SLAM: Robotic Simultaneous Location and Mapping

William Regli
Department of Computer Science
(and Departments of ECE and MEM)
Drexel University

Acknowledgments to Sebastian Thrun & others…

SLAM Lecture Outline

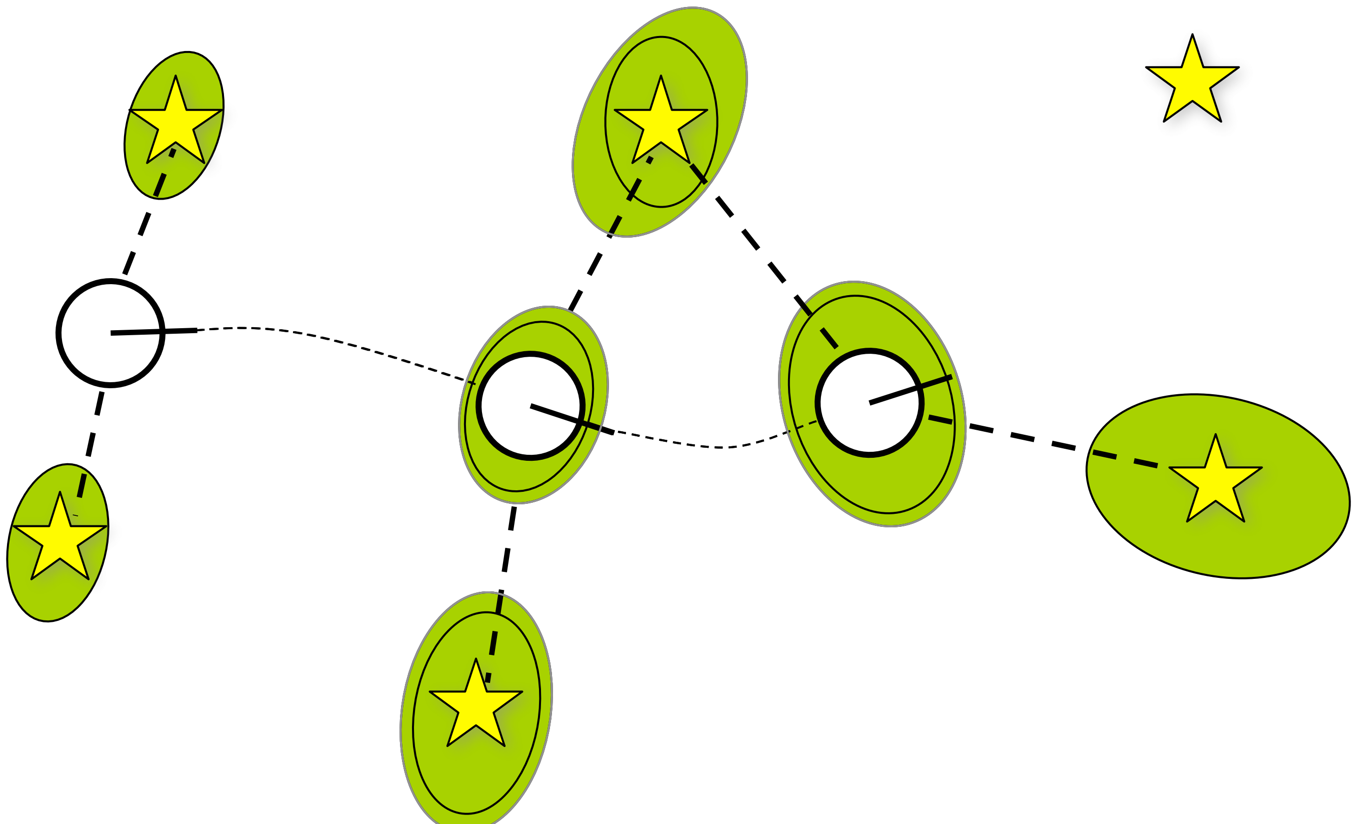
- **SLAM**
- Robot Sensing and Localization
- Robot Mapping
- Robot Motion Models

The SLAM Problem

- SLAM stands for *simultaneous localization and mapping*
- The task of building a map while estimating the pose of the robot relative to this map

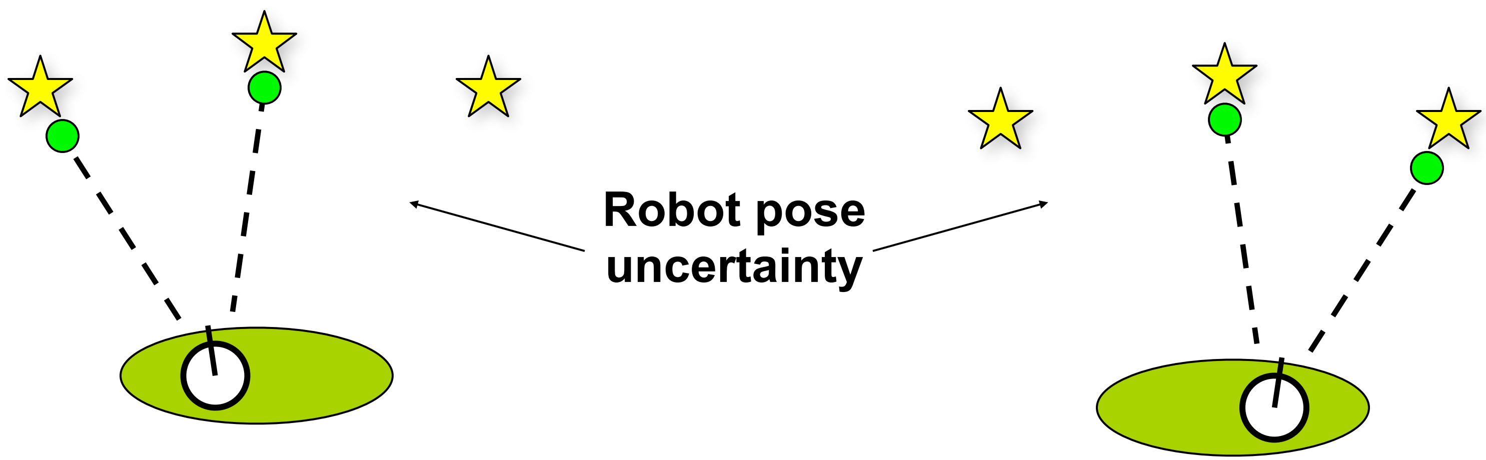
- Why is SLAM hard?
  Chicken and egg problem:
  a map is needed to localize the robot and
  a pose estimate is needed to build a map

Why is SLAM a hard problem?



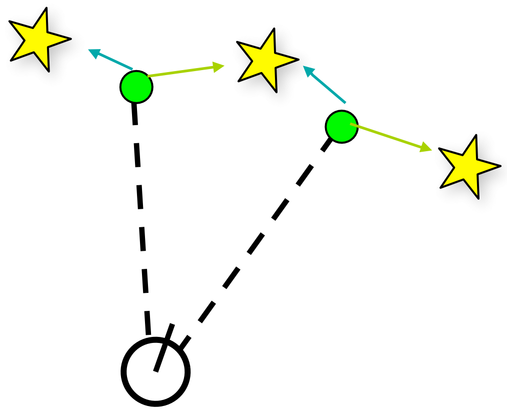
SLAM: robot path and map are both **unknown!**
Robot path error correlates errors in the map

Why is SLAM a hard problem?



- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

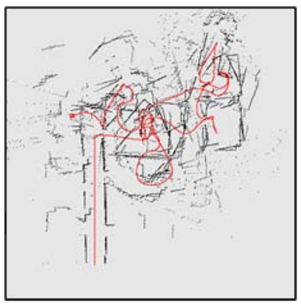
Data Association Problem

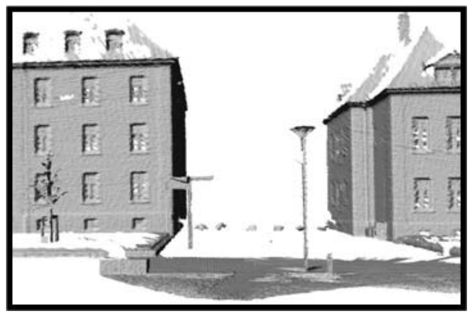


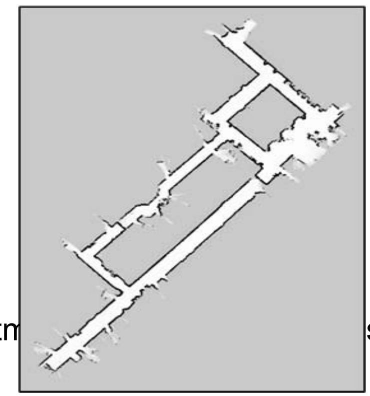
- A data association is an assignment of observations to landmarks
- In general there are more than [n choose m] (n observations, m landmarks) possible associations
- Also called “assignment problem”

Representations

- Grid maps or scans

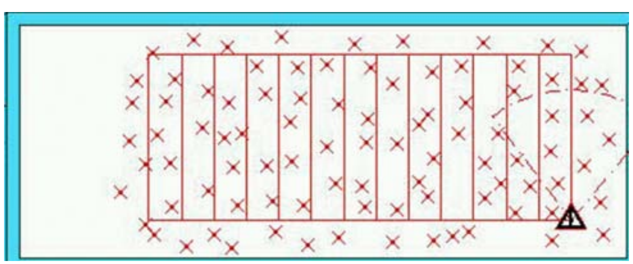


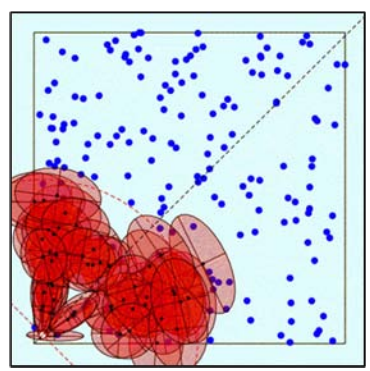


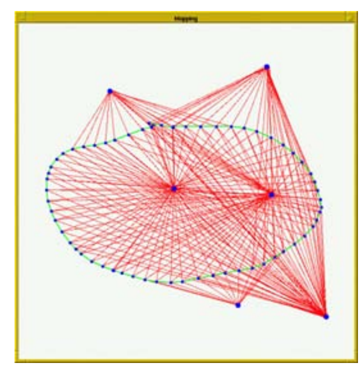


[Lu & 98: & Gutm s, 99; Haehnel, 01; …]

- Landmark-based

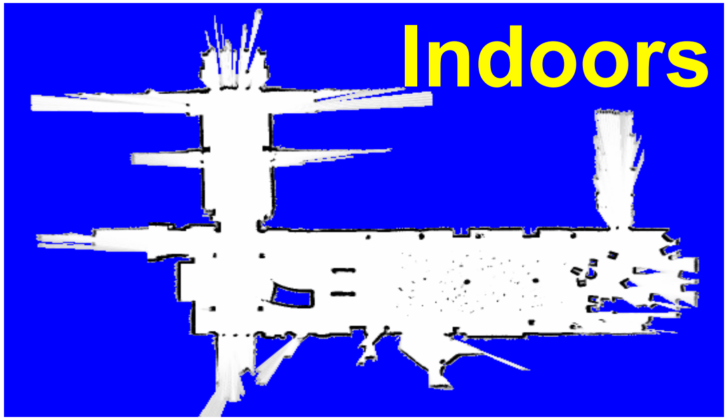


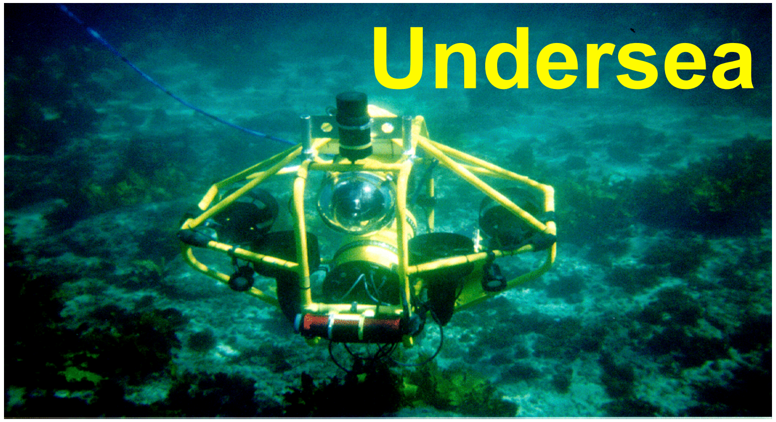


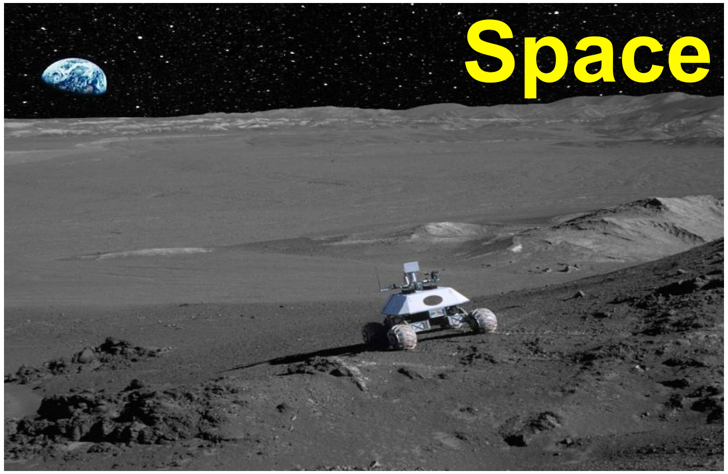


[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;…

SLAM Applications







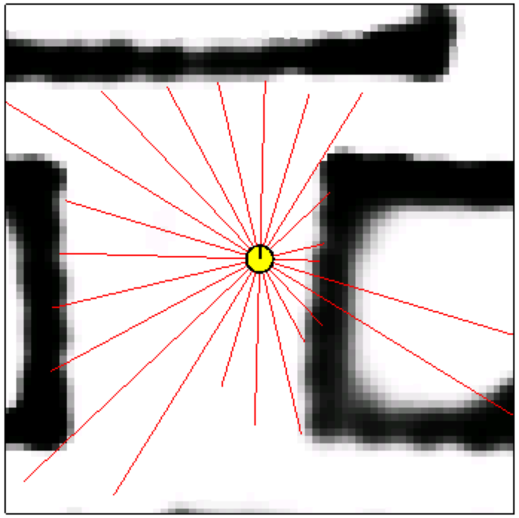
SLAM Lecture Outline

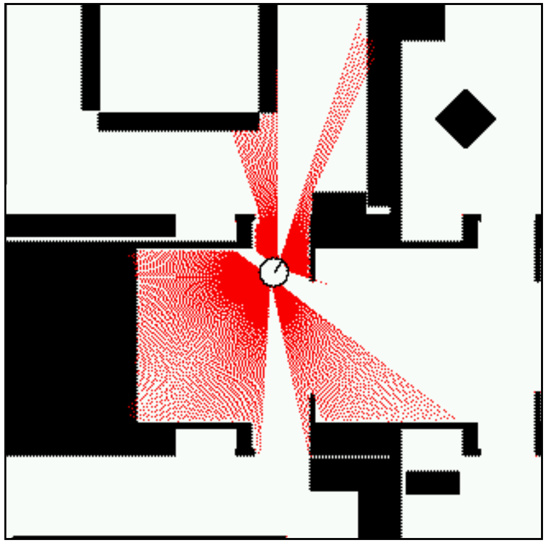
- SLAM
- **Robot Sensing and Localization**
- Robot Mapping
- Robot Motion Models

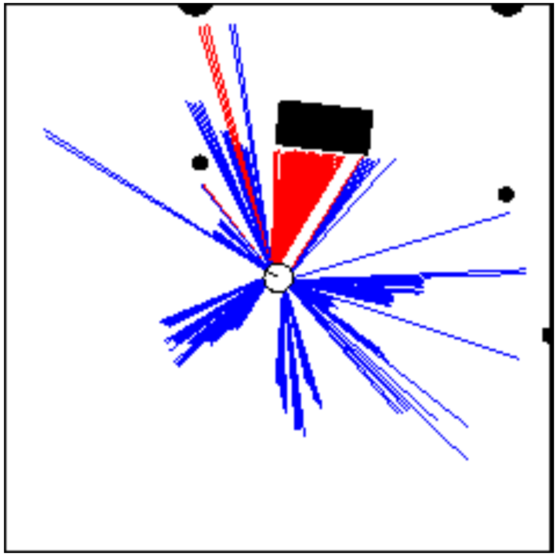
Sensors for Mobile Robots

- Contact sensors: Bumpers
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth’s magnetic field, gravity)
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, time of flight, phase)
  - Infrared (intensity)
- Visual sensors: Cameras
- Satellite-based sensors: GPS

Proximity Sensors



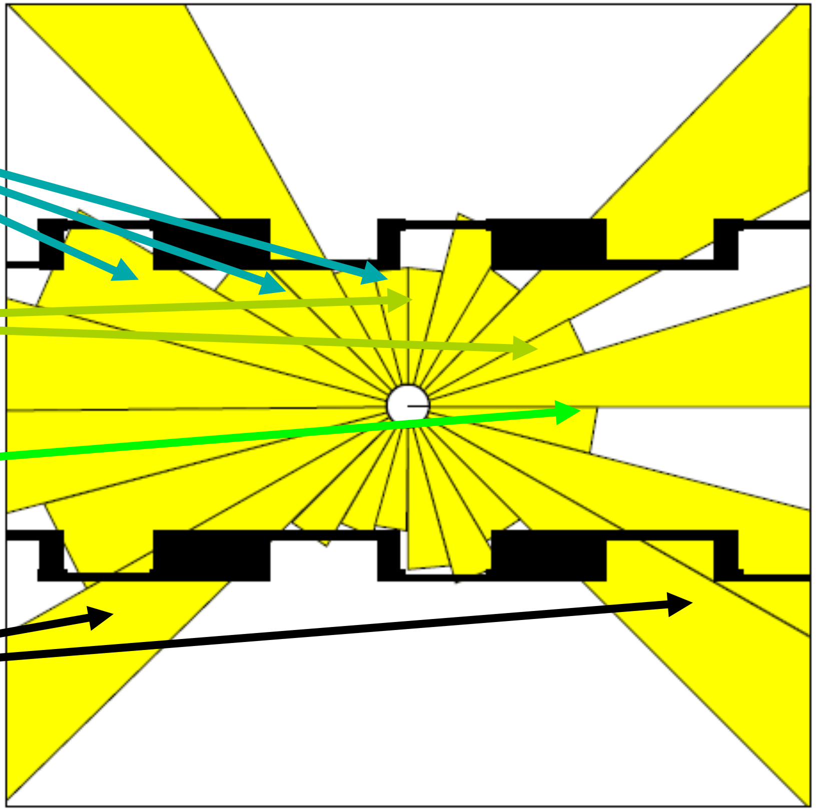




- The central task is to determine *P(z|x)*, i.e., the probability of a measurement *z* given that the robot is at position *x*.

Typical Range Measurement Errors

- Beams reflected by obstacles
- Beams reflected by persons / caused by crosstalk
- Random measurements
- Maximum range measurements



Proximity Measurement

- Measurement can be caused by …
  - a known obstacle.
  - cross-talk.
  - an unexpected obstacle (people, furniture, …).
  - missing all obstacles (total reflection, glass, …).
- Noise is due to uncertainty …
  - in measuring distance to known obstacle.
  - in position of known obstacles.
  - in position of additional obstacles.
  - whether obstacle is missed.

Additional Models of Proximity Sensors

- Map matching (sonar,laser): generate small, local maps from sensor data and match local maps against global model.

- Scan matching (laser): map is represented by scan endpoints, match scan into this map.

- Features (sonar, laser, vision): Extract features such as doors, hallways from sensor data.

Important points about Sensor Models in Localization

- Explicitly modeling uncertainty in sensing is key to robustness.
- In many cases, good models can be found by the following approach:
  - Determine parametric model of noise free measurement.
  - Analyze sources of noise.
  - Add adequate noise to parameters (eventually mix in densities for noise).
  - Learn (and verify) parameters by fitting model to data.
  - Likelihood of measurement is given by “probabilistically comparing” the actual with the expected measurement.
- This holds for motion models as well.
- It is extremely important to be aware of the underlying assumptions!

Localization

"Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." [Cox '91]

Given

- Map of the environment.
- Sequence of sensor measurements.

Wanted

- Estimate of the robot's position.

Problem classes

- Position tracking
- Global localization
- Kidnapped robot problem (recovery)

Localization using Kinematics

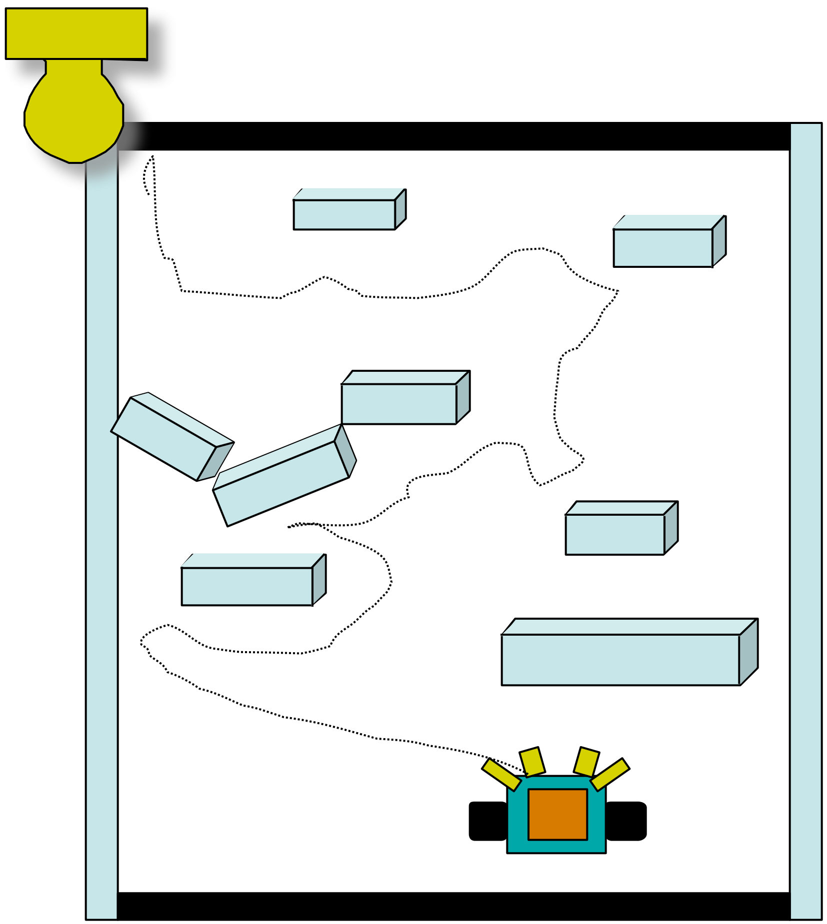
- Issue: We can’t (necessarily) tell direction from encoders alone
- Solution: Keep track of forward/backward motor command sent to each wheel
- Localization program: Build new arrays into behavior/priority-based controller and use to continually update location
- Doesn’t solve noise problems, though

Localization Using Landmarks

- Active beacons (*e.g*., radio, GPS)
- Passive (*e.g*., visual, retro-reflective)
- Standard approach is triangulation

- Sensor provides
  - distance, or
  - bearing, or
  - distance and bearing.

Correcting Localization with Landmarks

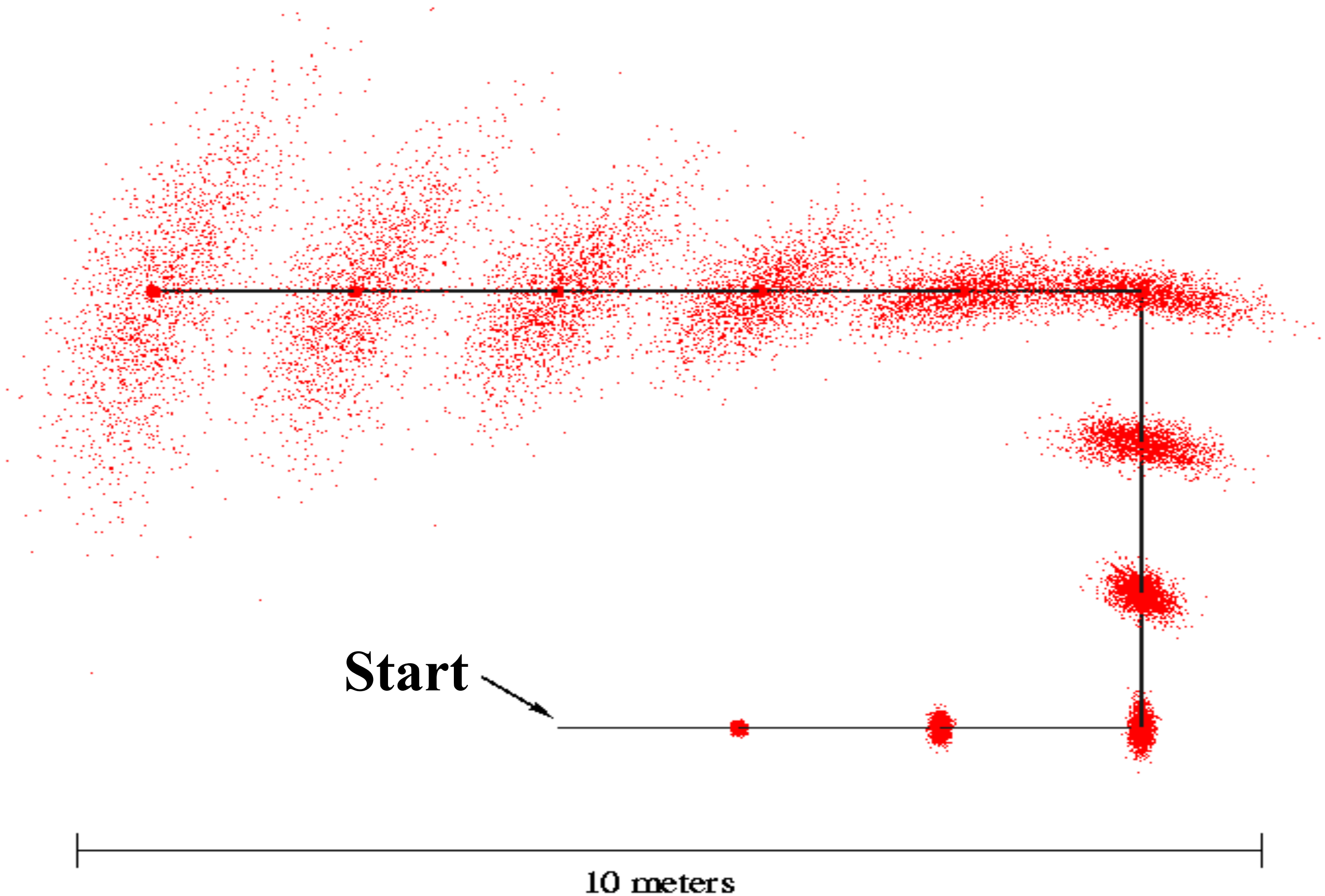


- Keep track of (x,y,theta) between landmarks
- Correct for absolute y (known) when ground sensor triggers landmark
- Issues:
  - Uncertainty in x and theta not corrected using this method
  - Possible to confuse landmarks

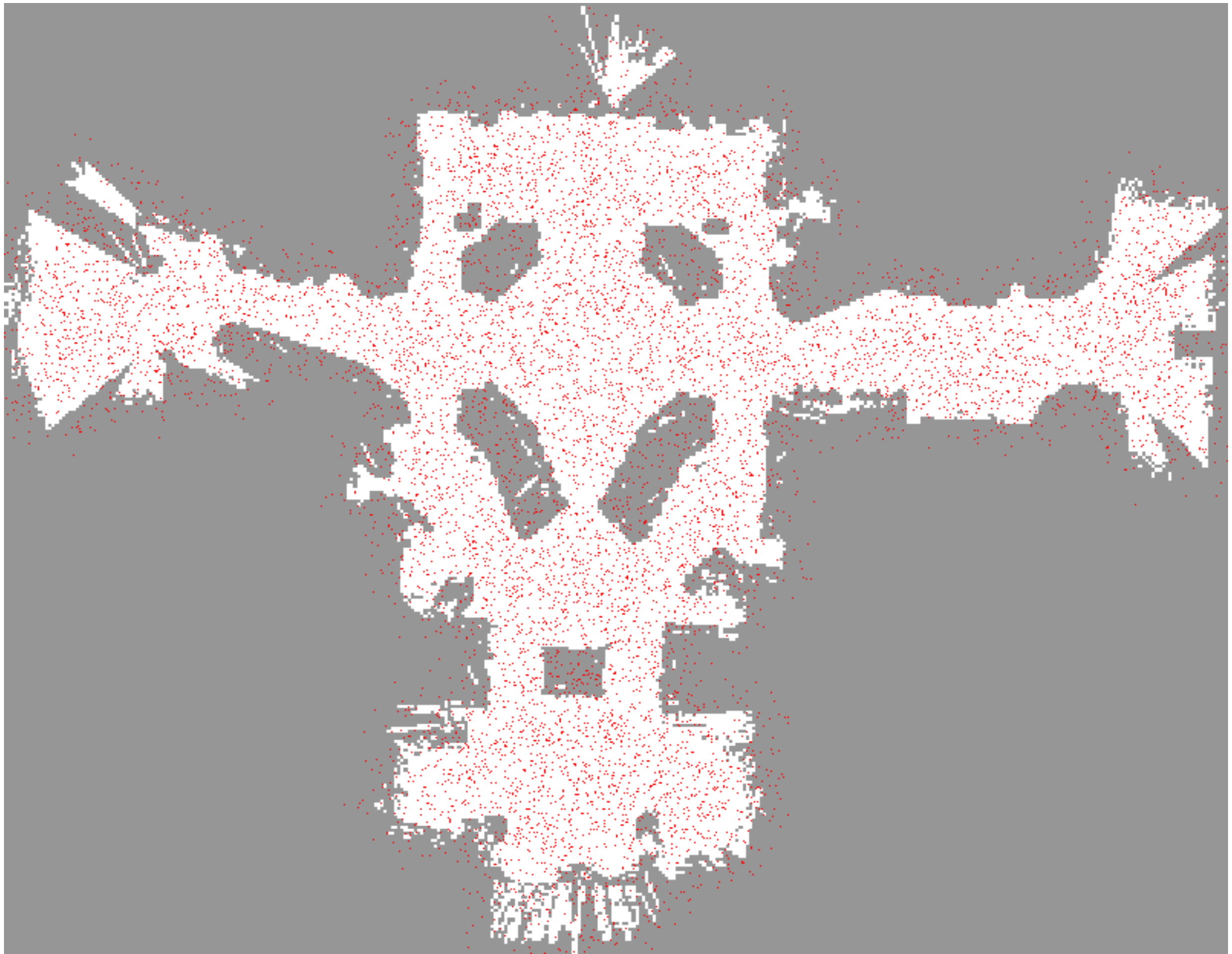
Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes
- Sampling Importance Resampling (SIR) principle
  - Draw the new generation of particles
  - Assign an importance weight to each particle
  - Resampling
- Typical application scenarios are tracking, localization, …

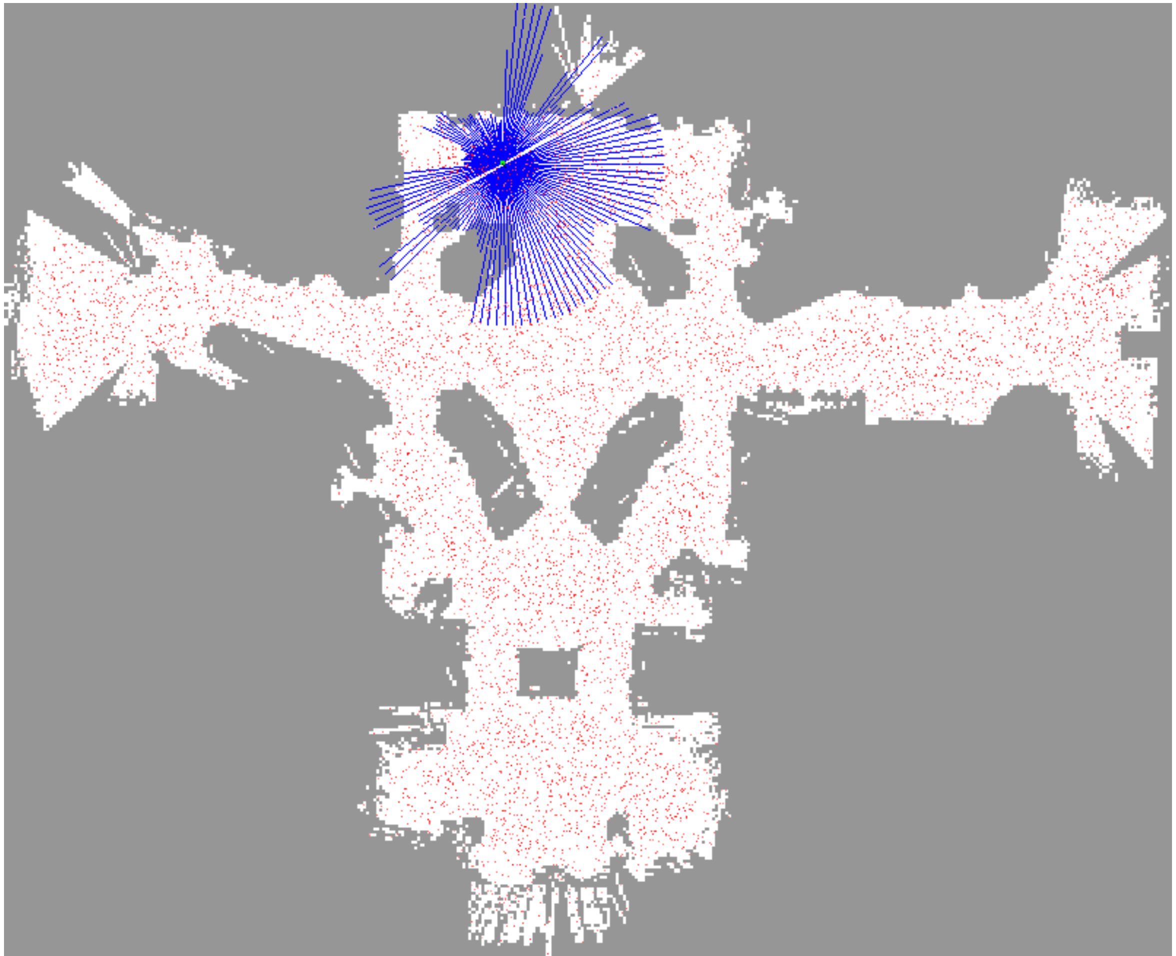
Motion Model Reminder



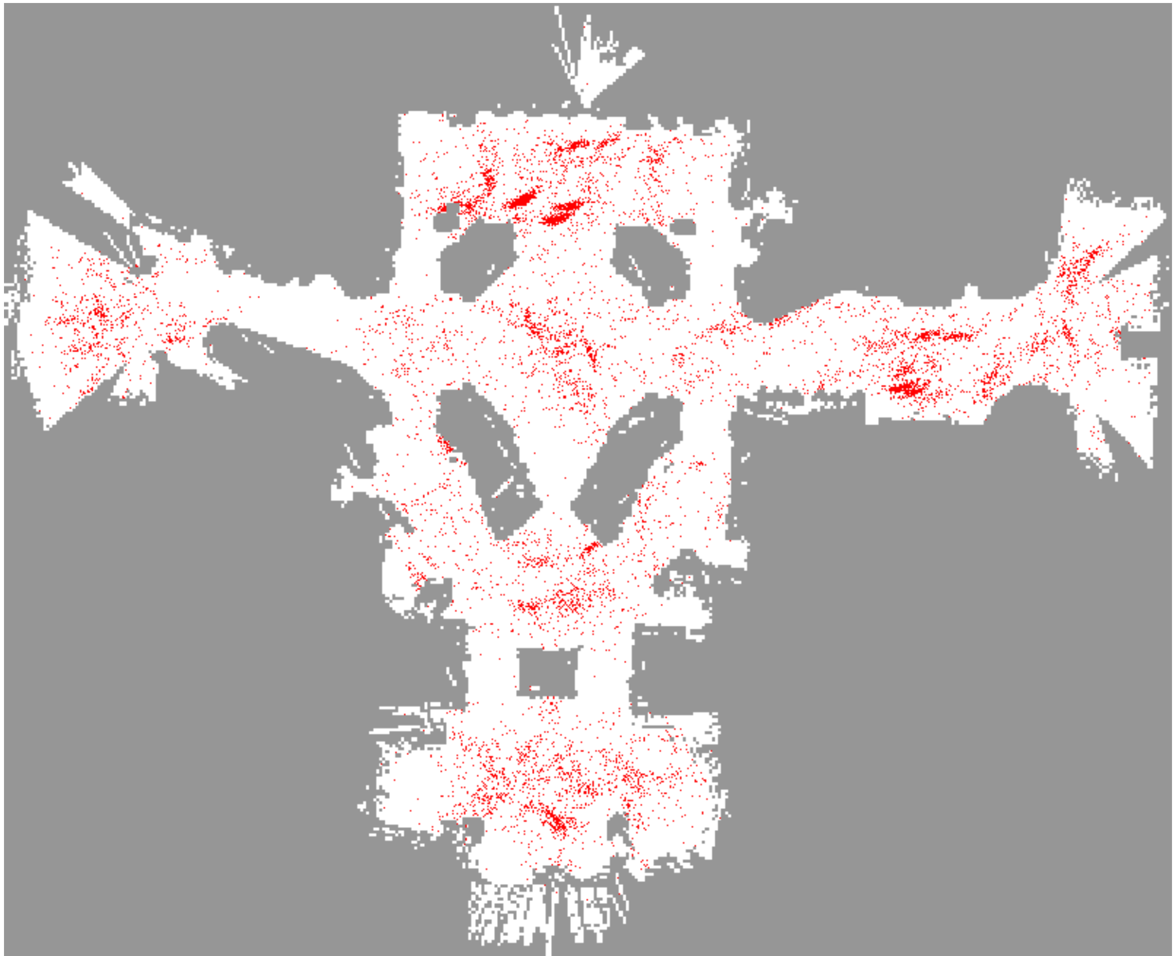
Importance Sampling with Resampling



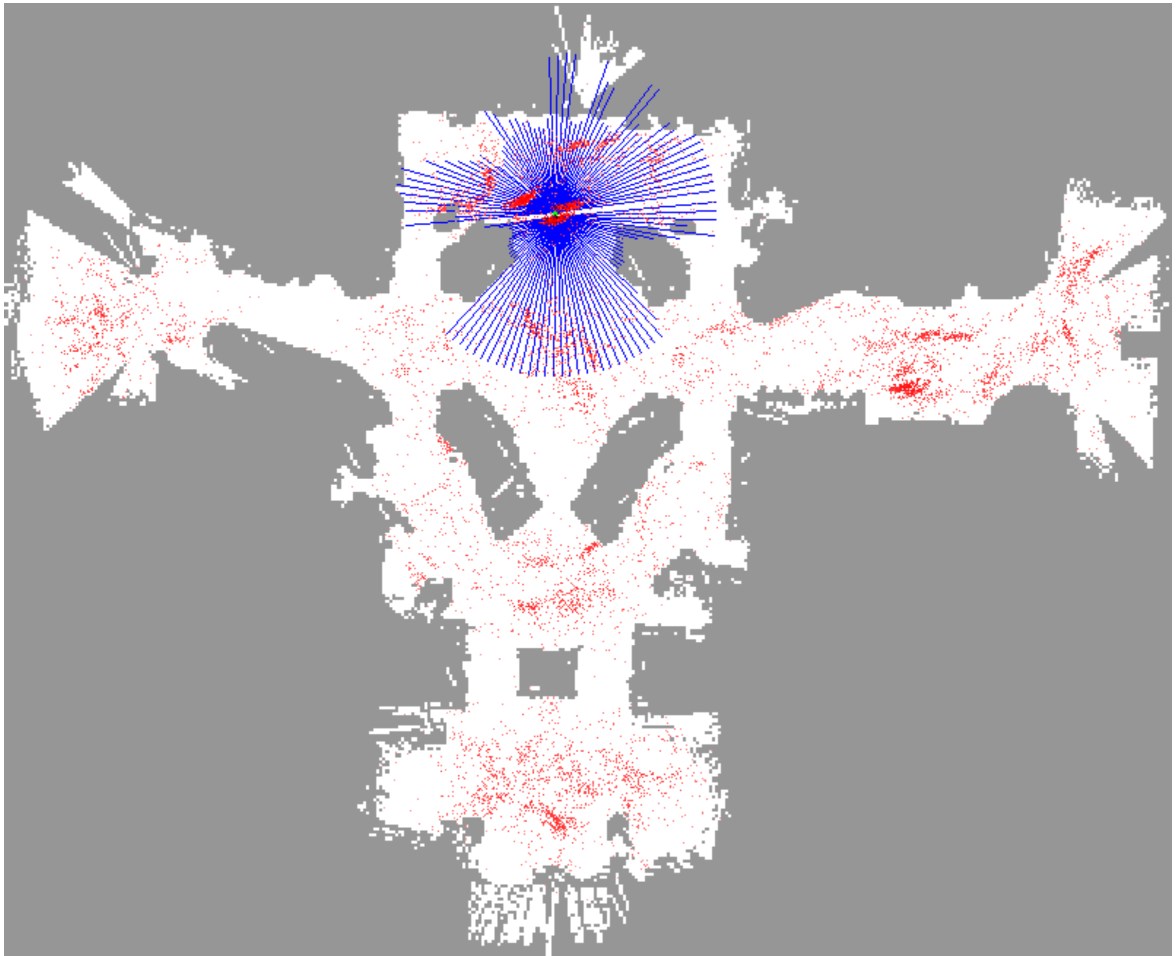
Importance Sampling with Resampling



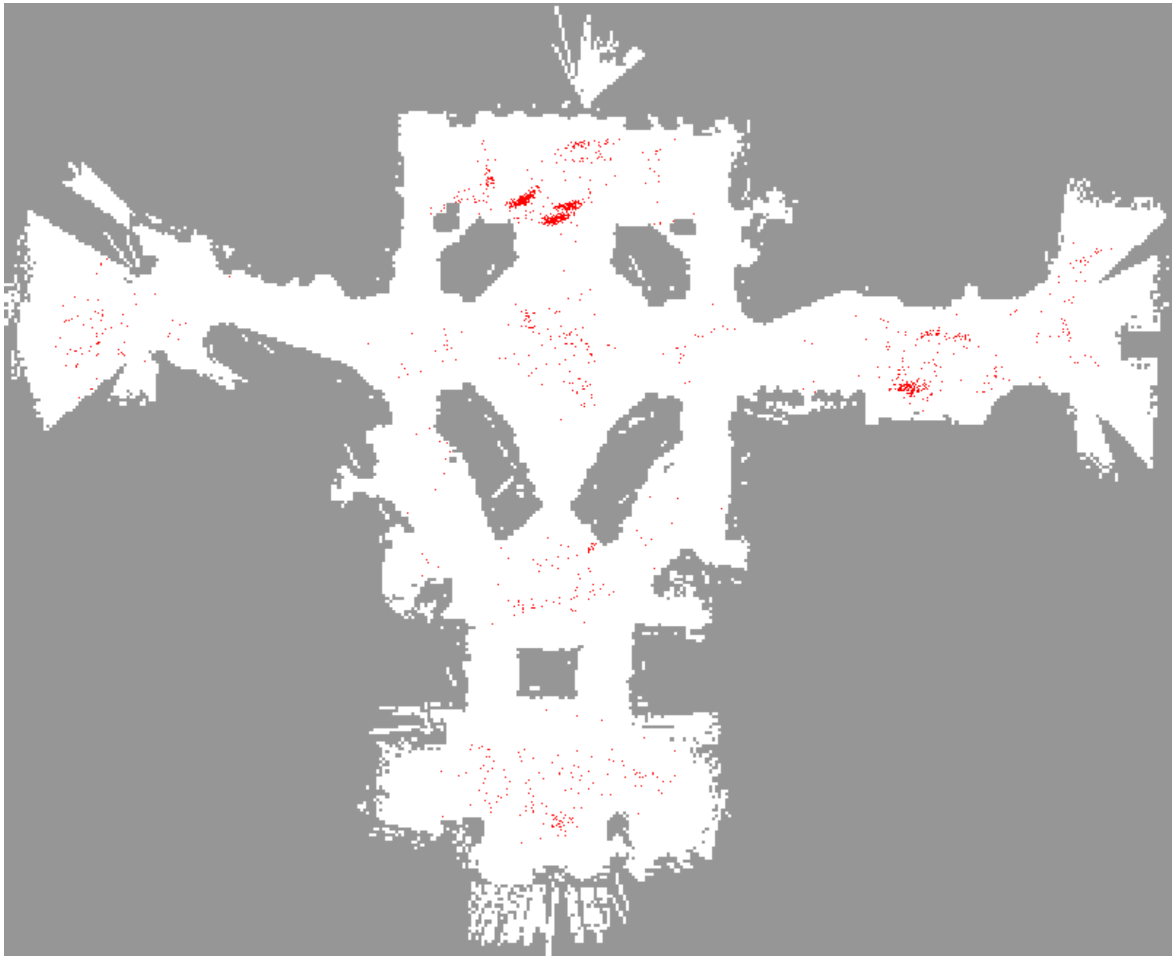
Importance Sampling with Resampling



Importance Sampling with Resampling

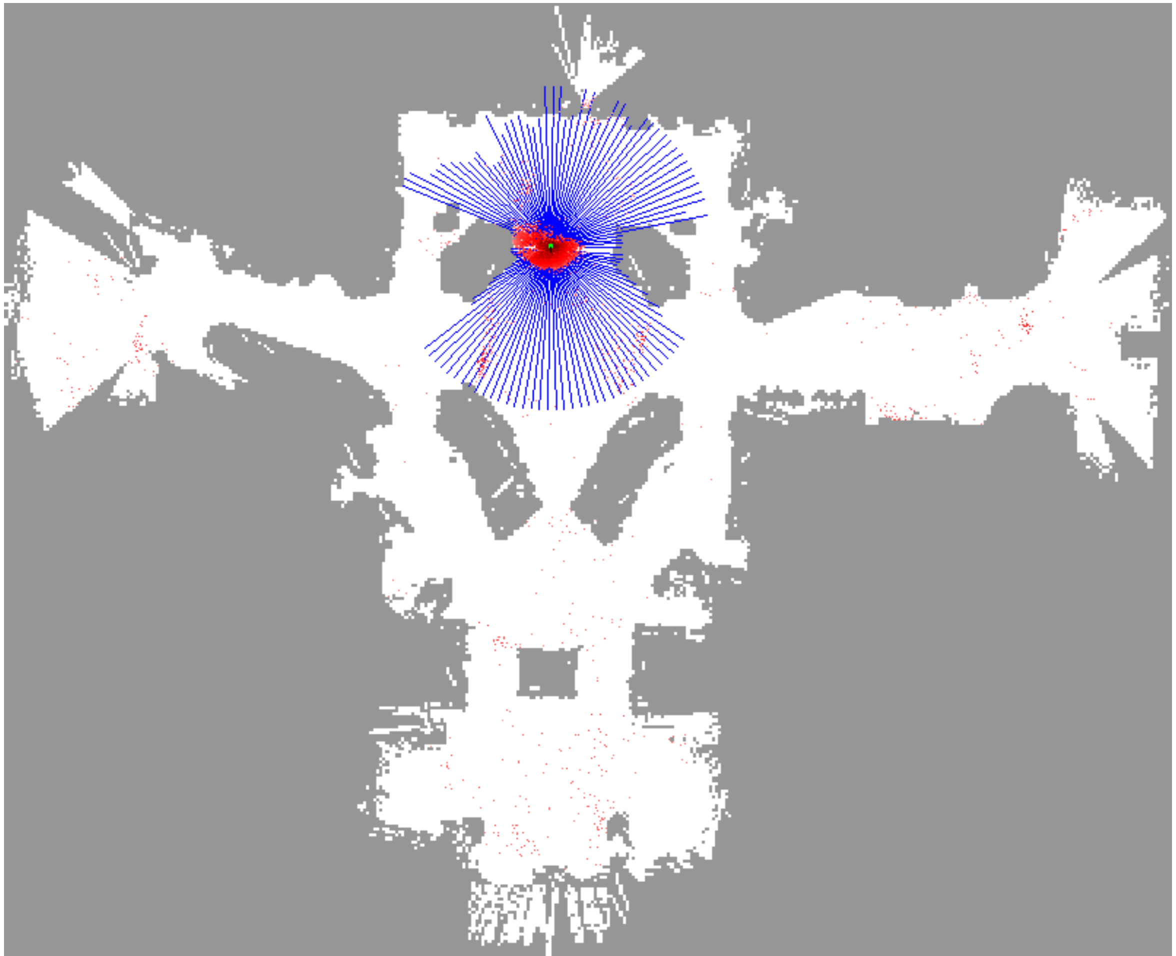


Importance Sampling with Resampling

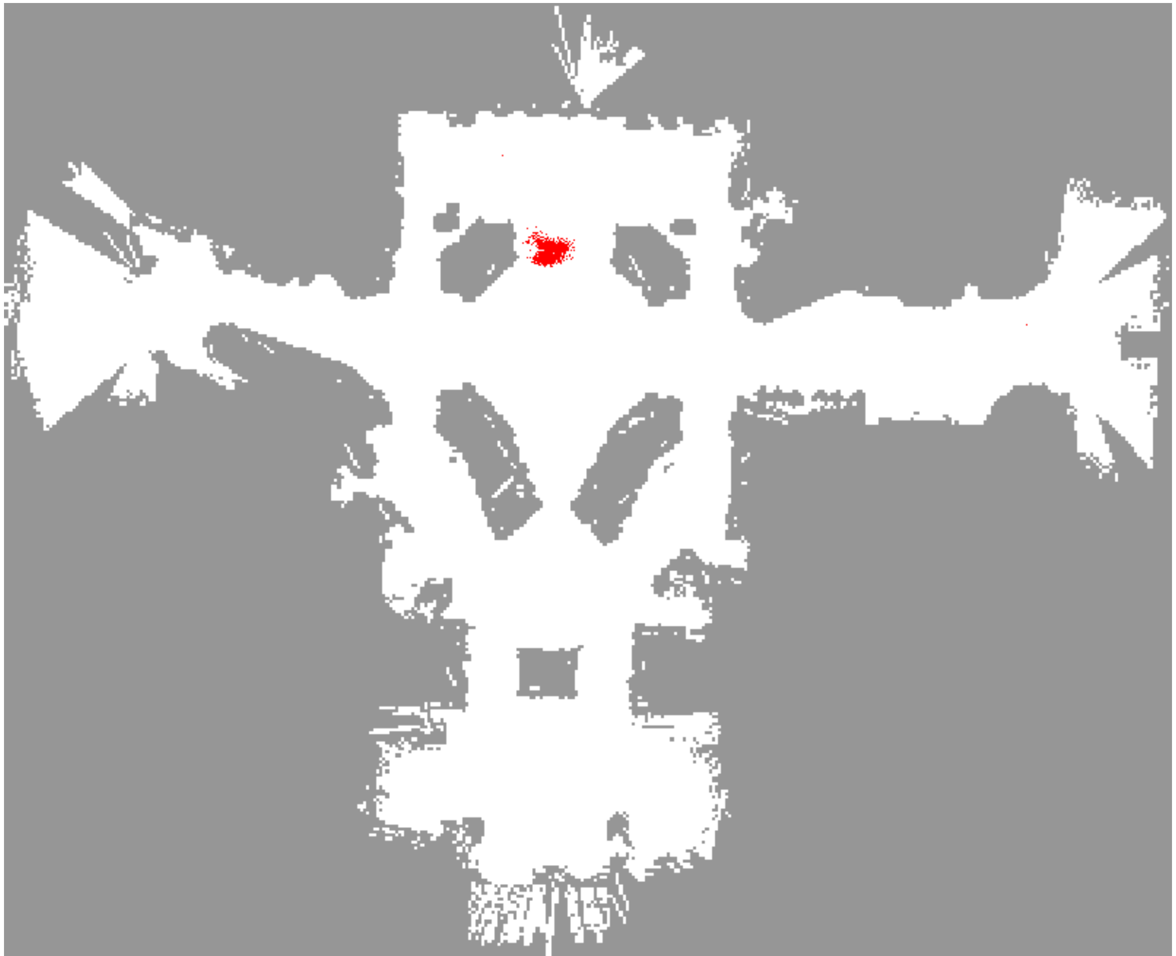


Importance Sampling with Resampling

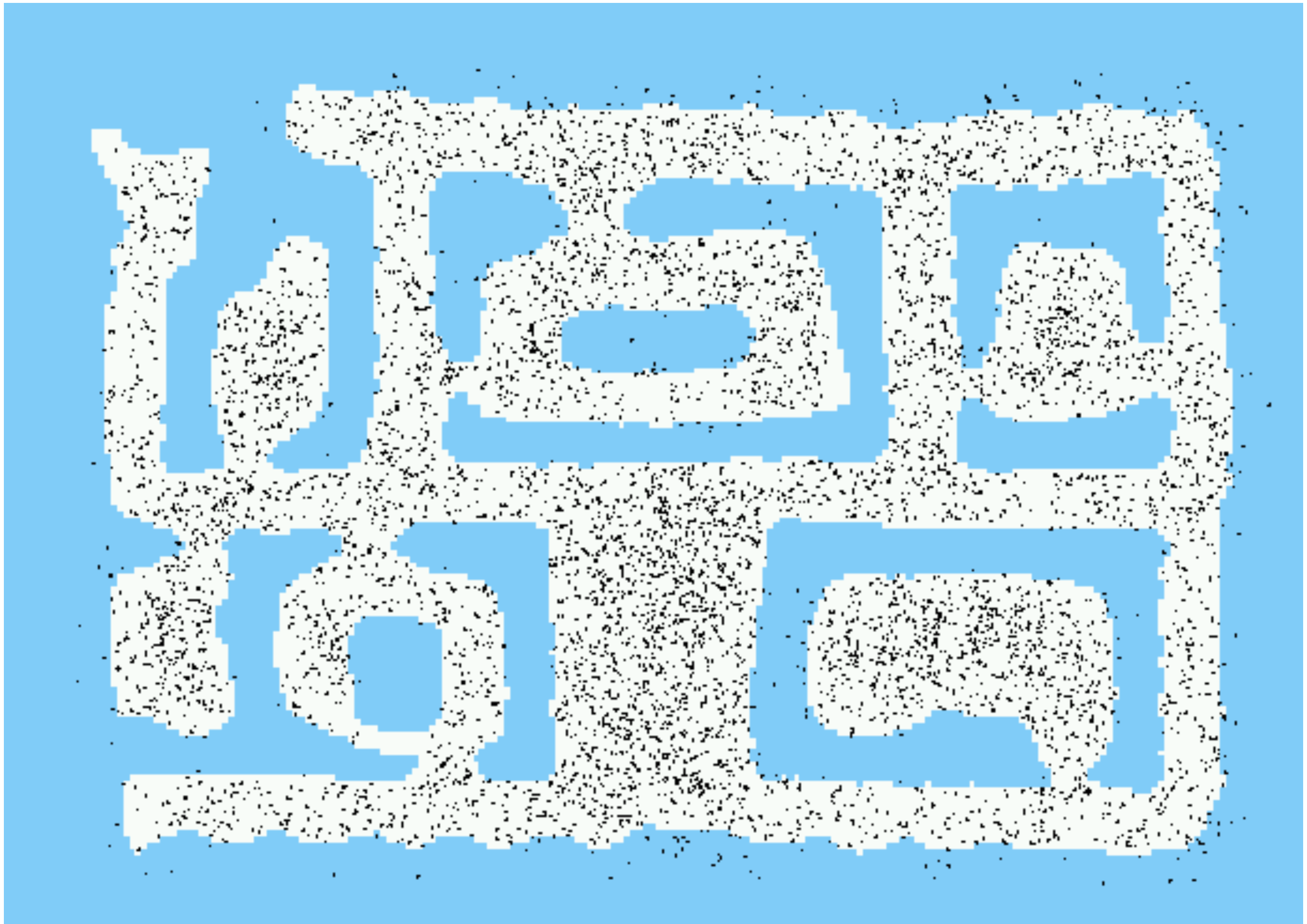
Importance Sampling with Resampling



Importance Sampling with Resampling



Monte Carlo Localization: Initial Distribution



Monte Carlo Localization: After Incorporating 65 Ultrasound Scans

SLAM Lecture Outline

- SLAM
- Robot Sensing and Localization
- **Robot Mapping**
- Robot Motion Models

Why Mapping?

- Learning maps is one of the fundamental problems in mobile robotics
- Maps allow robots to efficiently carry out their tasks, allow localization …
- Successful robot systems rely on maps for localization, path planning, activity planning etc.

The General Problem of Mapping

What does the environment look like?

Formally, mapping involves, given the sensor data, to calculate the most likely map

Mapping as a Chicken and Egg Problem

- So far we learned how to estimate the pose of the vehicle given the data and the map (localization).
- Mapping, however, involves to simultaneously estimate the pose of the vehicle and the map.
- The general problem is therefore denoted as the simultaneous localization and mapping problem (SLAM).
- Throughout this section we will describe how to calculate a map given we know the pose of the vehicle

Problems in Mapping

- Sensor interpretation
  - How do we extract relevant information from raw sensor data?
  - How do we represent and integrate this information over time?
- Robot locations have to be estimated
  - How can we identify that we are at a previously visited place?
  - This problem is the so-called data association problem.

Occupancy Grid Maps

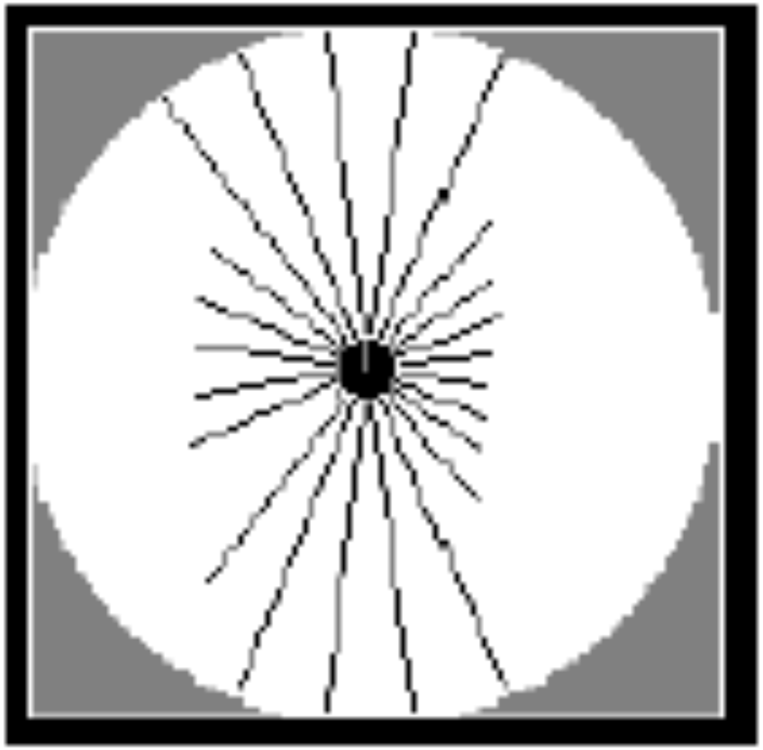
- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid.
- Estimate the probability that a location is occupied by an obstacle.
- Key assumptions
  - Occupancy of individual cells (m[xy]) is independent

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

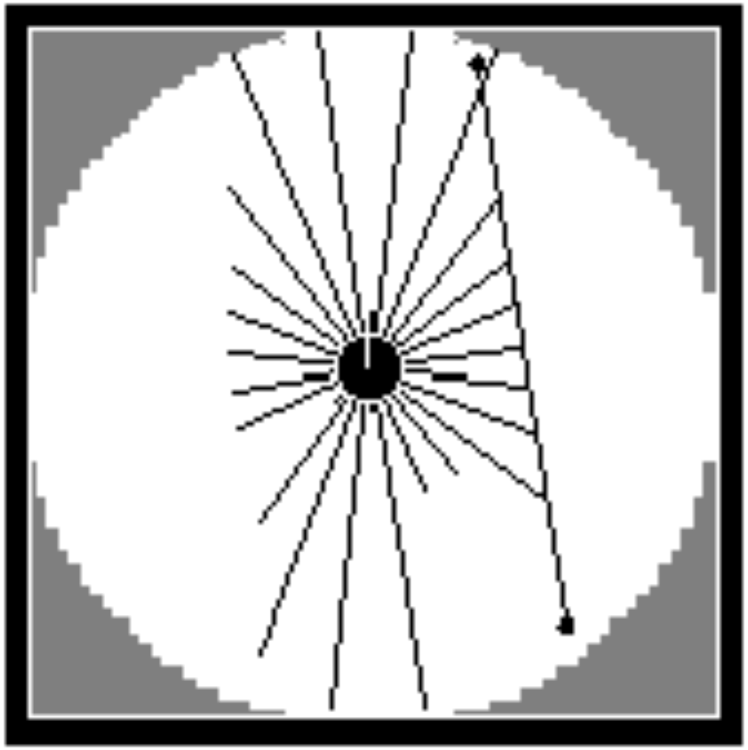
  - Robot positions are known!

Example Sonar Sweep

- Distance measurements from circular sonar scan
- What is robot seeing?

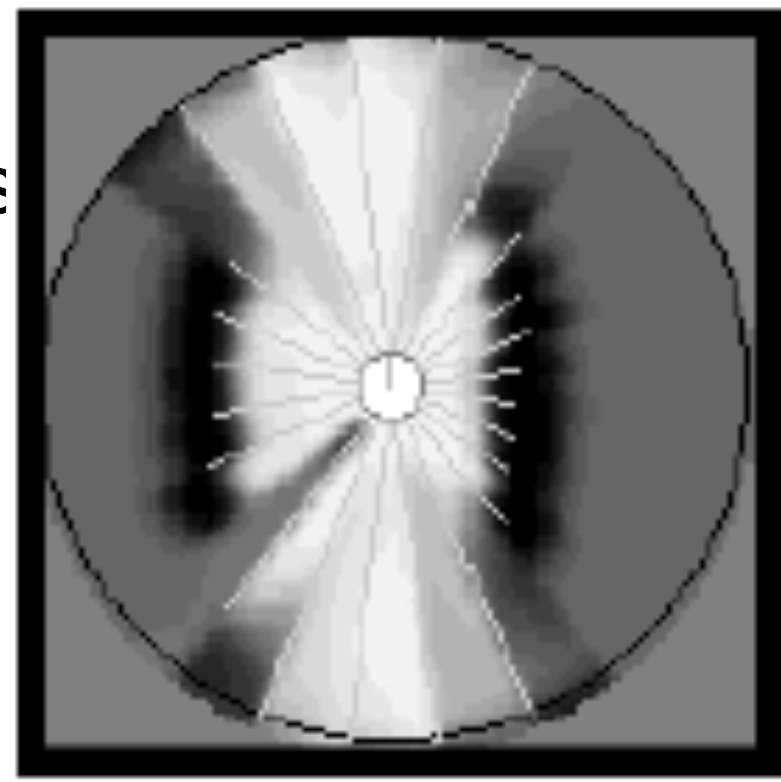


Detecting a Wall



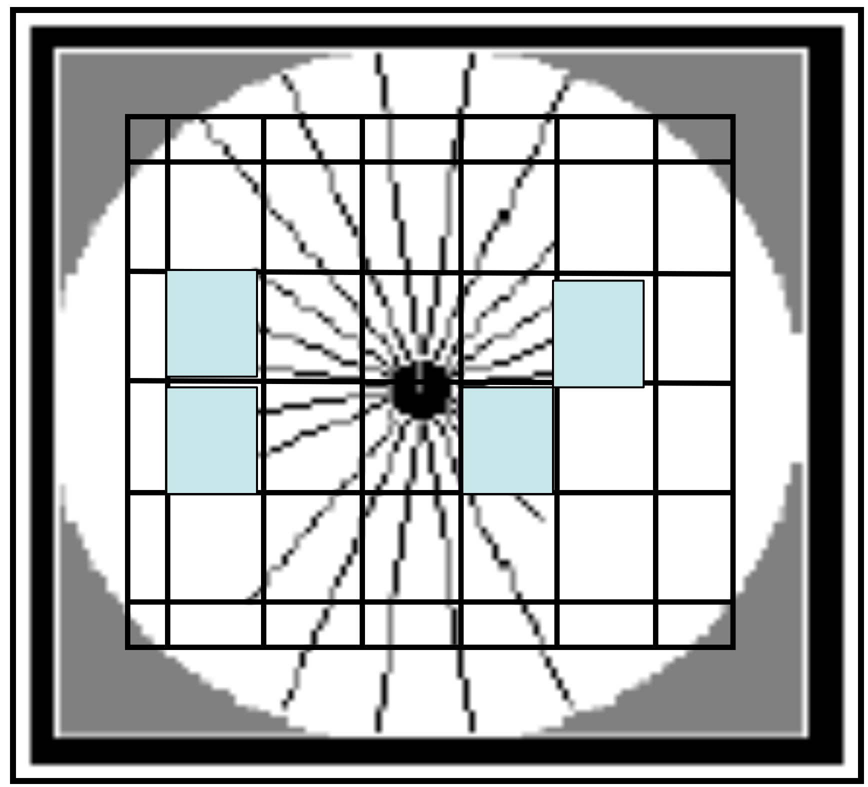
Partitioning Space into Regions

Process sweeps to partition space into free space (white), and walls and obstacles (black and grey)



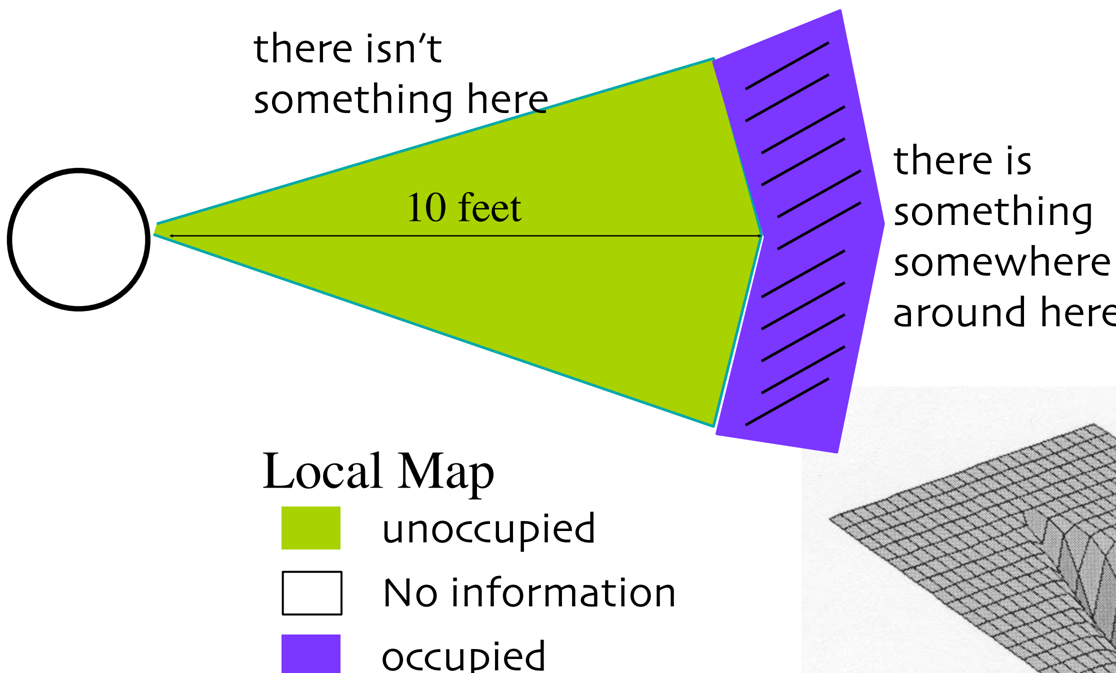
Grid-based Algorithm

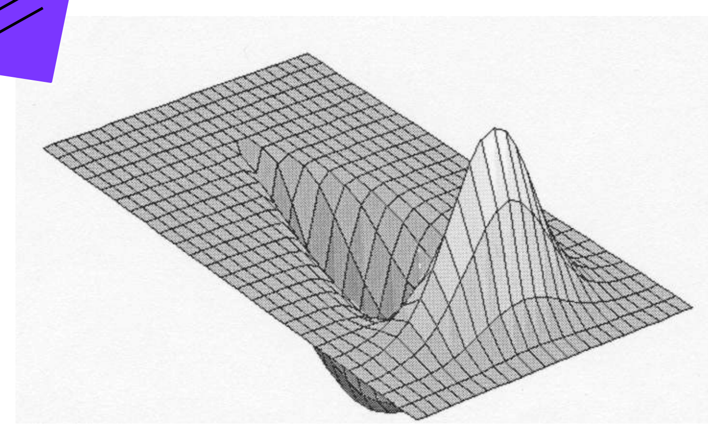
- Superimpose "grid" on robot field of view
- Indicate some measure of "obstacleness" in each grid cell based on sonar readings



So how do we use sonar to create maps?

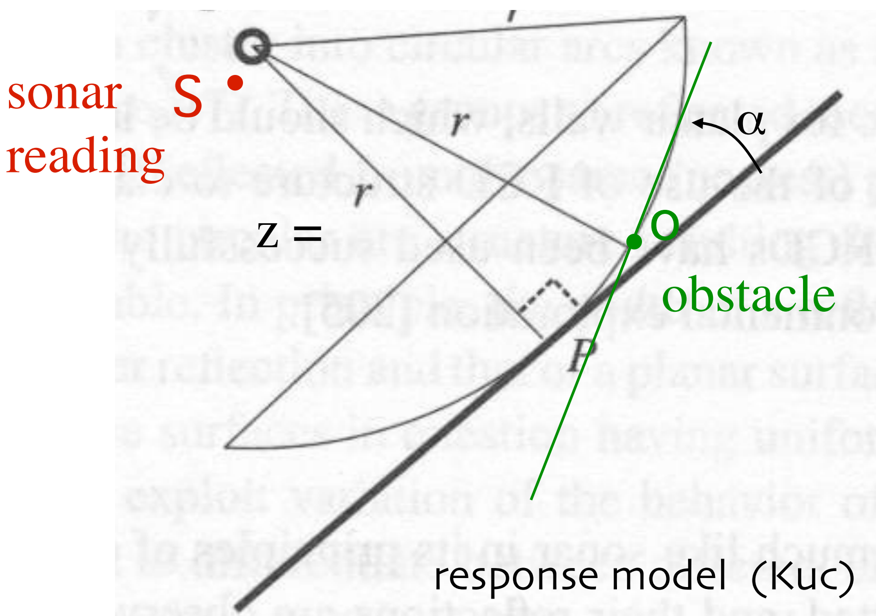
What should we conclude if this sonar reads 10 feet?





(Courtesy of Dodds)

Sonar Modeling



$$h_R(t, z, a, \alpha) = \frac{2c\cos\alpha}{\pi a \sin\alpha}\sqrt{1 - \frac{c^2(t - 2z/c)^2}{a^2 \sin^2\alpha}}$$

- Models the response, h_R, with

  c = speed of sound

  a = diameter of sonar element

  t = time

  z = orthogonal distance

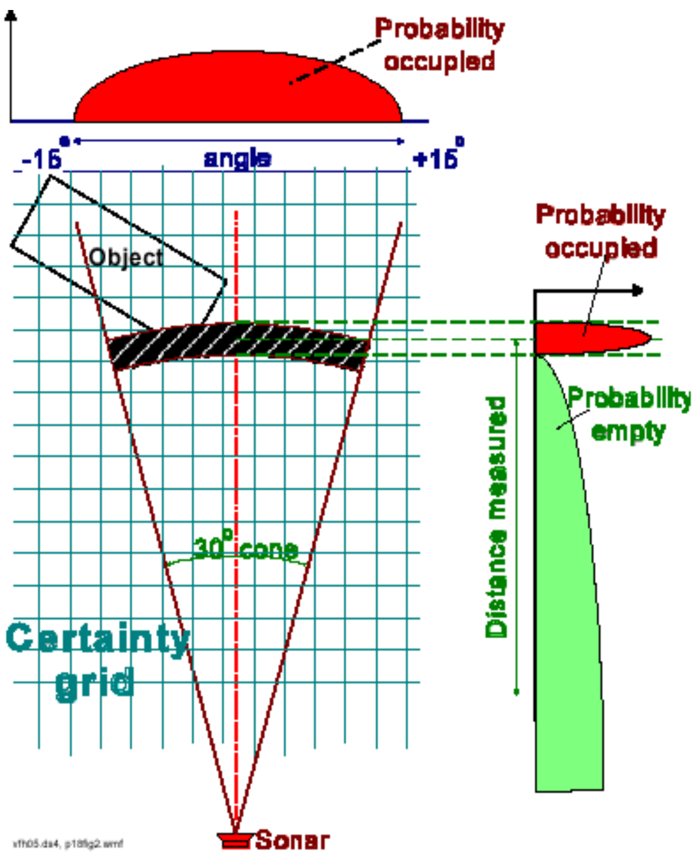
  α = angle of environment surface
- Then, add noise to the model to obtain a probability:

p(S | o)

chance that the sonar reading is S, given an obstacle at location o

(Courtesy of Dodds)

Typical Sonar Probability Model

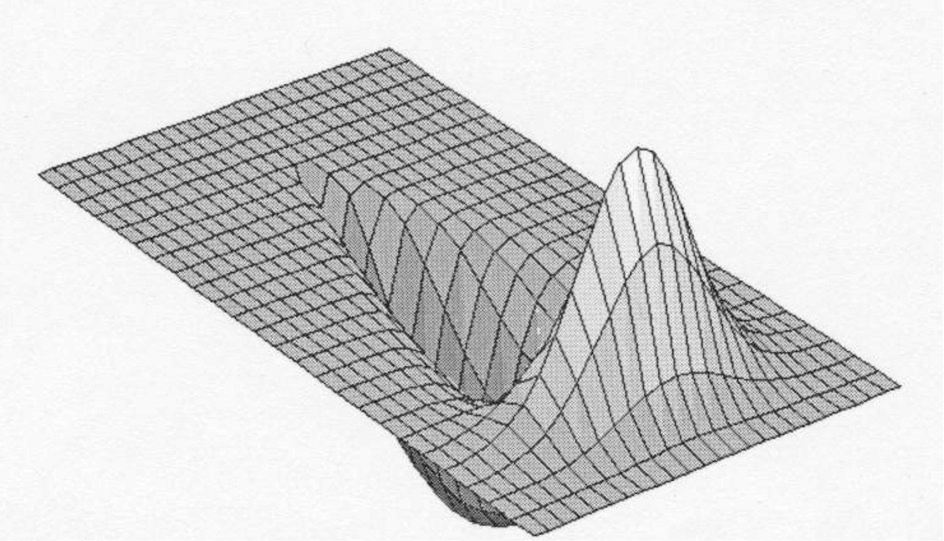


(From Borenstein et. Al.)

Building a Map

- The key to making accurate maps is combining lots of data.
- But combining these numbers means we have to know what they are !
- What should our map contain ?
  - small cells
  - each represents a bit of the robot's environment
  - larger values => obstacle
  - smaller values => free

  - Courtesy of Dodds



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>

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 (windows etc.).

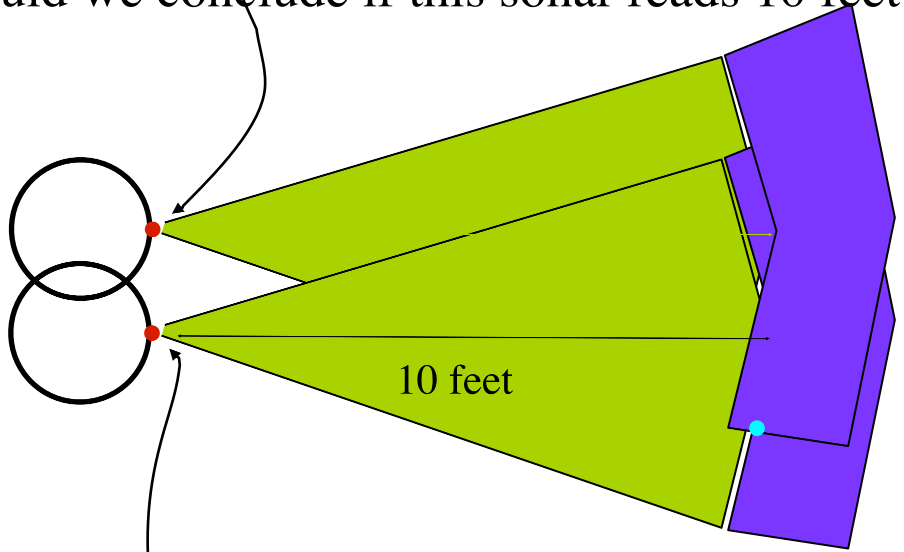
Example Occupancy Map

Properties of Mapping Methods

- Occupancy grid maps are a popular approach to represent the environment of a mobile robot given known poses.
- In this approach each cell is considered independently from all others.
- It stores the posterior probability that the corresponding area in the environment is occupied.
- Occupancy grid maps can be learned efficiently using a probabilistic approach.
- Reflection maps are an alternative representation.
- They store in each cell the probability that a beam is reflected by this cell.

Using sonar to create maps

What should we conclude if this sonar reads 10 feet...



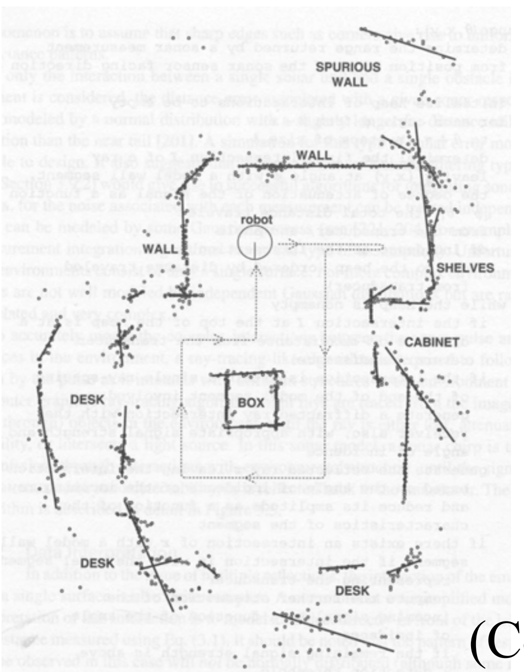
and how do we add the information that the next sonar reading (as the robot moves) reads 10 feet, too?

(Courtesy of Dodds)

What is it a map of?

Several answers to this question have been tried:

pre '83 It's a map of occupied cells. o_{xy} cell (x,y) is occupied $\overline{o}_{xy}$ cell (x,y) is unoccupied

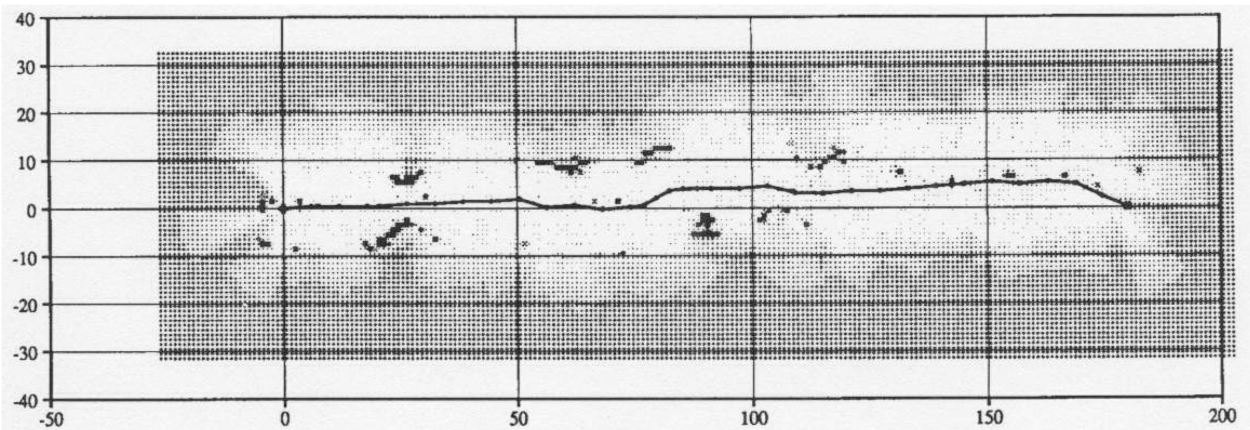


Each cell is either occupied or unoccupied -- this was the approach taken by the Stanford Cart.

(Courtesy of Dodds)

What information should this map contain, given that it is created with sonar ?

An example map



units: feet

Evidence grid of a tree-lined outdoor path

lighter areas: *lower* odds of obstacles being present

darker areas: *higher* odds of obstacles being present

(Courtesy of Dodds)

how to combine them?

Conditional probability

Some intuition...

The probability of event o, given event S .

$p(o \mid S) =$ The probability that a certain cell o is occupied, given that the robot sees the sensor reading S .

The probability of event S, given event o .

$p(S \mid o) =$ The probability that the robot sees the sensor reading S, given that a certain cell o is occupied.

- What is really meant by conditional probability ?
- How are these two probabilities related?

(Courtesy of Dodds)

Bayes Rule

- Conditional probabilities

$$p(\,o \wedge S\,) \;=\; p(o|S)p(S)$$

$$p(\,o \wedge S\,) \;=\; p(S|o)p(o)$$

- Bayes rule relates conditional probabilities

$$\boxed{p(\,o \mid S\,) \;=\; \frac{P(S|o)\,p(o)}{p(\,S\,)}}$$

Bayes rule

Can we update easily ?

(Courtesy of Dodds)

Combining evidence (sensor fusion)

So, how do we combine evidence to create a map?

What we want --

$odds(o|S_2 \wedge S_1)$ — the new value of a cell in the map after the sonar reading S_2

What we know --

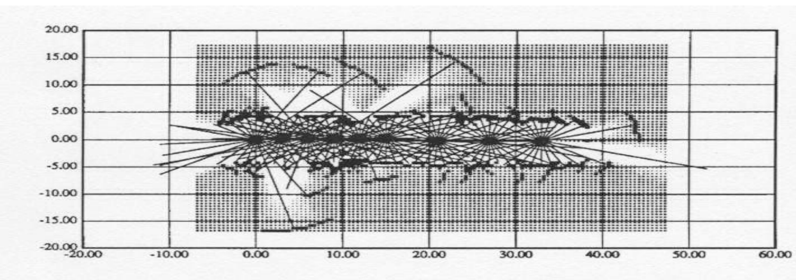
$odds(o | S_1)$ — the old value of a cell in the map (before sonar reading S_2)

$p(S_i|o)$&$p(S_i|\bar{o})$ — the probabilities that a certain obstacle causes the sonar reading S_i

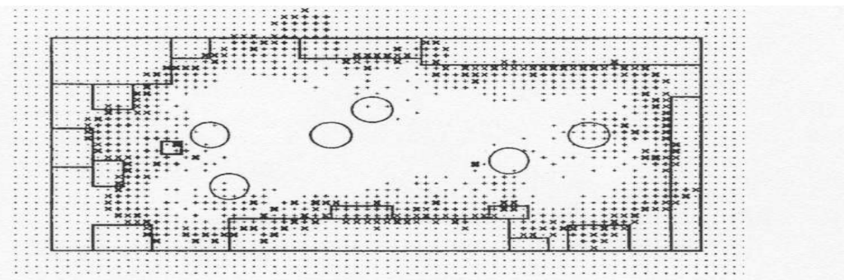
(Courtesy of Dodds)

Evidence grids

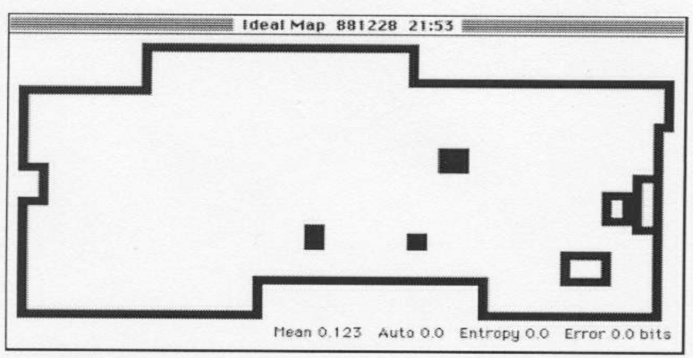
hallway with some open doors

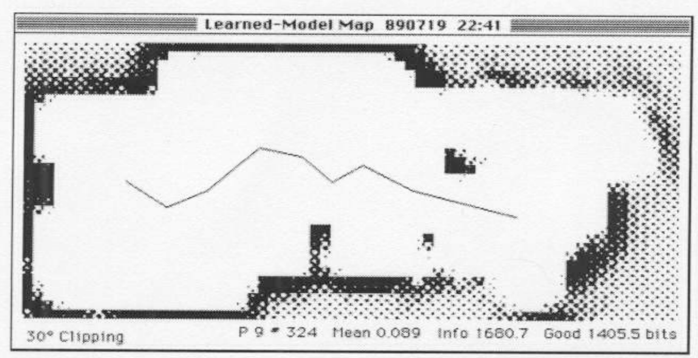


lab space



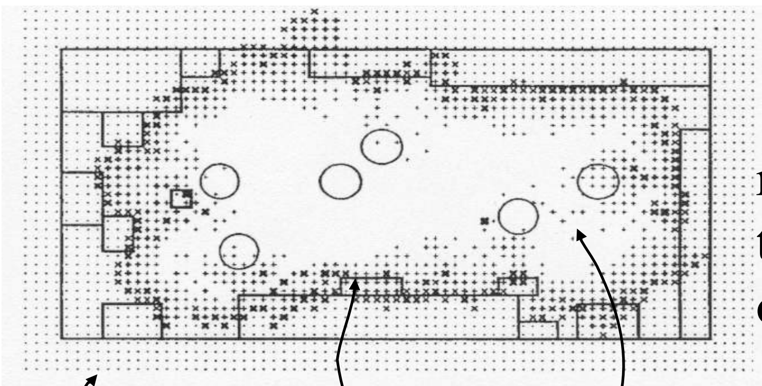
known map and estimated evidence grid





(Courtesy of Dodds)

Robot Mapping



Evidence Grids...

represent space as a collection of cells, each with the odds (or probability) that it contains an obstacle

Lab environment

likely obstacle

likely free space

not sure

- The relative locations of the robot within the map are assumed known.
- It is important that the robot odometry is correct
- Equally plausible to consider the converse problem...

Given a map of the environment, how do I determine where I am?

(Courtesy of Dodds)

"Robot localization problem"

SLAM Lecture Outline

- SLAM
- Robot Sensing and Localization
- Robot Mapping
- **Robot Motion Models**

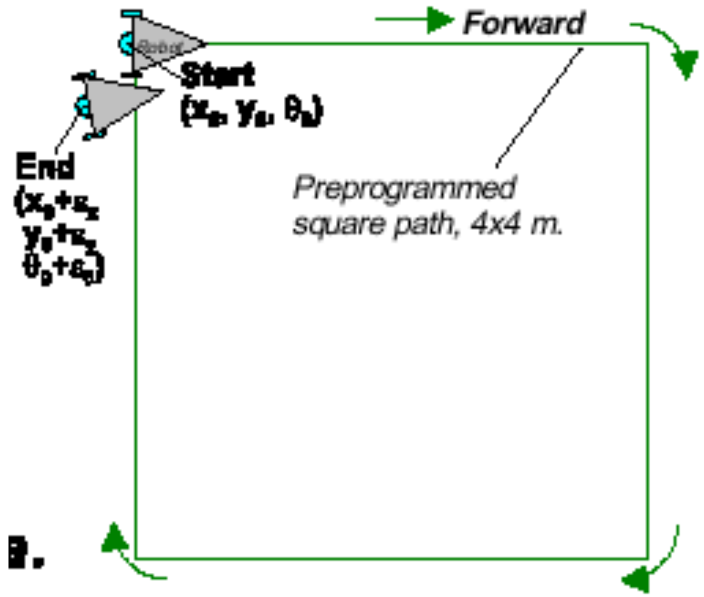
Typical Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based (dead reckoning)**
- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.

Dead Reckoning

- Derived from “deduced reckoning” though this is greatly disputed (see the straight dope)
- Mathematical procedure for determining the present location of a vehicle.
- Achieved by calculating the current pose of the vehicle based on its velocities and the time elapsed.

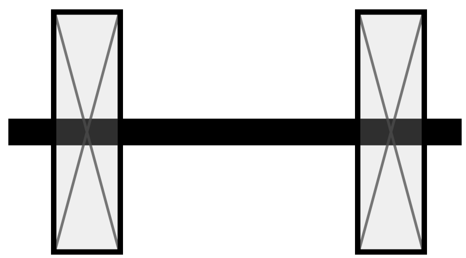
Dead Reckoning



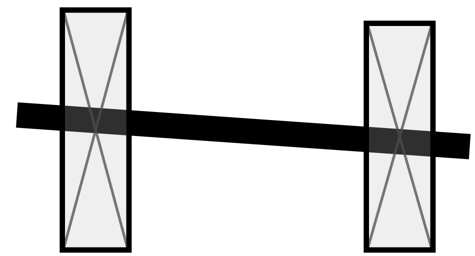
- Integration of incremental motion over time
- Given known start position/ orientation (pose)
- Given relationship between motor commands and robot displacement (linear and rotational)
- Compute current robot pose with simple geometric equations
- Provides good short-term relative position accuracy
- Accumulation of errors in long-term – wheel slippage, bumps, etc.,

(From Borenstein et. al.)

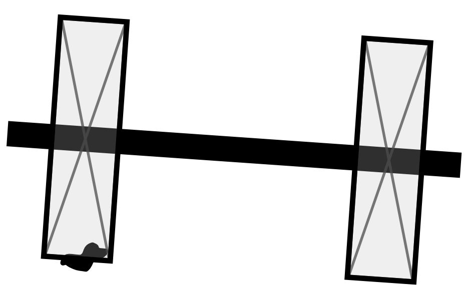
Reasons for Motion Errors



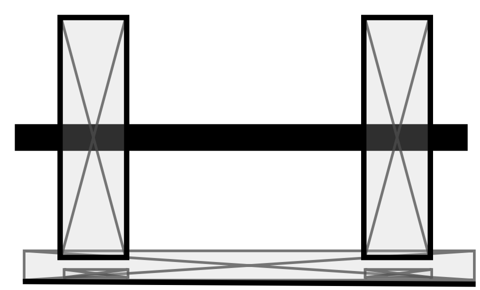
ideal case



different wheel diameters



bump



carpet

and many more …

Reducing Odometry Error with Absolute Measurements

- Uncertainty Ellipses
- Change shape based on other sensor information
- Artificial/natural landmarks
- Active beacons
- Model matching – compare sensor-induced features to features of known map – geometric or topological

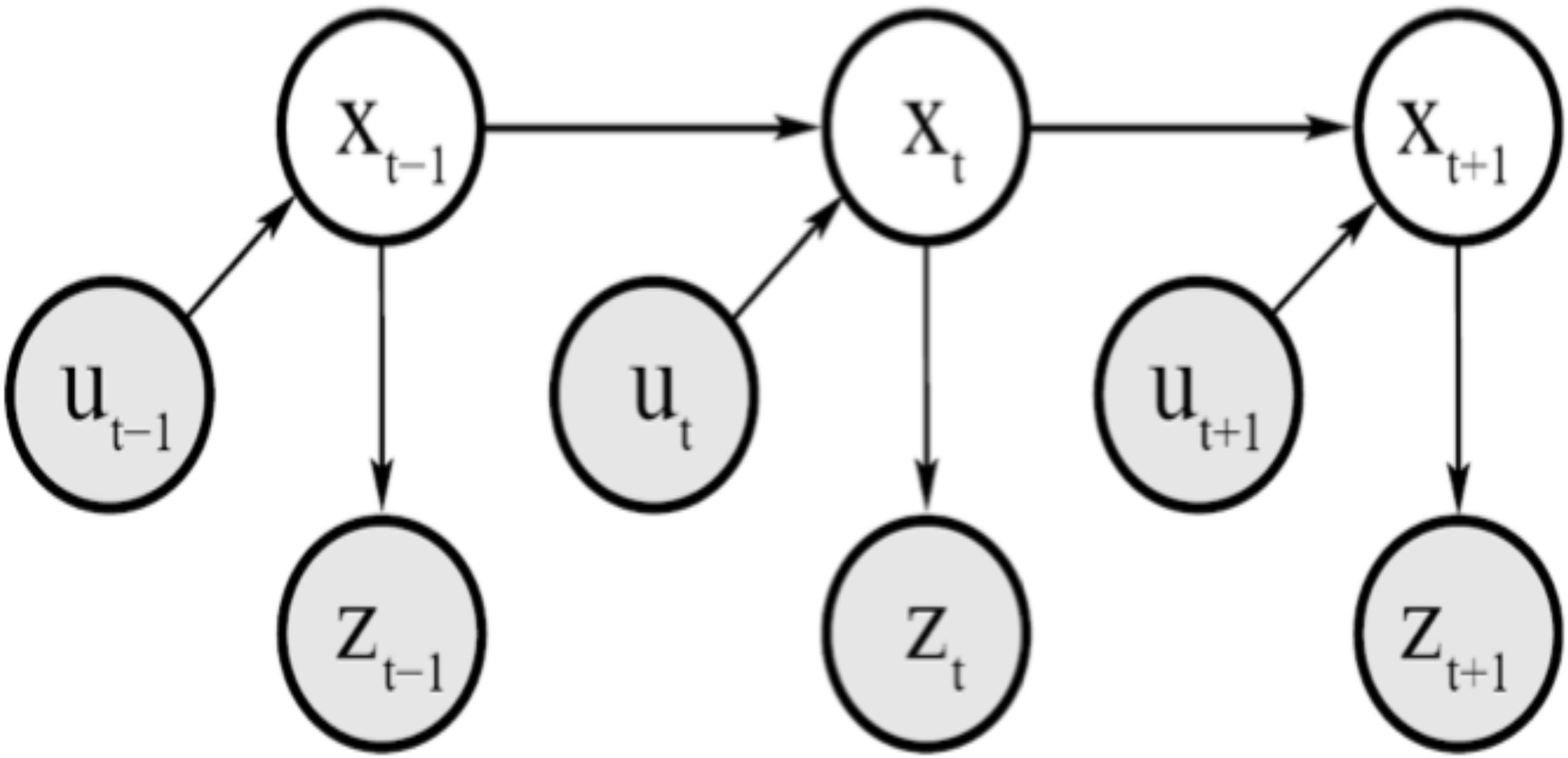
Types of Sensors

- Odometry
- Laser Ranging and Detection (LIDAR)
- Acoustic (sonar, ultrasonic)
- Radar
- Vision (monocular, stereo etc.)
- GPS
- Gyroscopes, Accelerometers (Inertial Navigation)
- Etc.

Sensor Characteristics

- Noise
- Dimensionality of Output
  - LIDAR- 3D point
  - Vision- Bearing only (2D ray in space)
- Range
- Frame of Reference
  - Most in robot frame (Vision, LIDAR, etc.)
  - GPS earth centered coordinate frame
  - Accelerometers/Gyros in inertial coordinate frame

Dynamic Bayesian Network for Controls, States, and Sensations



Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model $p(x \mid x', u)$.
- The term $p(x \mid x', u)$ specifies a posterior probability, that action u carries the robot from x' to x.
- $p(x \mid x', u)$ can be modeled based on the motion equations.

A Probabilistic Approach

- The following algorithms take a probabilistic approach

$$\mathrm{p}(x_t, m \mid z_{1:t}, u_{1:t})$$

$$x_t = \text{State of the robot at time } t$$

$$m = \text{Map of the environment}$$

$$z_{1:t} = \text{Sensor inputs from time 1 to } t$$

$$u_{1:t} = \text{Control imputs from time 1 to } t$$

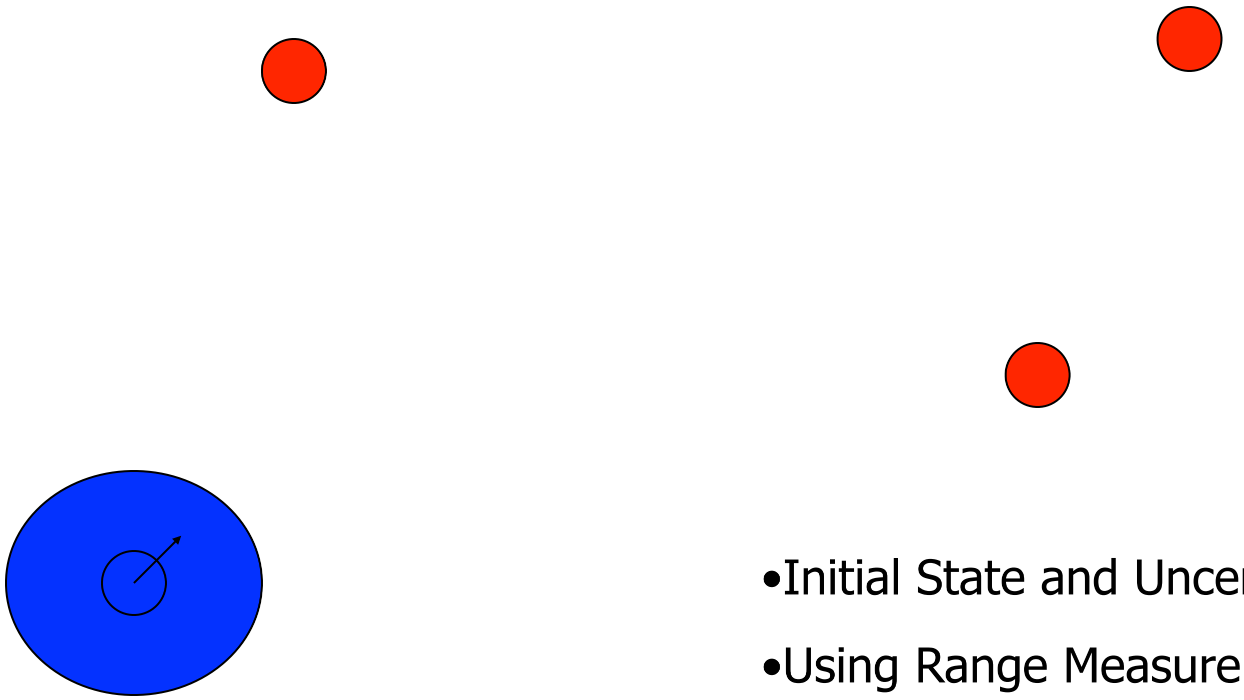
Two Example SLAM Algorithms

- Extended Kalman Filter (EKF) SLAM
  - Solves online SLAM problem
  - Uses a linearized Gaussian probability distribution model
- FastSLAM
  - Solves full SLAM problem
  - Uses a sampled particle filter distribution model

Extended Kalman Filter SLAM

- Solves the Online SLAM problem using a linearized Kalman filter
- One of the first probabilistic SLAM algorithms
- Not used frequently today but mainly shown for its explanatory value

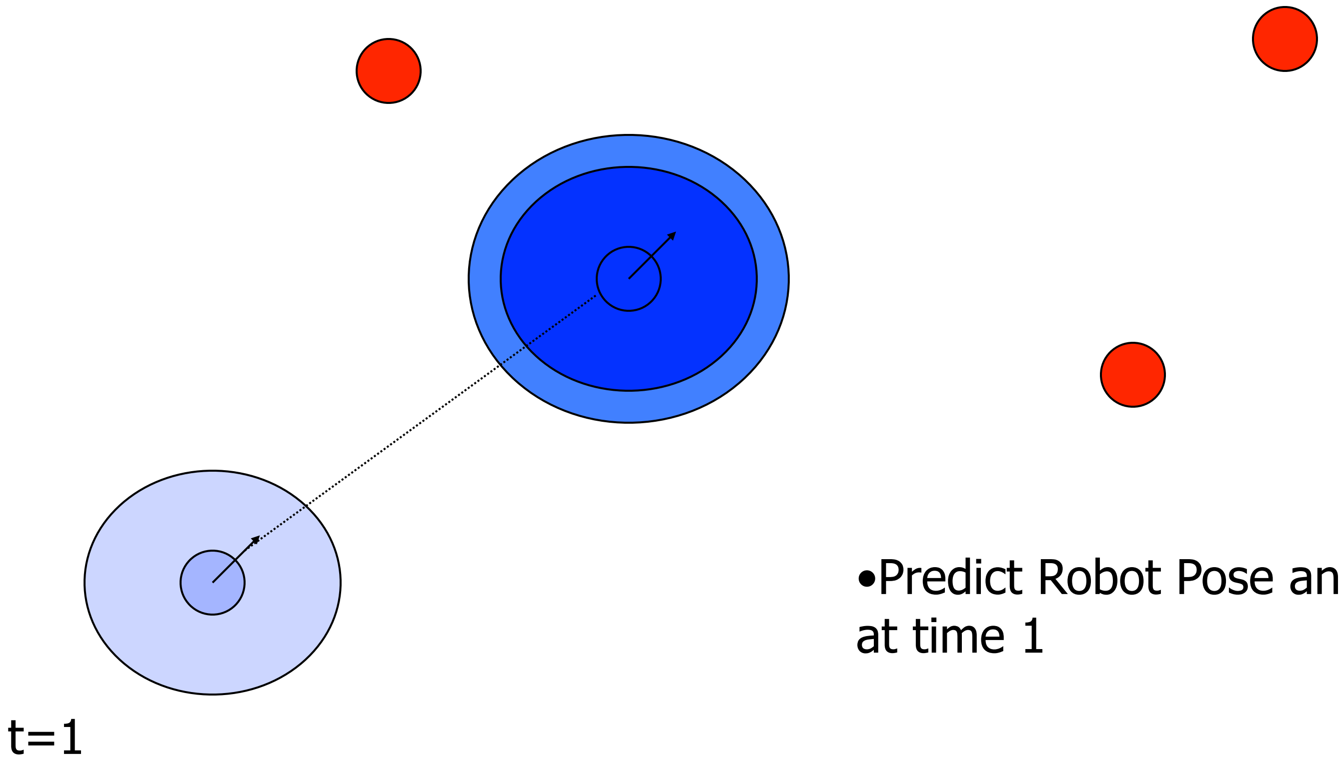
EKF Example



t=0

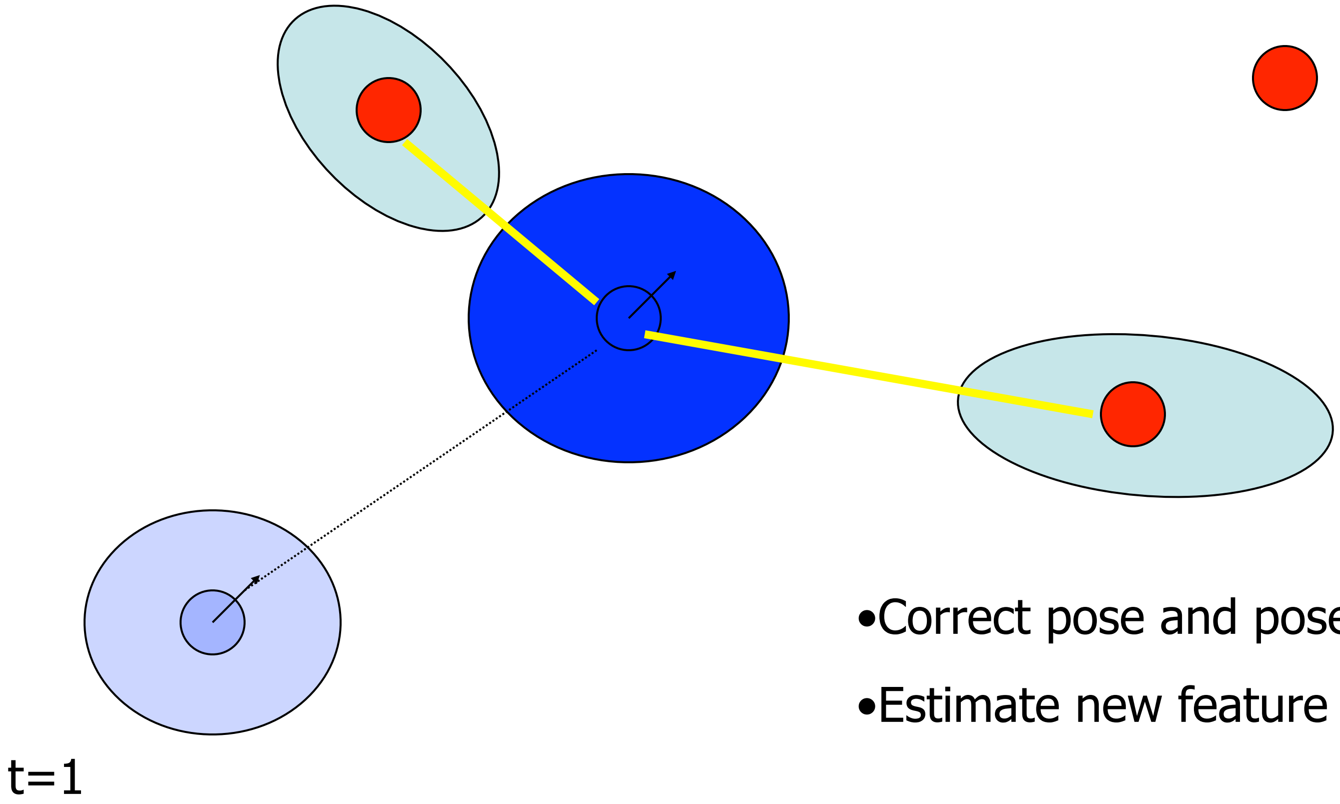
- Initial State and Uncertainty
- Using Range Measurements

EKF Example



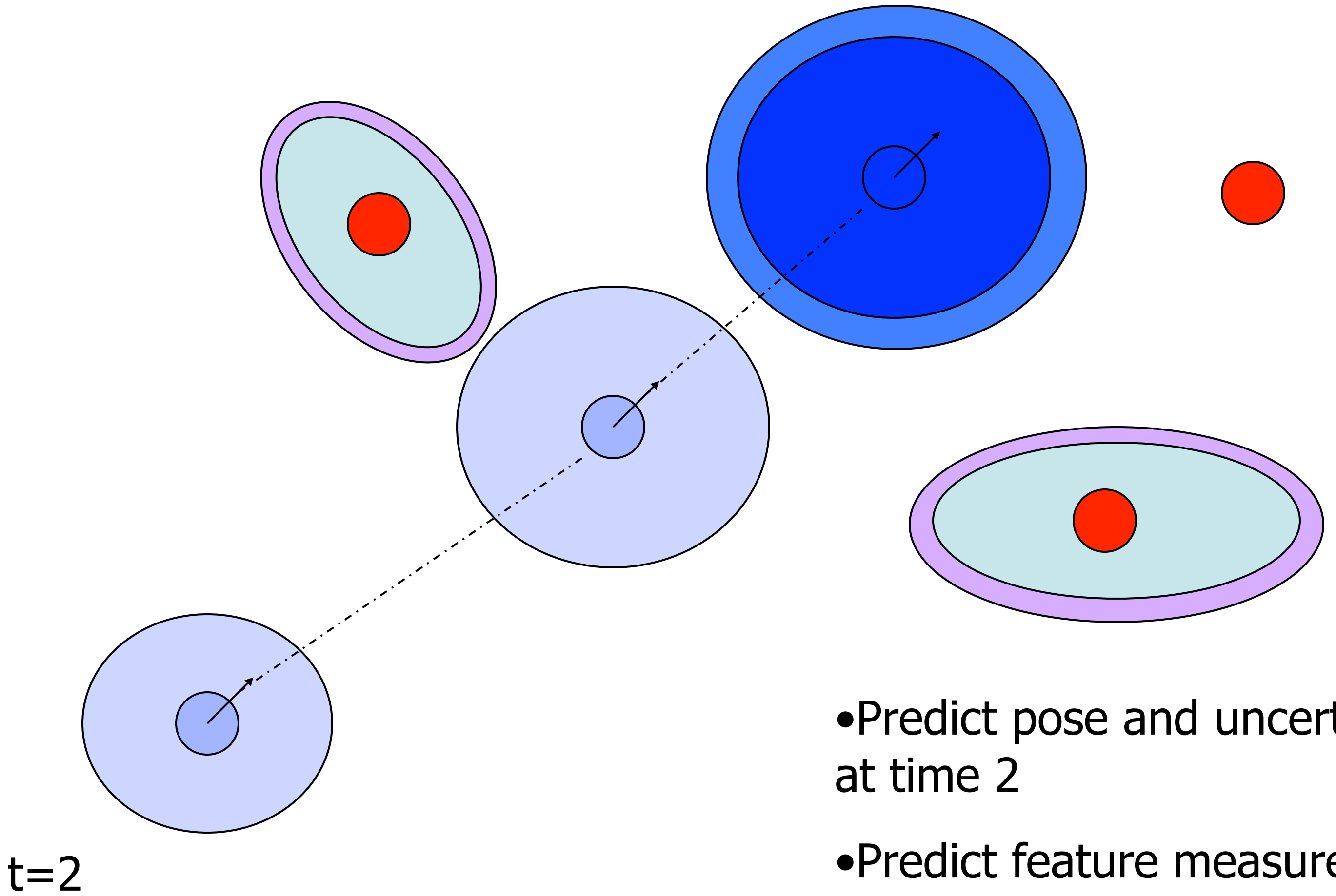
- Predict Robot Pose and Uncertainty at time 1

EKF Example



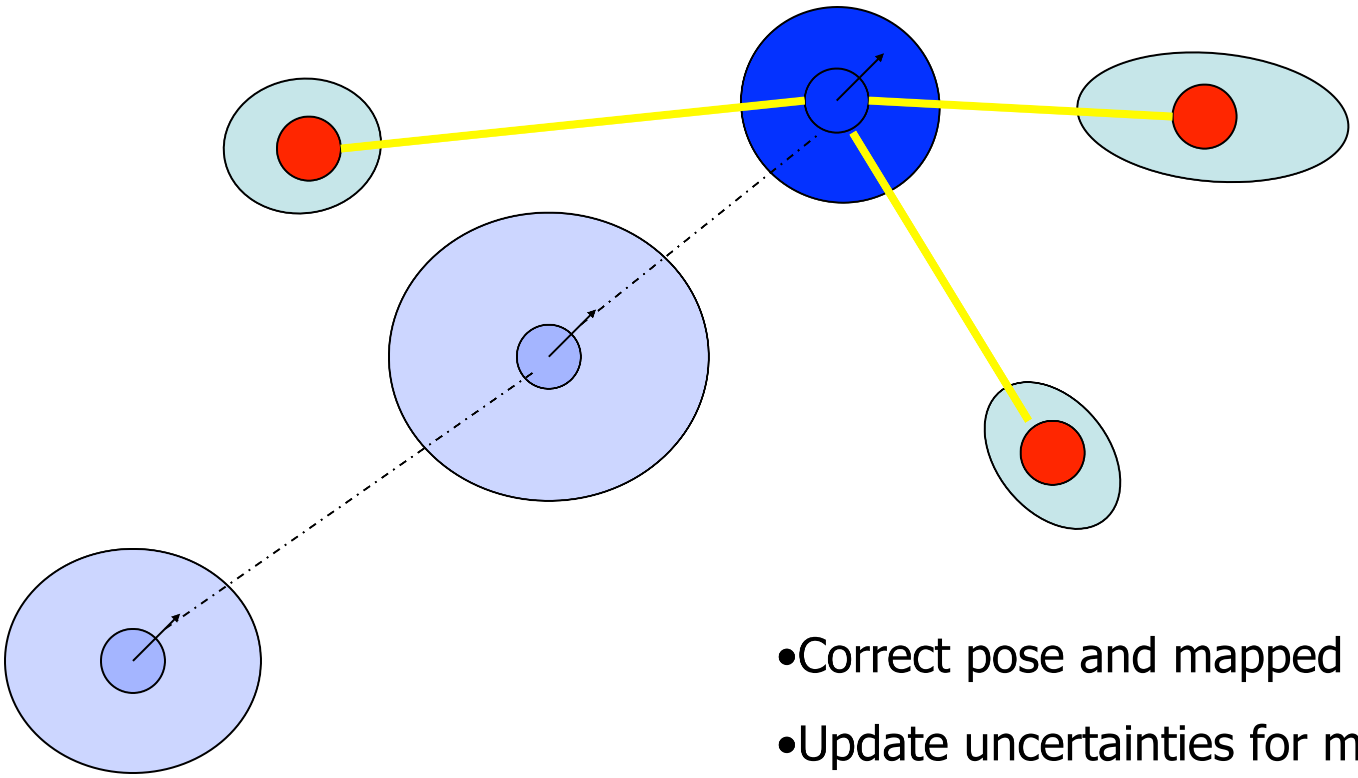
- Correct pose and pose uncertainty
- Estimate new feature uncertainties

EKF Example



- Predict pose and uncertainty of pose at time 2
- Predict feature measurements and their uncertainties

EKF Example



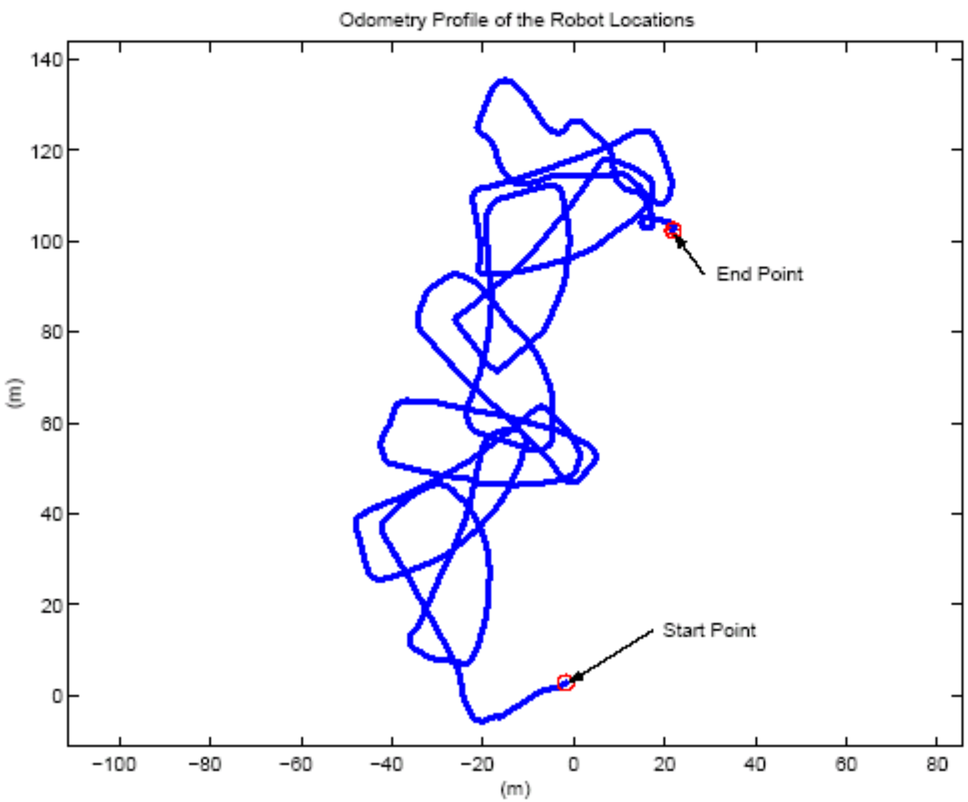
t=2

- Correct pose and mapped features
- Update uncertainties for mapped features
- Estimate uncertainty of new features

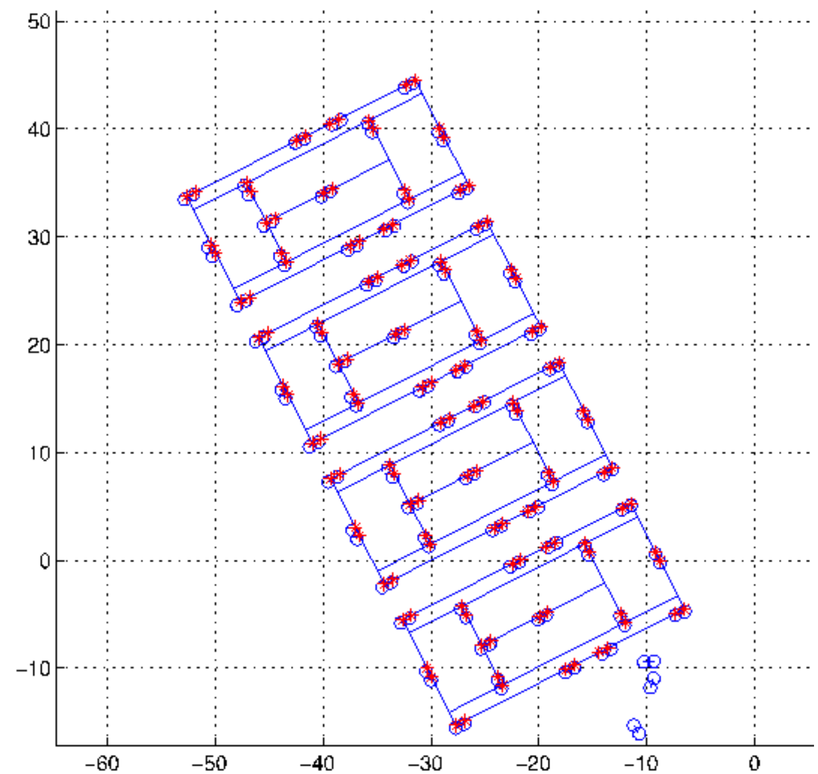
Application from Probabilistic Robotics

[courtesy by John Leonard]

Application from Probabilistic Robotics



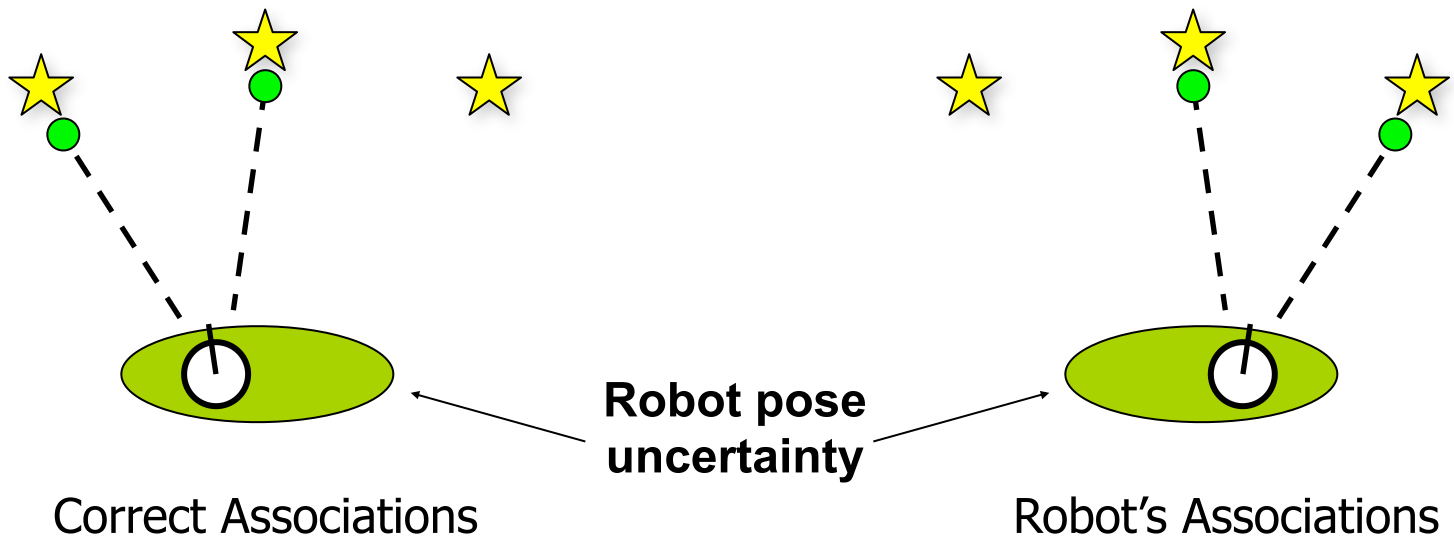
odometry



estimated trajectory

[courtesy by John Leonard]

Correlation Between Measurement Association and State Errors



- Association between measurements and features is unknown
- Errors in pose and measurement associations are correlated

Measurement Associations

- Measurements must be associated with particular features
  - If the feature is new add it to the map
  - Otherwise update the feature in the map
- Discrete decision must be made for each feature association, c_t

$$\mathrm{p}(x_t, m, c_t \mid z_{1:t}, u_{1:t})$$

$$x_t = \text{State of the robot at time } t$$

$$m = \text{Map of the environment}$$

$$c_t = \text{Measurement to feature associations a time t}$$

$$z_{1:t} = \text{Sensor inputs from time 1 to } t$$

$$u_{1:t} = \text{Control imputs from time 1 to } t$$

Problems With EKF SLAM

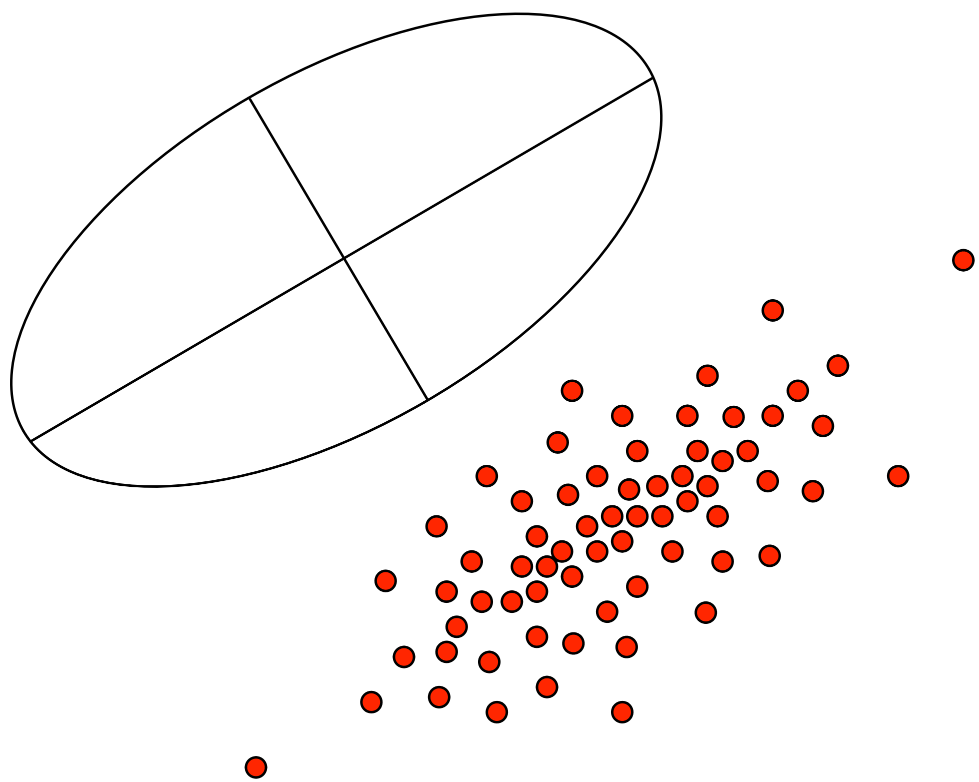
- Only one set of measurement to feature associations considered
  - Uses maximum likelihood association
  - Little chance of recovery from bad associations
- $O(N^3)$ matrix inversion required

FastSLAM

- Solves the Full SLAM problem using a particle filter

Particle Filters

- Represent probability distribution as a set of discrete particles which occupy the state space



Particle Filter Update Cycle

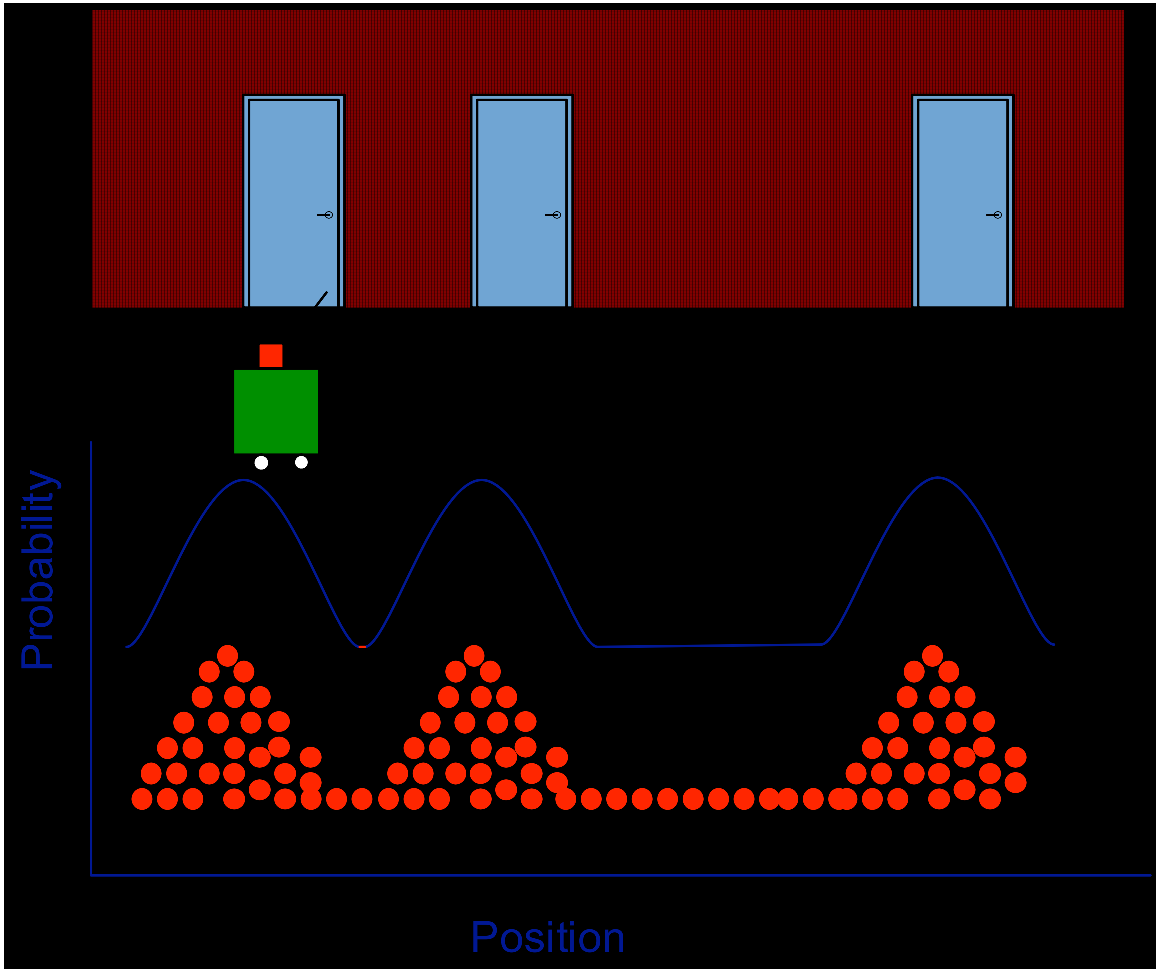
- Generate new particle distribution given motion model and controls applied
- For each particle
  - Compare particle's prediction of measurements with actual measurements
  - Particles whose predictions match the measurements are given a high weight
- Resample particles based on weight

Resampling

- Assign each particle a weight depending on how well its estimate of the state agrees with the measurements
- Randomly draw particles from previous distribution based on weights creating a new distribution

Particle Filter Advantages

- Can represent multi-modal distributions



Particle Filter Disadvantages

- Number of particles grows exponentially with the dimensionality of the state space
  - 1D – n particles
  - 2D – n^2 particles
  - mD – n^m particles

FastSLAM Formulation

- Decouple map of features from pose
  - Each particle represents a robot pose
  - Feature measurements are correlated thought the robot pose
  - If the robot pose was known all of the features would be uncorrelated
  - Treat each pose particle as if it is the true pose, processing all of the feature measurements independently

Factored Posterior (Landmarks)

poses map observations & movements

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

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

SLAM posterior

Robot path posterior

landmark positions

Factorization first introduced by Murphy in 1999

Factored Posterior

$$
\begin{aligned}
& p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) \\
&= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) \\
&= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^{M} p(l_i \mid x_{1:t}, z_{1:t})
\end{aligned}
$$

Robot path posterior (localization problem)

Conditionally independent landmark positions

Rao-Blackwellization

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

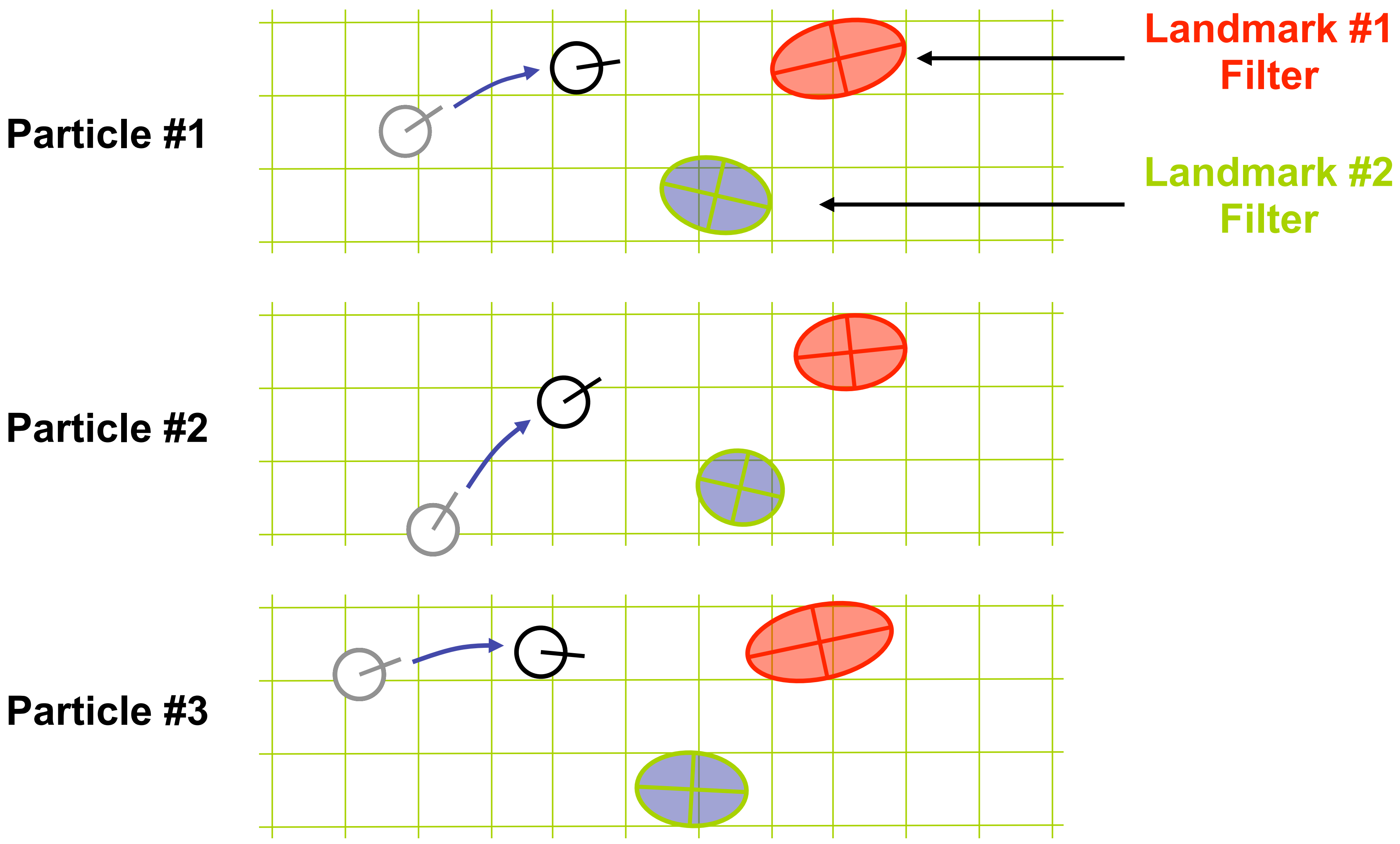
$$p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^{M} p(l_i \mid x_{1:t}, z_{1:t})$$

- Dimension of state space is drastically reduced by factorization making particle filtering possible

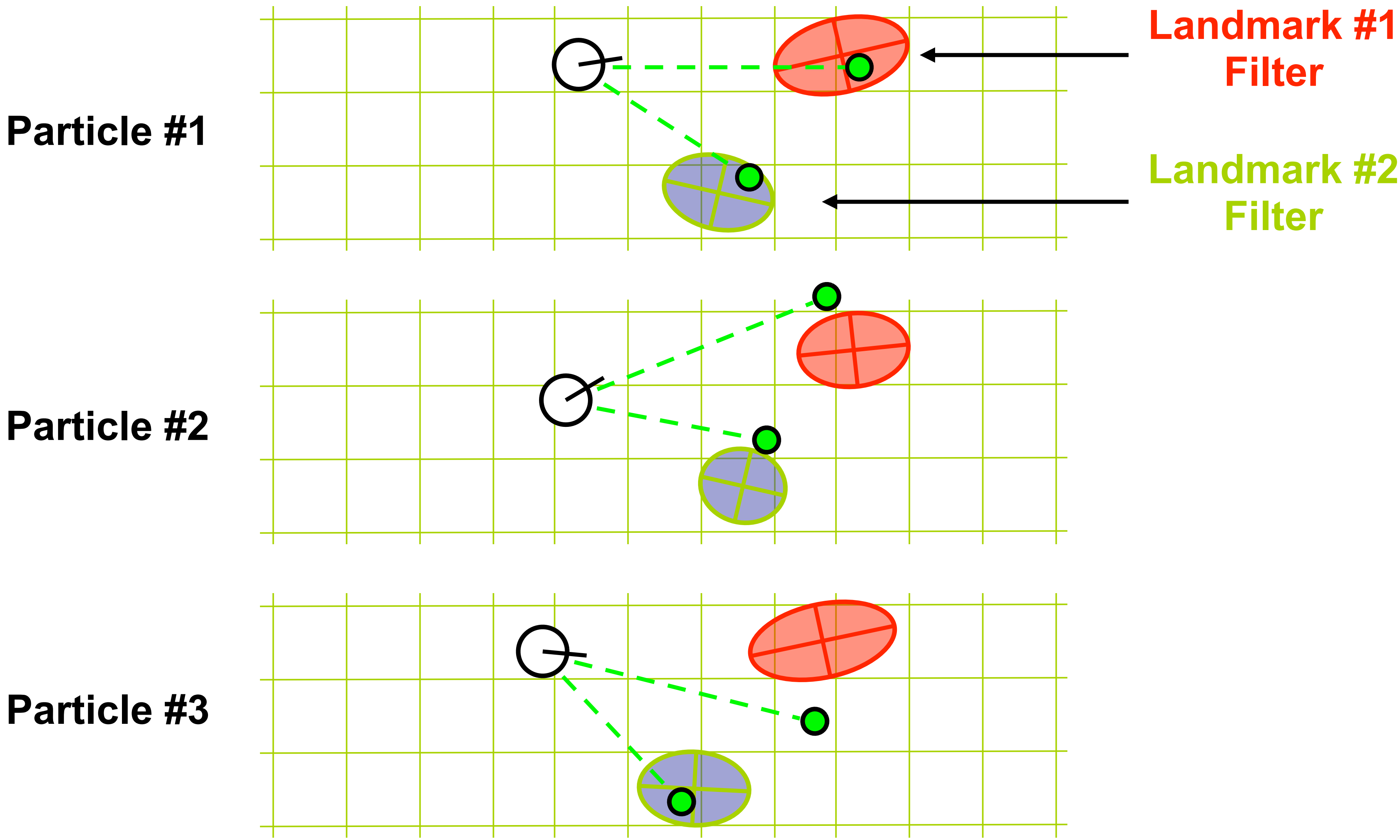
FastSLAM

- Rao-Blackwellized particle filtering based on landmarks [Montemerlo et al., 2002]
- Each landmark is represented by a 2x2 Extended Kalman Filter (EKF)
- Each particle therefore has to maintain *M* EKFs

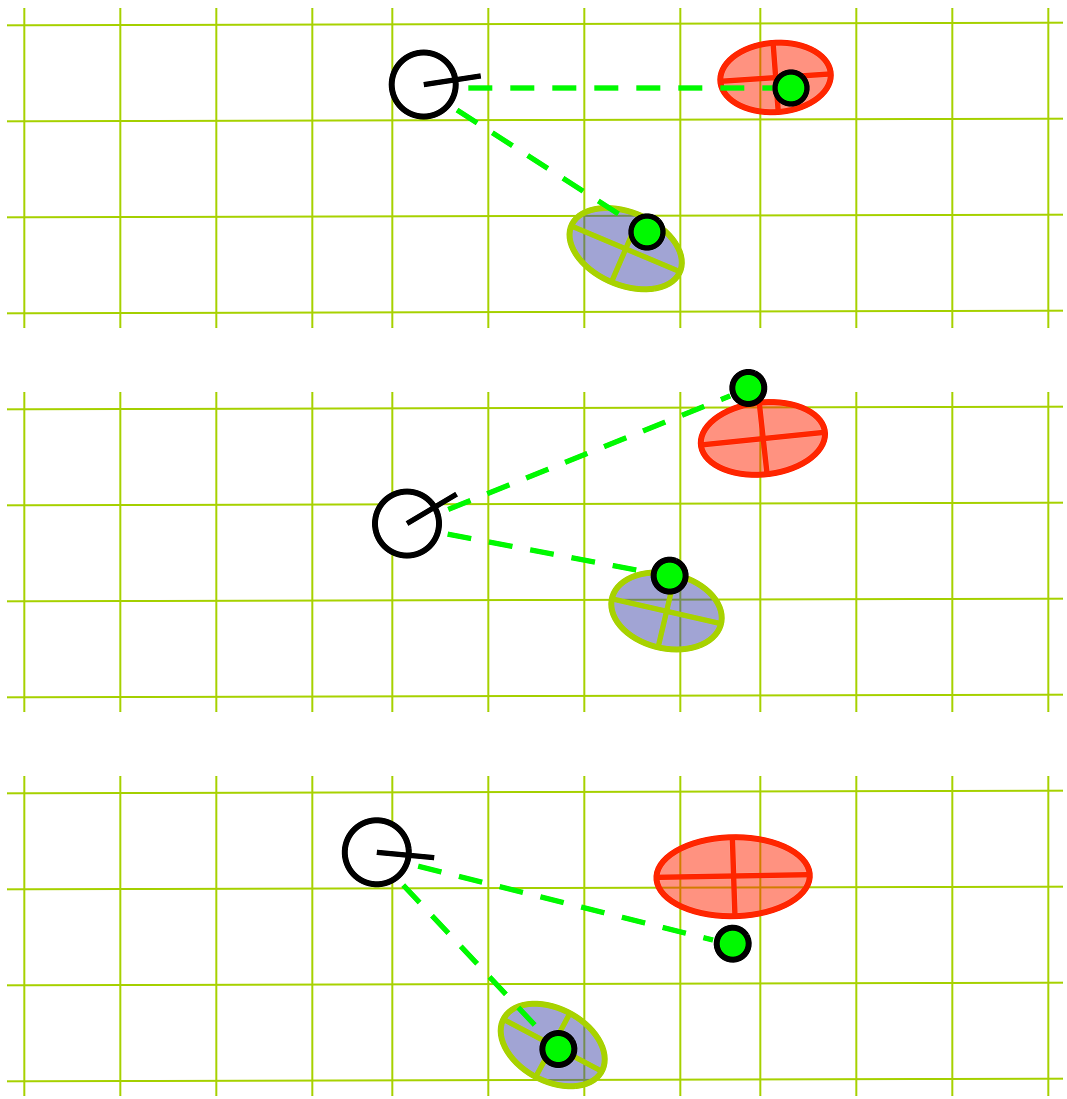
FastSLAM – Action Update



FastSLAM – Sensor Update



FastSLAM – Sensor Update



Particle #1 **Weight = 0.8**

Particle #2 **Weight = 0.4**

Particle #3 **Weight = 0.1**

FastSLAM Complexity

- Update robot particles based on control u_{t-1}

O(N)
Constant time per particle

- Incorporate observation z_t into Kalman filters

O(N•log(M))
Log time per particle

- Resample particle set

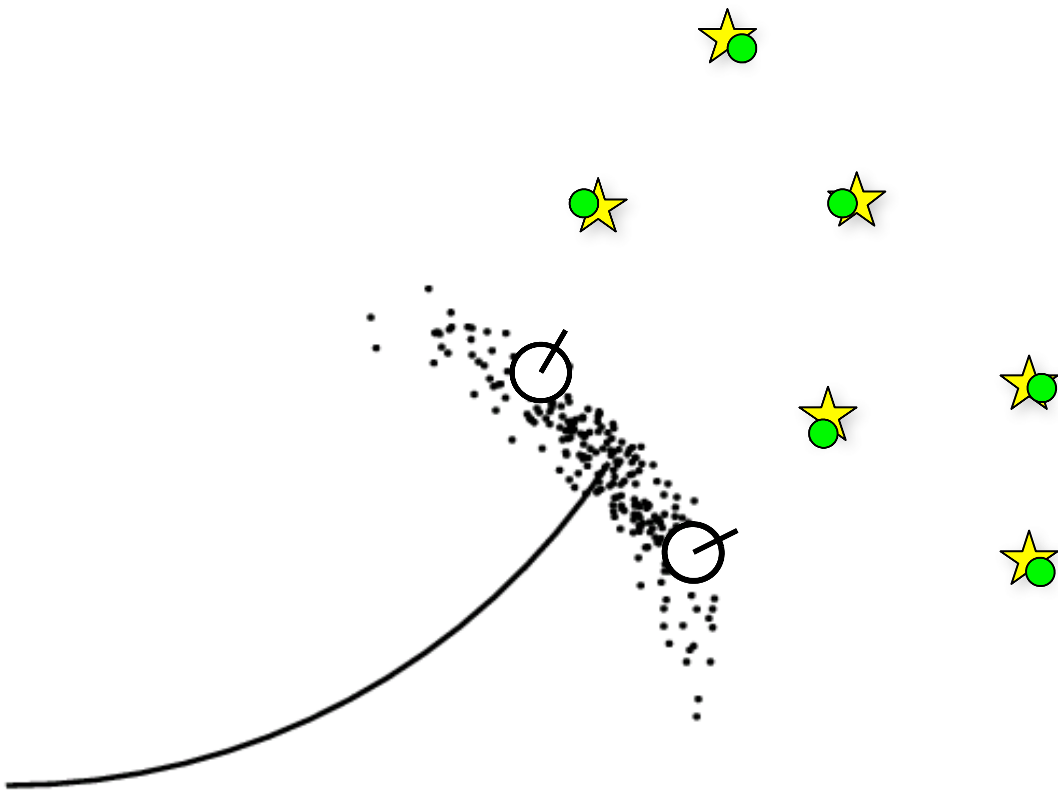
O(N•log(M))
Log time per particle

O(N•log(M))
Log time per particle

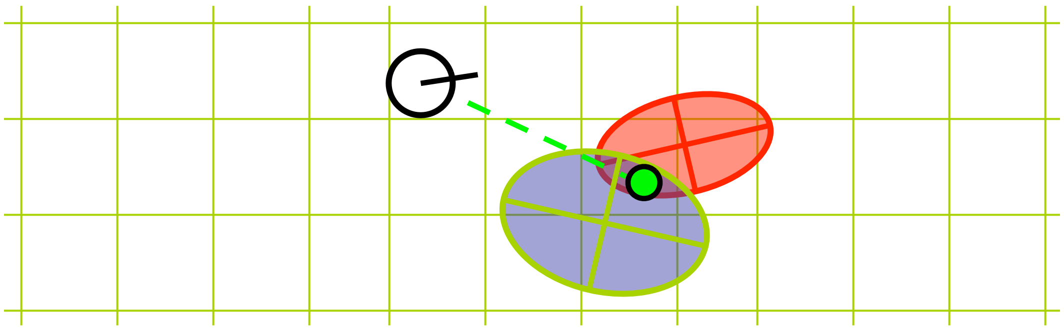
N = Number of particles
M = Number of map features

Multi-Hypothesis Data Association

- Data association is done on a per-particle basis
- Robot pose error is factored out of data association decisions



Per-Particle Data Association



Was the observation generated by the red or the blue landmark?

P(observation|red) = 0.3 P(observation|blue) = 0.7

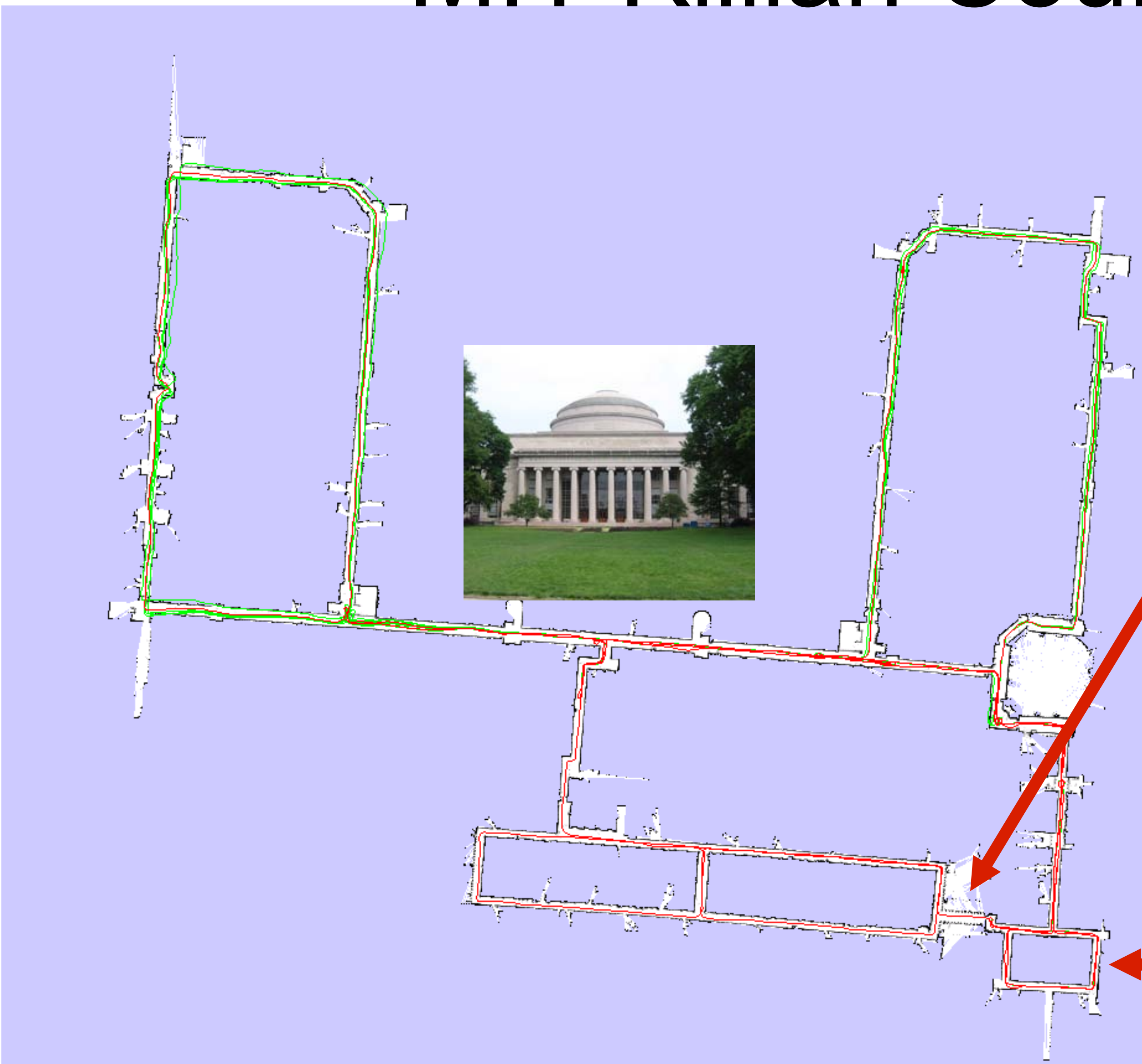
- Two options for per-particle data association
  - Pick the most probable match
  - Pick an random association weighted by the observation likelihoods
- If the probability is too low, generate a new landmark

MIT Killian Court



- The **"infinite-corridor-dataset"** at MIT

MIT Killian Court



Conclusion

- SLAM is a hard problem which is not yet fully solved
- Probabilistic methods which take account of sensor and process model error tend to work best
- Effective algorithms must be robust to bad data associations which EKF SLAM is not
- Real time operation limits complexity of algorithms which can be applied

References on EKF SLAM

- P. Moutarlier, R. Chatila, "Stochastic Multisensory Data Fusion for Mobile Robot Localization and Environment Modelling", In Proc. of the International Symposium on Robotics Research, Tokyo, 1989.

- R. Smith, M. Self, P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics", In Autonomous Robot Vehicles, I. J. Cox and G. T. Wilfong, editors, pp. 167-193, Springer-Verlag, 1990.

- Ali Azarbayejani, Alex P. Pentland, "Recursive Estimation of Motion, Structure, and Focal Length," *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,vol. 17, no. 6, pp. 562-575, June, 1995.

References on FastSLAM

- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping, *AAAI02*
- D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, IROS03
- M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. FastSLAM 2.0: An Improved particle filtering algorithm for simultaneous localization and mapping that provably converges. IJCAI-2003
- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, ICRA05
- A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultanous localization and mapping without predetermined landmarks, IJCAI03

Additional Reference

- Many of the slides for this presentation are from the book Probabilistic Robotic's website
  - http://www.probabilistic-robotics.org