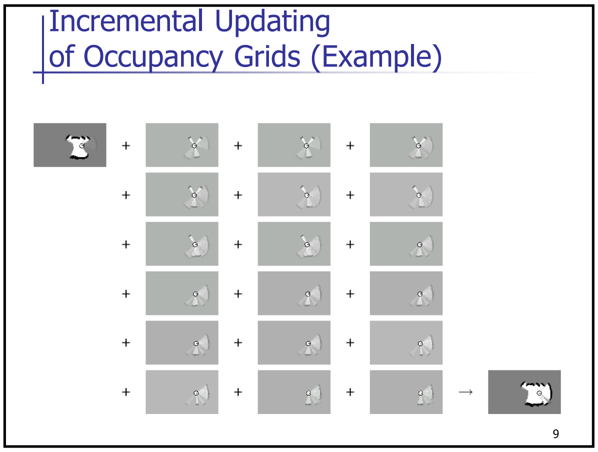
$$Bel(m_t) = P(m_t \mid u_1, z_2 \ldots, u_{t-1}, z_t) \\ = \prod_{x,y} Bel(m_t^{[xy]})$$

  - Robot positions are known!

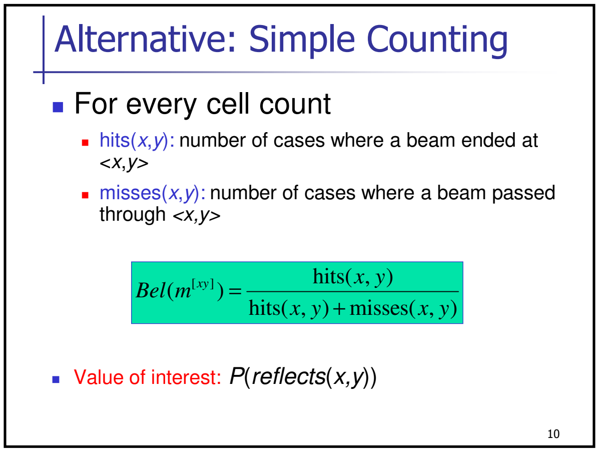


Occupancy Value Depending on the Measured Distance

$$\log\frac{P(m^{[xy]}=1)}{P(m^{[xy]}=0)} \leftarrow \log\frac{P(m^{[xy]}=1)}{P(m^{[xy]}=0)} + \log\frac{P(m^{[xy]}=1|z_t)}{P(m^{[xy]}=0|z_t)}$$



Incremental Updating of Occupancy Grids (Example)

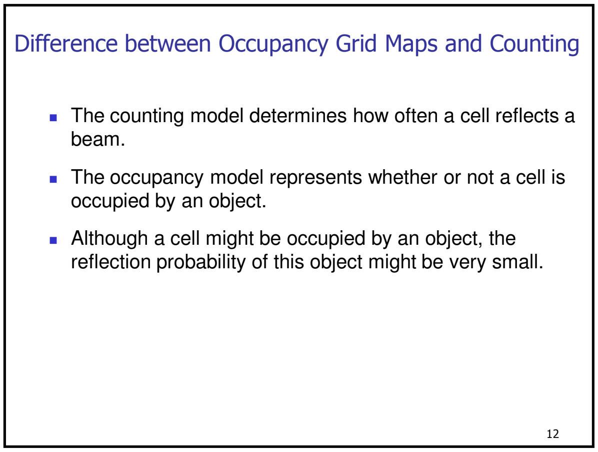


Alternative: Simple Counting

- For every cell count
  - hits(*x*,*y*): number of cases where a beam ended at <*x*,*y*>
  - misses(*x*,*y*): number of cases where a beam passed through <*x*,*y*>

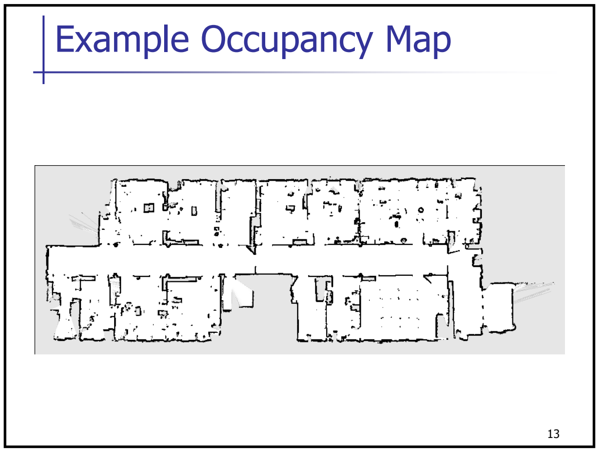
$$Bel(m^{[xy]}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y)+\text{misses}(x,y)}$$

- Value of interest: *P*(*reflects*(*x*,*y*))

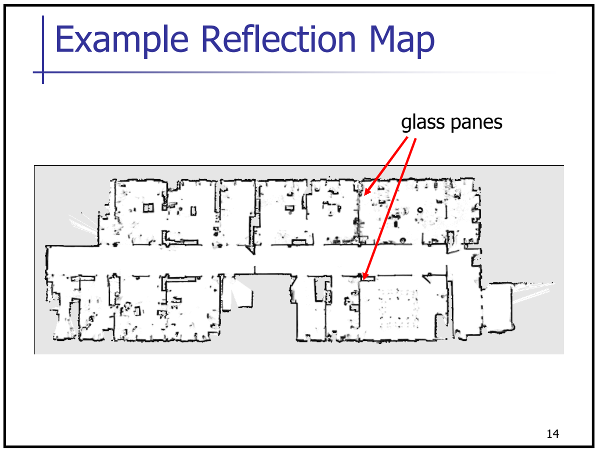


Difference between Occupancy Grid Maps and Counting

- The counting model determines how often a cell reflects a beam.
- The occupancy model represents whether or not a cell is occupied by an object.
- Although a cell might be occupied by an object, the reflection probability of this object might be very small.



Example Occupancy Map



Example Reflection Map

Example

- Out of 1000 beams only 60% are reflected from a cell and 40% intercept it without ending in it.
- Accordingly, the reflection probability will be 0.6.
- Suppose *p(occ | z)* = 0.55 when a beam ends in a cell and *p(occ | z)* = 0.45 when a cell is intercepted by a beam that does not end in it.
- Accordingly, after n measurements we will have

$$\left(\frac{0.55}{0.45}\right)^{n*0.6} * \left(\frac{0.45}{0.55}\right)^{n*0.4} = \left(\frac{11}{9}\right)^{n*0.6} * \left(\frac{11}{9}\right)^{-n*0.4} = \left(\frac{11}{9}\right)^{n*0.2}$$

- Whereas the reflection map yields a value of 0.6, the occupancy grid value converges to 1.

Mapping using Raw Odometry

Distribution over robot poses and maps

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1})$$

- Standard particle filter represents the distribution by a set of samples

$$\{< (x_{1:t}^{(i)}, m^{(i)}), w^{(i)} >\}$$

Rao-Blackwellization

poses map observations & movements

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(m \mid x_{1:t}, z_{1:t})$$

Factorization first introduced by Murphy in 1999

Rao-Blackwellization

poses map observations & movements

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(m \mid x_{1:t}, z_{1:t})$$

SLAM posterior

Robot path posterior

Mapping with known poses

Factorization first introduced by Murphy in 1999

Rao-Blackwellization

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(m \mid x_{1:t}, z_{1:t})$$

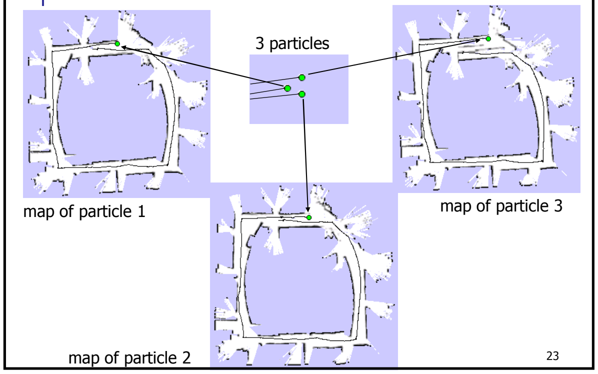
This is localization, use MCL

Use the pose estimate from the MCL part and apply mapping with known poses

Rao-Blackwellized Mapping

- Each particle represents a possible trajectory of the robot
- Each particle
  - maintains its own map and
  - updates it upon "mapping with known poses"
- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map

Particle Filter Example



Problem

- Each map is quite big in case of grid maps
- Since each particle maintains its own map
- Therefore, one needs to keep the number of particles small

- **Solution**:
  Compute better proposal distributions!
- **Idea**:
  Improve the pose estimate **before** applying the particle filter

Pose Correction Using Scan Matching

Maximize the likelihood of the i-th pose and map relative to the (i-1)-th pose and map

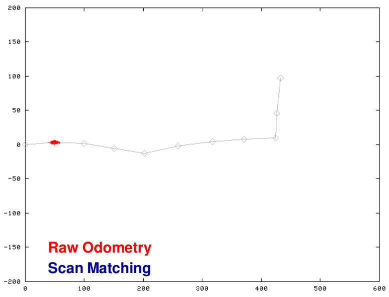
$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \left\{ p(z_t \mid x_t, \hat{m}_{t-1}) \cdot p(x_t \mid u_{t-1}, \hat{x}_{t-1}) \right\}$$

current measurement

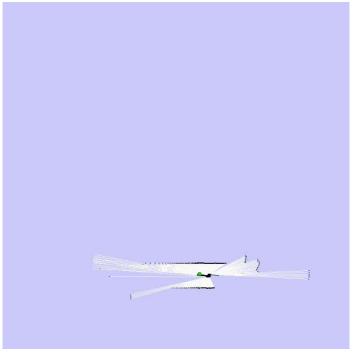
robot motion

map constructed so far

Motion Model for Scan Matching



FastSLAM with Scan-Matching

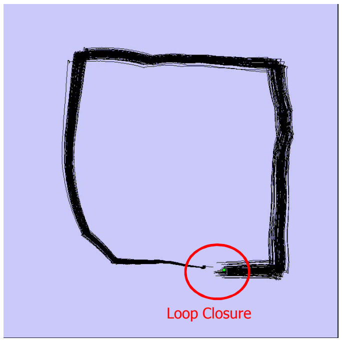


Map: Intel Research Lab Seattle

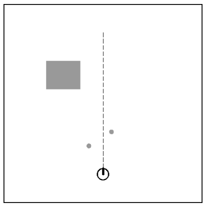
Map of the Intel Lab

- 15 particles
- four times faster than real-time P4, 2.8GHz
- 5cm resolution during scan matching
- 1cm resolution in final map

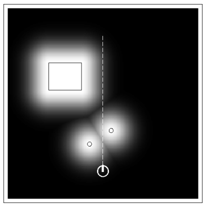
FastSLAM with Scan-Matching



Scan matching: likelihood field



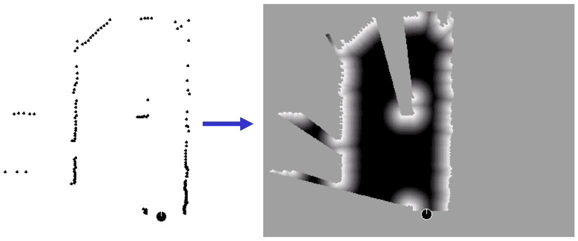
Map *m*



Likelihood field
=map convolved with a Gaussian

Scan Matching

- Extract likelihood field from scan and use it to match different scan.



FastSLAM recap

- Rao-Blackwellized representation:
  - Particle instantiates entire path of robot
  - Map associated with each path
- Scan matching: improves proposal distribution

- Original FastSLAM:
  - Map associated with each particle was a Gaussian distribution over landmark positions
- DP-SLAM: extension which has very efficient map management, enabling having a relatively large number of particles [Eliazar and Parr, 2002/2005]

SLAM thus far

- Landmark based vs. occupancy grid
- Probability distribution representation:
  - EKF vs. particle filter vs. Rao-Blackwellized particle filter

- EKF, SEIF, FastSLAM are all “online”

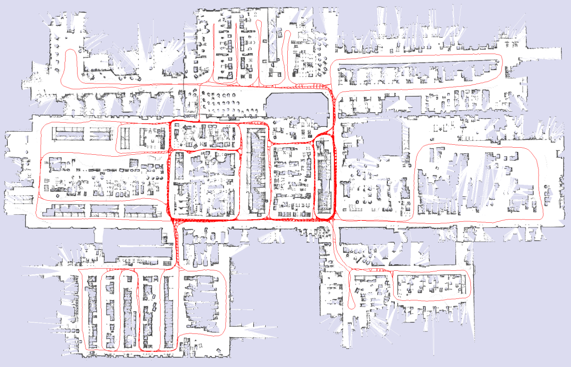
- Currently popular 4th alternative: GraphSLAM

Graph-based Formulation

- Use a **graph** to represent the problem
- **Every node** in the graph **corresponds to a pose** of the robot during mapping
- **Every edge** between two nodes **corresponds to the spatial constraints** between them
- **Goal**:
  Find a configuration of the nodes that **minimize the error** introduced by the constraints

$$\begin{aligned} J_{\text{GraphSLAM}} &= x_0^\top \Omega_0 x_0 + \sum_t (x_t - g(u_t, x_{t-1}))^\top R_t^{-1} (x_t - g(u_t, x_{t-1})) \\ &+ \sum_t \sum_i (z_t^i - h(x_t, m, c_t^i))^\top Q_t^{-1} (z_t^i - h(x_t, m, c_t^i)) \end{aligned}$$

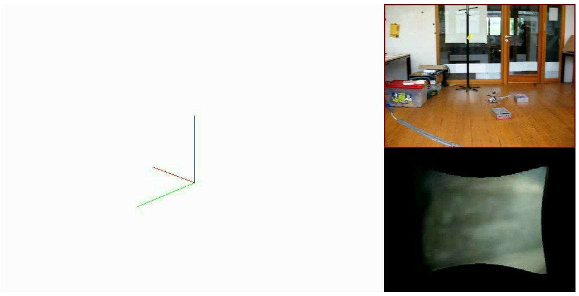
The KUKA Production Site



The KUKA Production Site

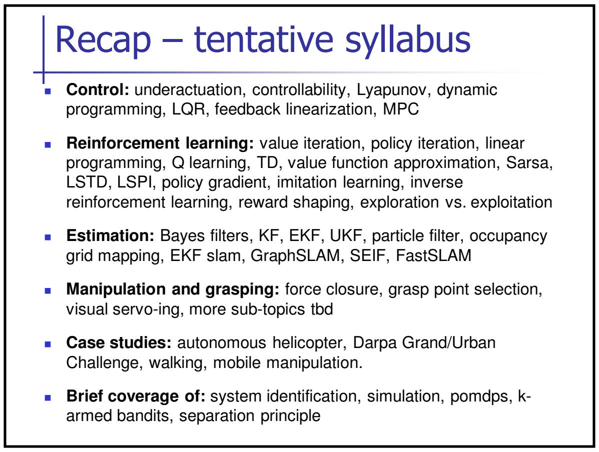
scans	59668
total acquisition time	4,699.71 seconds
traveled distance	2,587.71 meters
total rotations	262.07 radians
size	180 x 110 meters
processing time	< 30 minutes

GraphSLAM



Visual SLAM for Flying Vehicles
Bastian Steder, Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard

Autonomous Blimp



Recap – tentative syllabus

- **Control:** underactuation, controllability, Lyapunov, dynamic programming, LQR, feedback linearization, MPC
- **Reinforcement learning:** value iteration, policy iteration, linear programming, Q learning, TD, value function approximation, Sarsa, LSTD, LSPI, policy gradient, imitation learning, inverse reinforcement learning, reward shaping, exploration vs. exploitation
- **Estimation:** Bayes filters, KF, EKF, UKF, particle filter, occupancy grid mapping, EKF slam, GraphSLAM, SEIF, FastSLAM
- **Manipulation and grasping:** force closure, grasp point selection, visual servo-ing, more sub-topics tbd
- **Case studies:** autonomous helicopter, Darpa Grand/Urban Challenge, walking, mobile manipulation.
- **Brief coverage of:** system identification, simulation, pomdps, k-armed bandits, separation principle