EE565:Mobile Robotics

Welcome

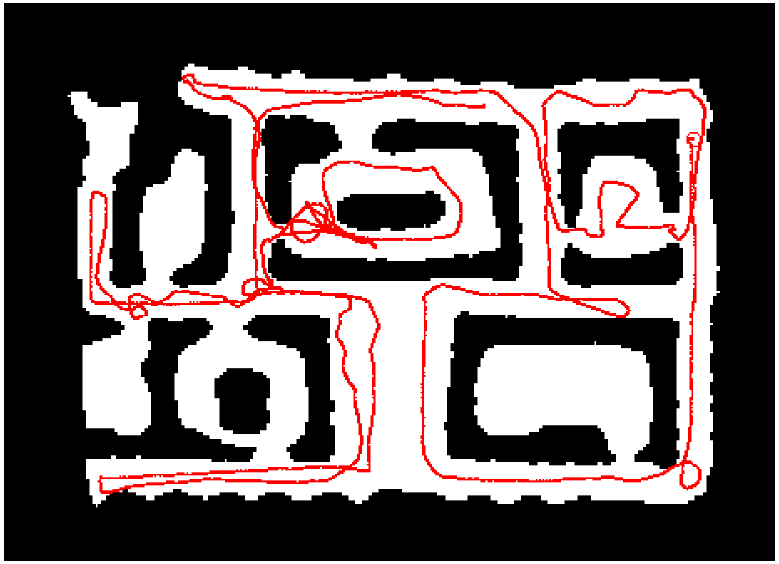
Dr. Ahmad Kamal Nasir

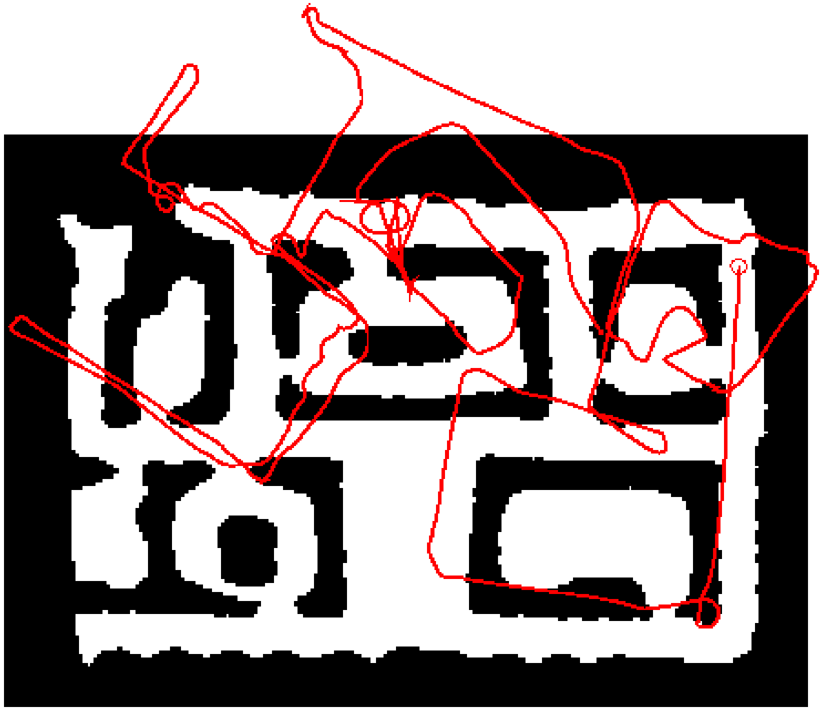
Today's Objectives

- Motion Models
  - Velocity based model (Dead-Reckoning)
  - Odometry based model (Wheel Encoders)
- Sensor Models
  - Beam model of range finders
  - Feature based sensor models
    - Camera
    - Laser scanner
    - Kinect

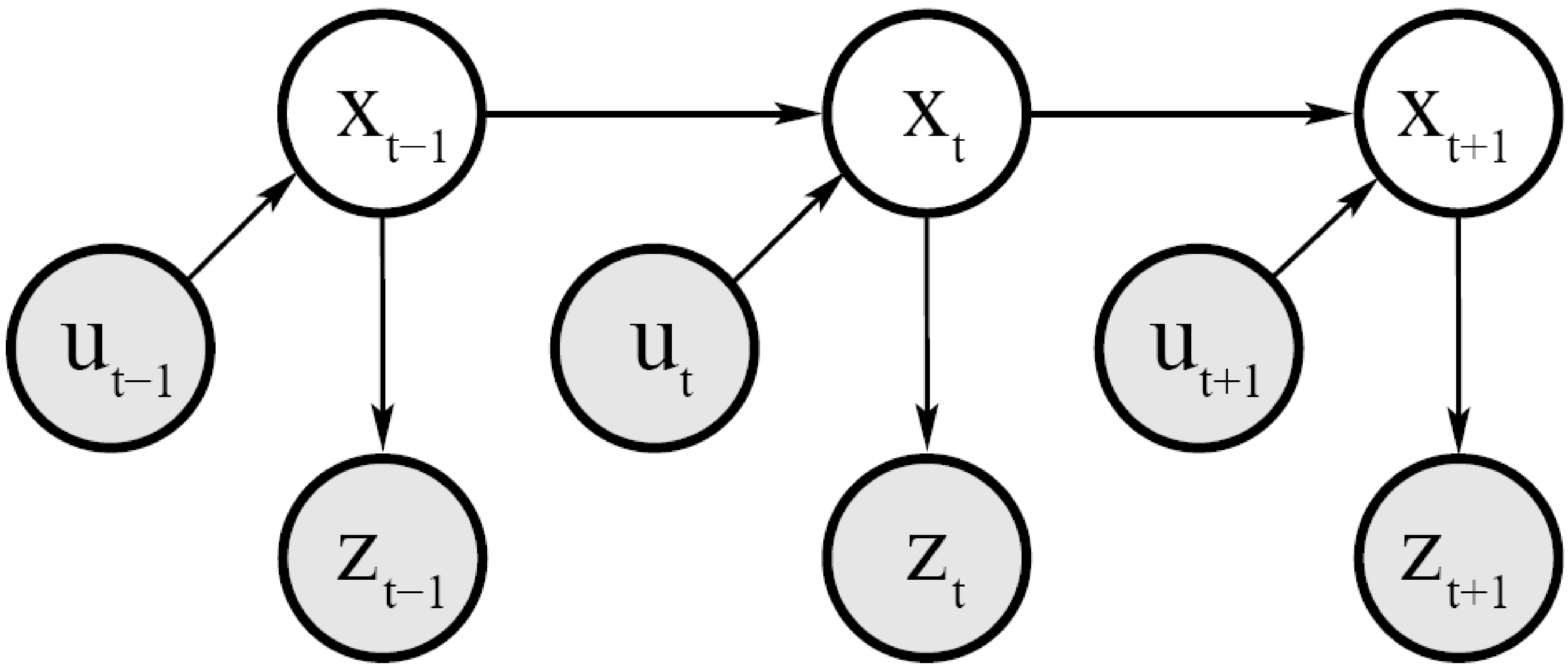
Robot Motion

- Robot motion is inherently uncertain.
- How can we model this uncertainty?





Dynamic Bayesian Network for Controls, States, and Sensations

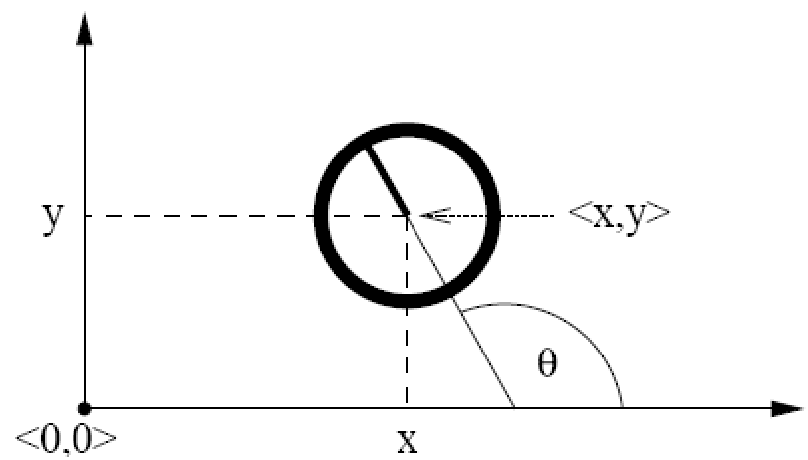


Probabilistic Motion Models

- 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.
- In this section we will specify, how $p(x \mid x', u)$ can be modeled based on the motion equations.

Coordinate Systems

- In general the configuration of a robot can be described by six parameters.
- Three-dimensional Cartesian coordinates plus three Euler angles pitch, roll, and tilt.
- Throughout this section, we consider robots operating on a planar surface.
- The state space of such systems is three-dimensional (x,y,θ).



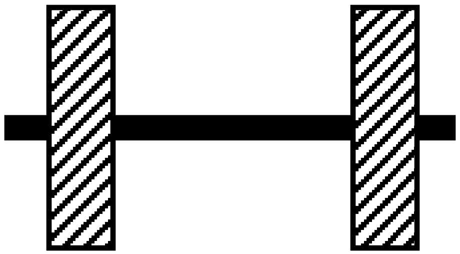
Typical Motion Models

- 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.

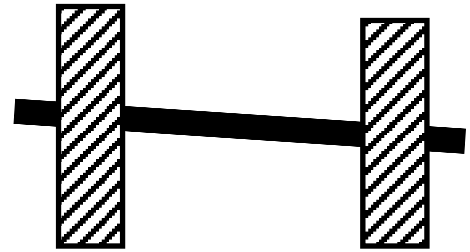
Dead Reckoning

- Derived from “deduced reckoning.”
- 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.

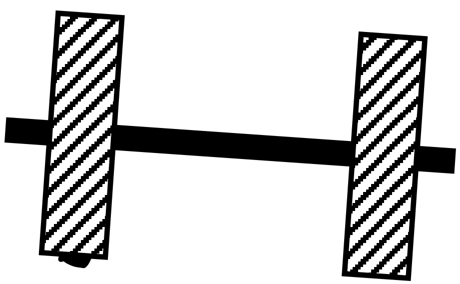
Reasons for Motion Errors



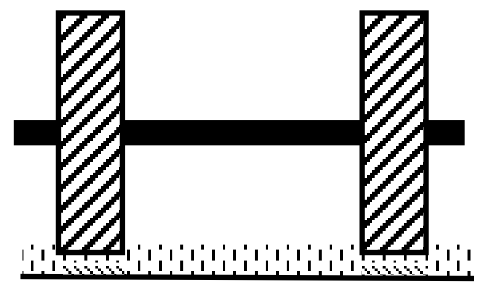
ideal case



different wheel diameters



bump

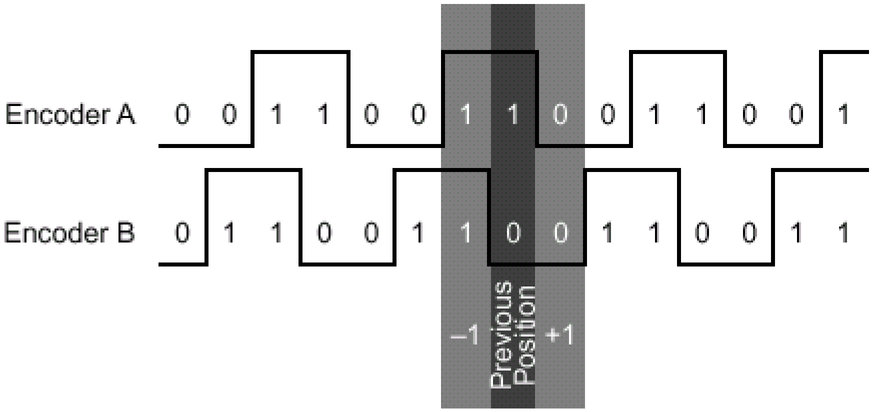


carpet

and many more …

Wheel Encoders

- A pair of encoders is used on a single shaft. The encoders are aligned so that their two data streams are one quarter cycle (90 deg.) out of phase.
- Which direction is shaft moving?
  - Suppose the encoders were previously at the position highlighted by the dark band; i.e., Encoder A as 1 and Encoder B as 0. The next time the encoders are checked:
  - If they moved to the position AB=00, the position count is incremented
  - If they moved to the position AB=11, the position count is decremented



Odometry Model

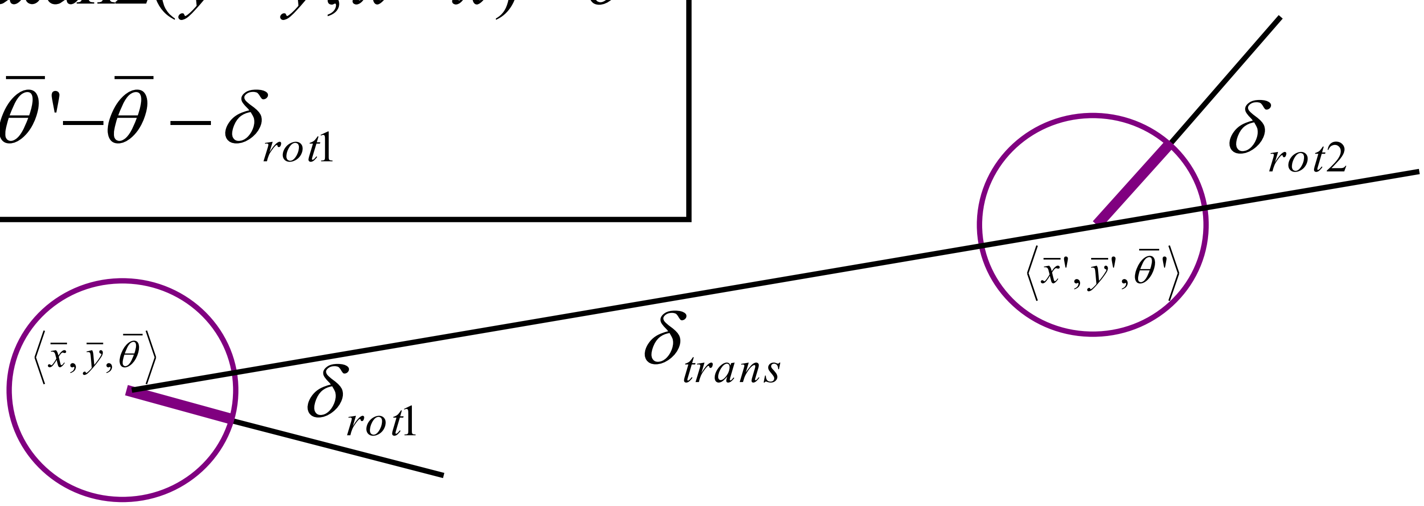
- Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$

Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

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

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

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



The atan2 Function

- Extends the inverse tangent and correctly copes with the signs of x and y.

$$\text{atan2}(y,x) = \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{sign}(y)\,(\pi - \text{atan}(|y/x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y)\,\pi/2 & \text{if } x = 0, y \neq 0 \end{cases}$$

Noise Model for Odometry

- The measured motion is given by the true motion corrupted with noise.

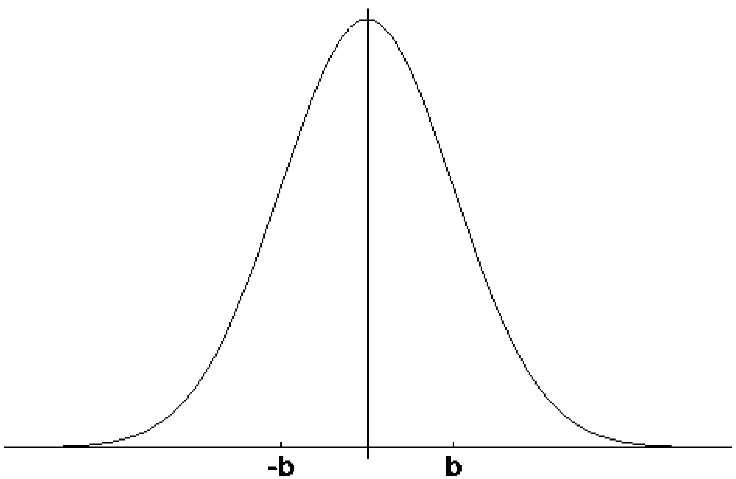
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|}$$

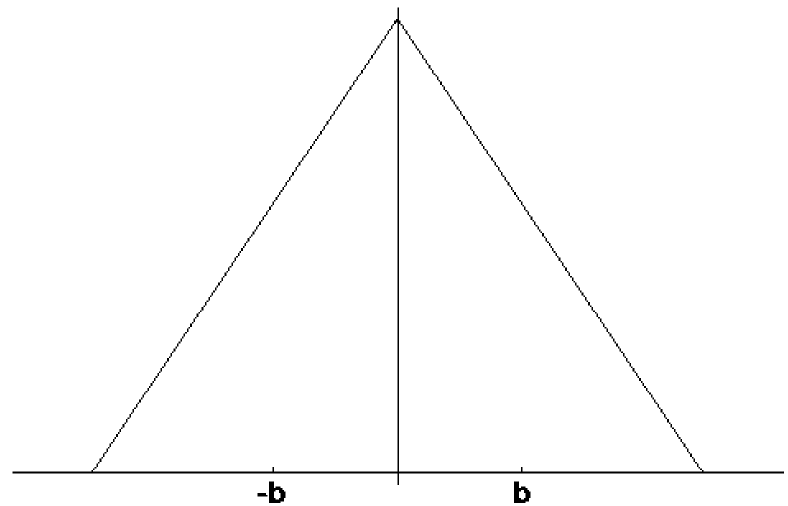
Typical Distributions for Probabilistic Motion Models

Normal distribution



$$\varepsilon_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$$

Triangular distribution



$$\varepsilon_{\sigma^2}(x) = \begin{cases} 0 \text{ if } |x| > \sqrt{6\sigma^2} \\ \dfrac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} \end{cases}$$

How to Sample from Normal or Triangular Distributions?

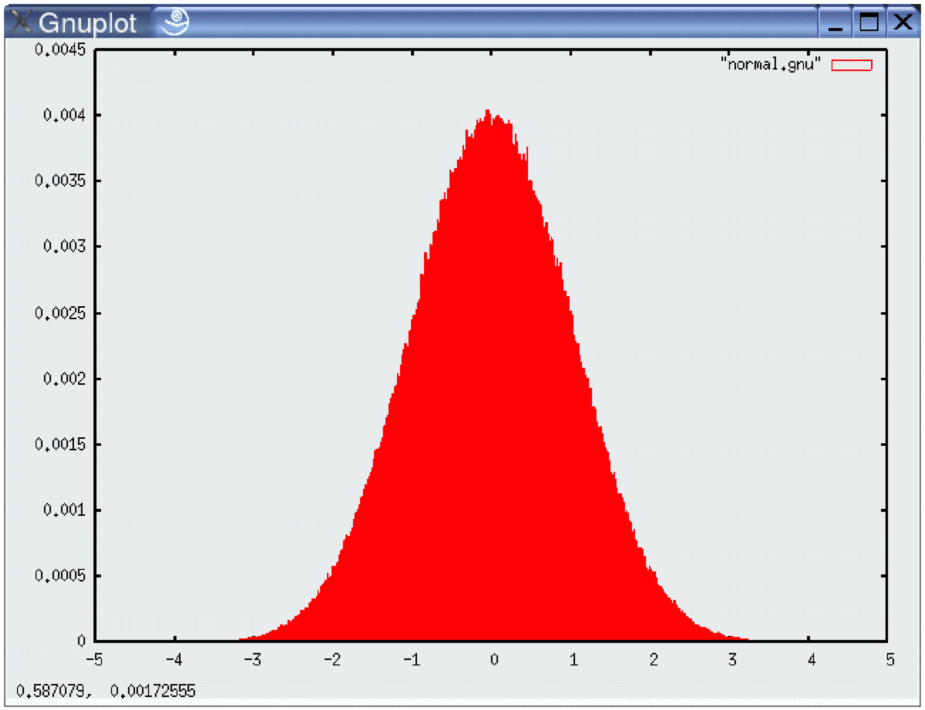
- Sampling from a normal distribution

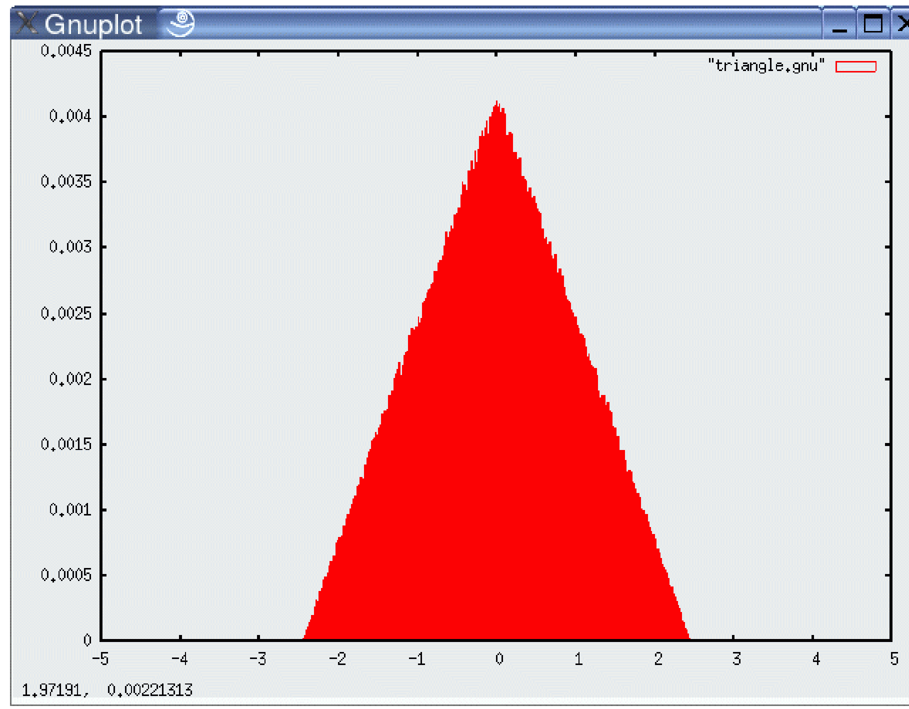
1. Algorithm **sample_normal_distribution**(b):
2. return $$\frac{1}{2}\sum_{i=1}^{12} rand(-b, b)$$

- Sampling from a triangular distribution

1. Algorithm **sample_triangular_distribution**(b):
2. return $$\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b, b) + \mathbf{rand}(-b, b)\right]$$

Normally/Triangular Distributed Samples





10^6 samples

Calculating the Posterior Given x, x', and u

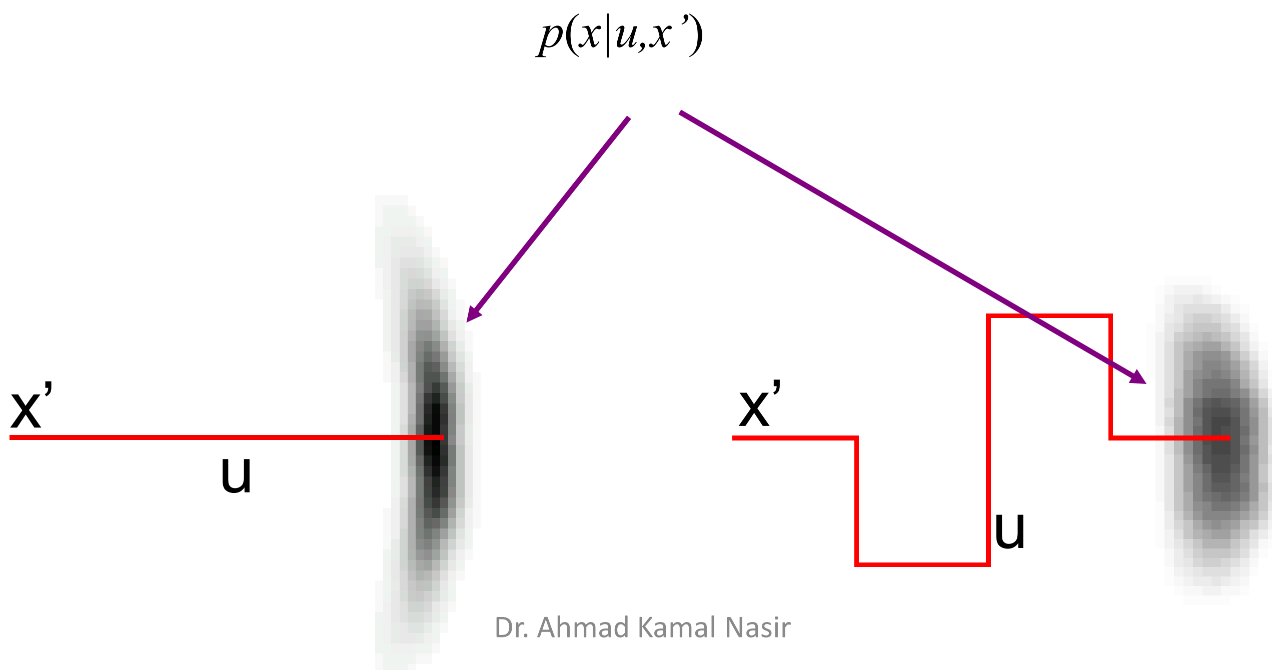
1. Algorithm **motion_model_odometry(x,x',u)**
2. $\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
3. $\delta_{rot1} = \mathrm{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
4. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x' - x)^2 + (y' - y)^2}$
6. $\hat{\delta}_{rot1} = \mathrm{atan2}(y' - y, x' - x) - \bar{\theta}$
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \mathrm{prob}(\delta_{\mathrm{rot1}} - \hat{\delta}_{\mathrm{rot1}}, \alpha_1 |\hat{\delta}_{\mathrm{rot1}}| + \alpha_2 \hat{\delta}_{\mathrm{trans}})$
9. $p_2 = \mathrm{prob}(\delta_{\mathrm{trans}} - \hat{\delta}_{\mathrm{trans}}, \alpha_3 \hat{\delta}_{\mathrm{trans}} + \alpha_4 (|\hat{\delta}_{\mathrm{rot1}}| + |\hat{\delta}_{\mathrm{rot2}}|))$
10. $p_3 = \mathrm{prob}(\delta_{\mathrm{rot2}} - \hat{\delta}_{\mathrm{rot2}}, \alpha_1 |\hat{\delta}_{\mathrm{rot2}}| + \alpha_2 \hat{\delta}_{\mathrm{trans}})$
11. return $p_1 \cdot p_2 \cdot p_3$

odometry values (u)

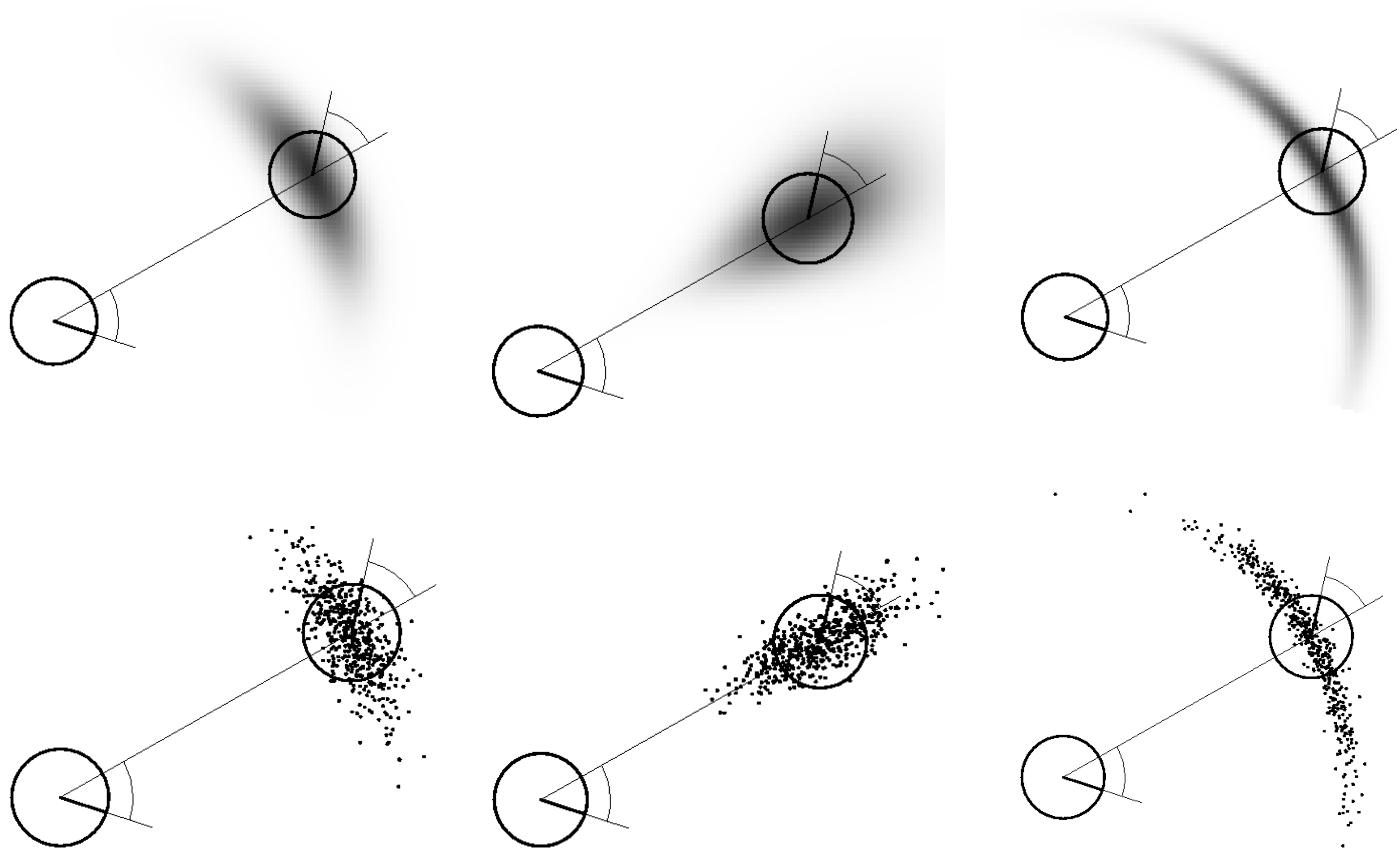
values of interest (x,x')

Application

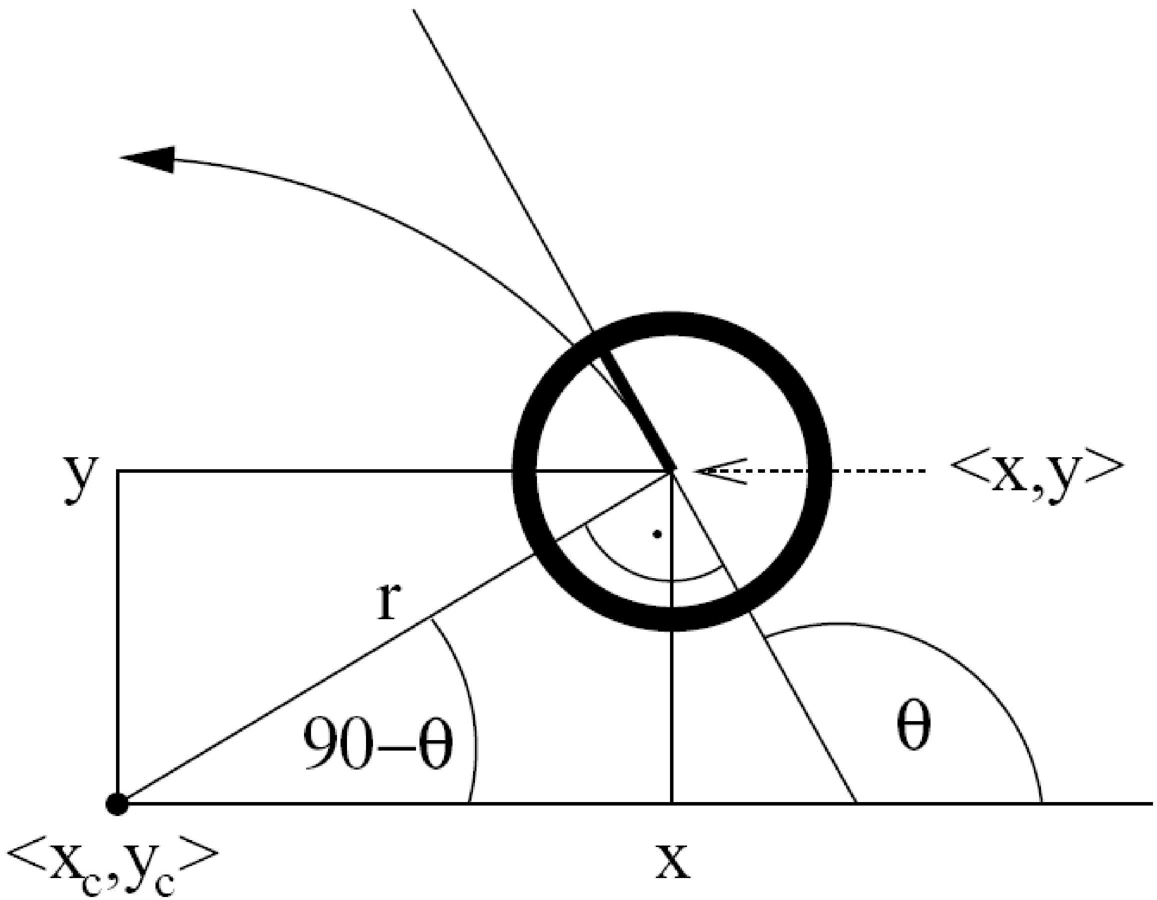
- Repeated application of the sensor model for short movements.
- Typical banana-shaped distributions obtained for 2d-projection of 3d posterior.



Examples (Odometry-Based)



Velocity-Based Model



Posterior Probability for Velocity Model

Algorithm motion_model_velocity(x_t, u_t, x_{t-1})**:**

$$\mu = \frac{1}{2} \frac{(x - x')\cos\theta + (y - y')\sin\theta}{(y - y')\cos\theta - (x - x')\sin\theta}$$

$$x^* = \frac{x + x'}{2} + \mu(y - y')$$

$$y^* = \frac{y + y'}{2} + \mu(x' - x)$$

$$r^* = \sqrt{(x - x^*)^2 + (y - y^*)^2}$$

$$\Delta\theta = \mathrm{atan2}(y' - y^*, x' - x^*) - \mathrm{atan2}(y - y^*, x - x^*)$$

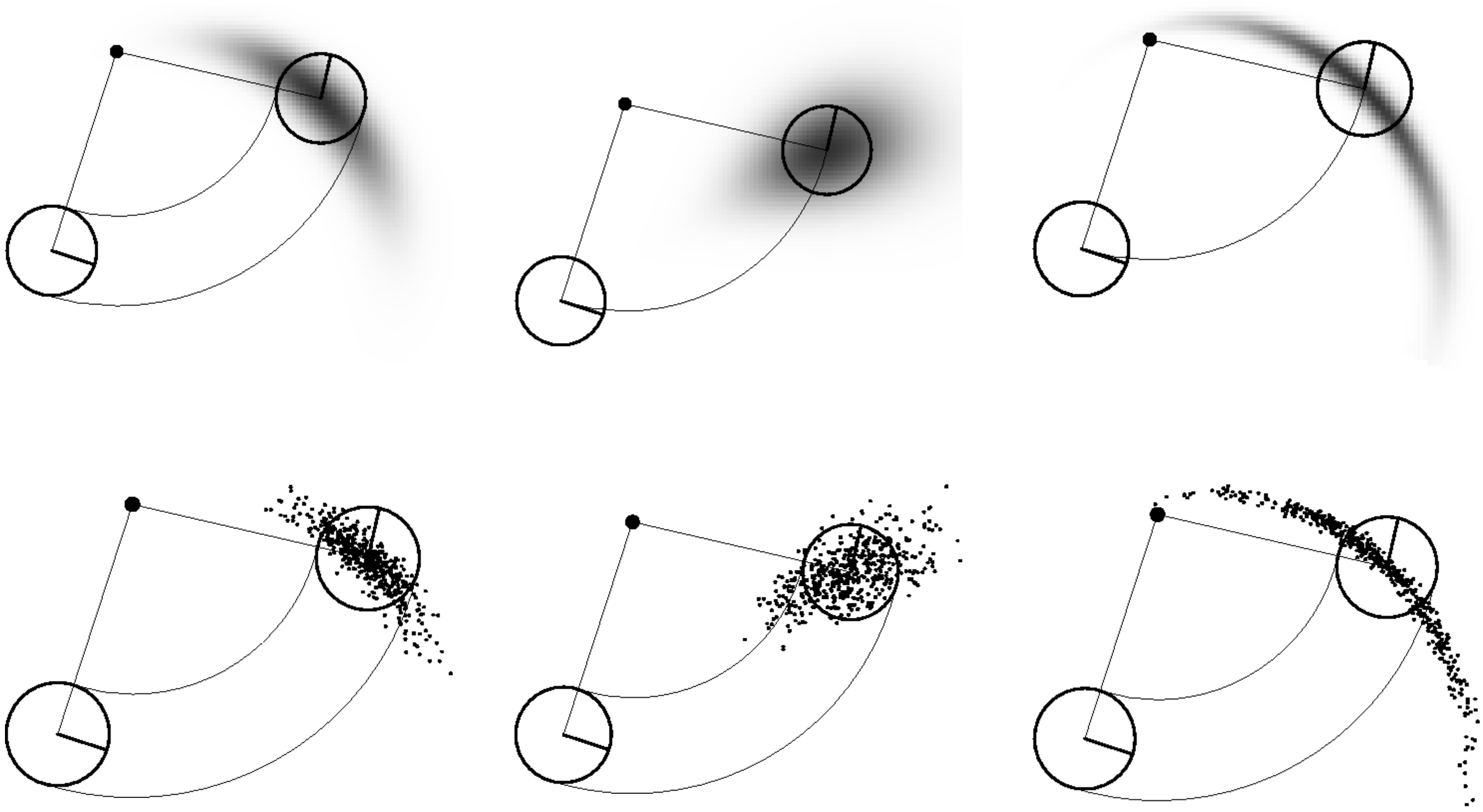
$$\hat{v} = \frac{\Delta\theta}{\Delta t}\, r^*$$

$$\hat{\omega} = \frac{\Delta\theta}{\Delta t}$$

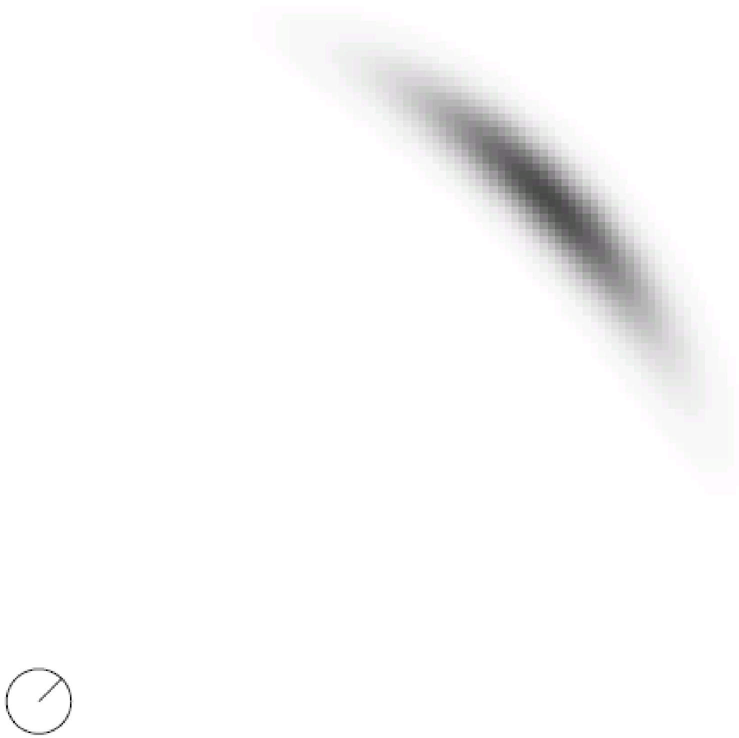
$$\hat{\gamma} = \frac{\theta' - \theta}{\Delta t} - \hat{\omega}$$

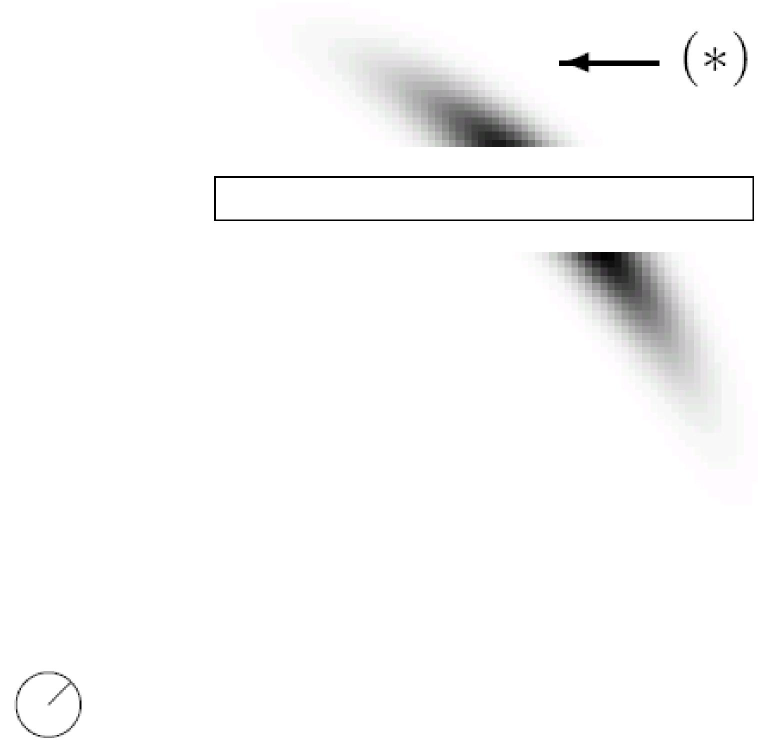
return $\mathbf{prob}(v - \hat{v}, \alpha_1|v| + \alpha_2|\omega|) \cdot \mathbf{prob}(\omega - \hat{\omega}, \alpha_3|v| + \alpha_4|\omega|) \cdot \mathbf{prob}(\hat{\gamma}, \alpha_5|v| + \alpha_6|\omega|)$

Examples (velocity based)



Map-Consistent Motion Model





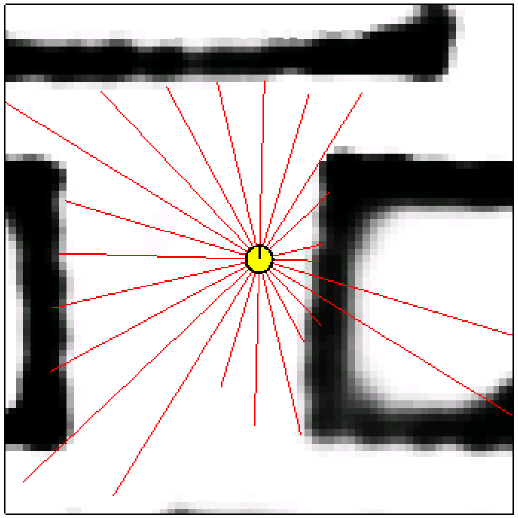
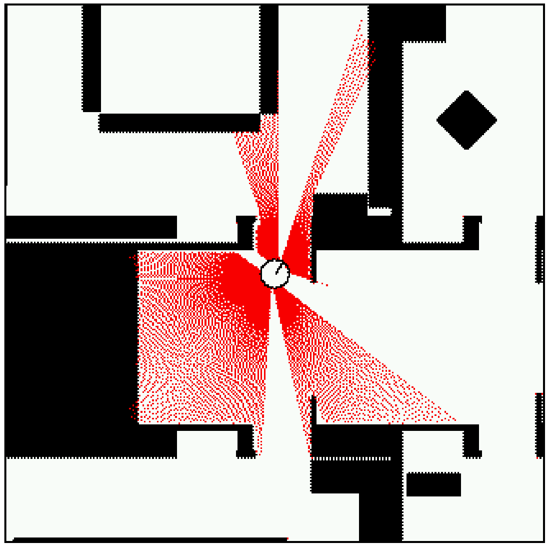
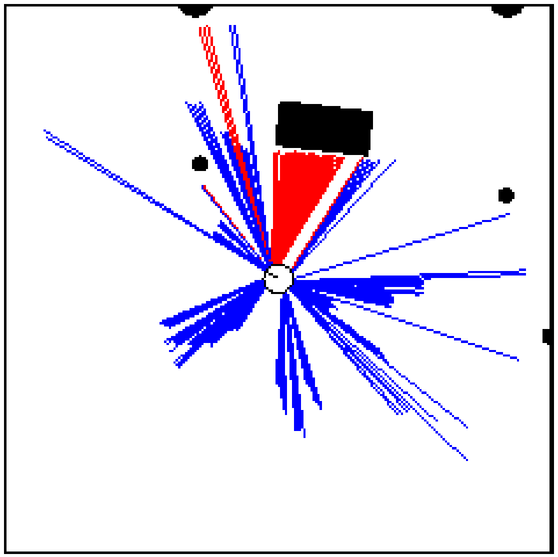
$$p(x \mid u, x') \neq p(x \mid u, x', m)$$

Approximation: $$p(x \mid u, x', m) = \eta \; p(x \mid m) \; p(x \mid u, x')$$

Sensors for Mobile Robots

- Contact sensors: Bumpers
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, tof, 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*.
- **Question**: Where do the probabilities come from?
- **Approach**: Let's try to explain a measurement.

Beam-based Sensor Model

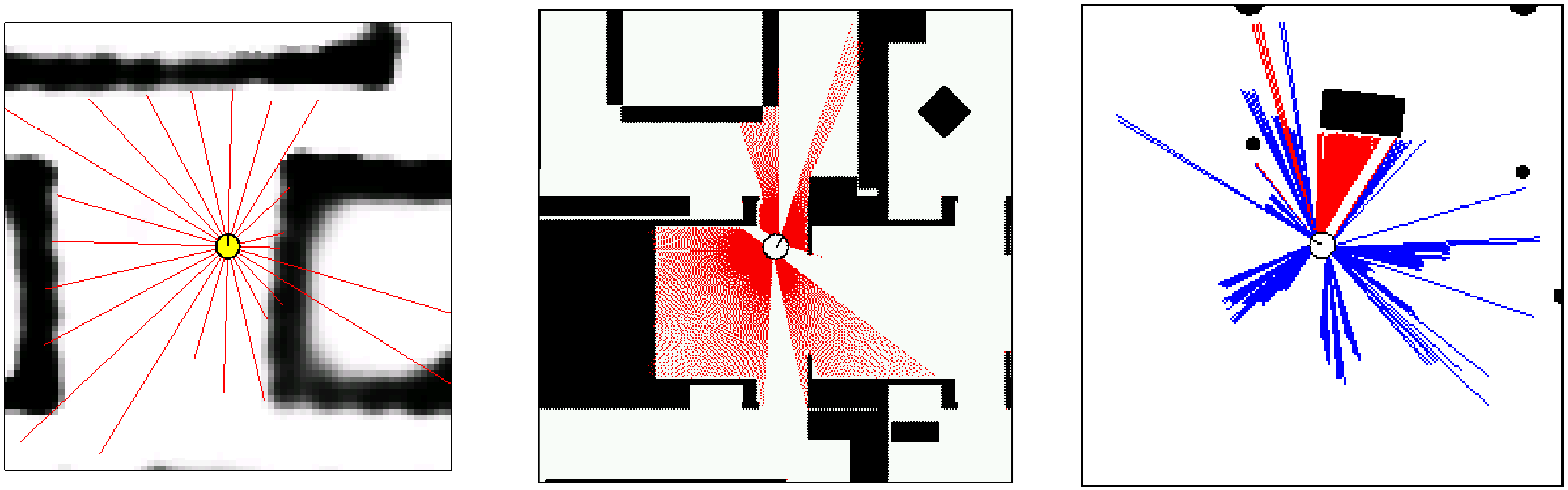
- Scan *z* consists of *K* measurements.

$$z = \{z_1, z_2, \ldots, z_K\}$$

- Individual measurements are independent given the robot position.

$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

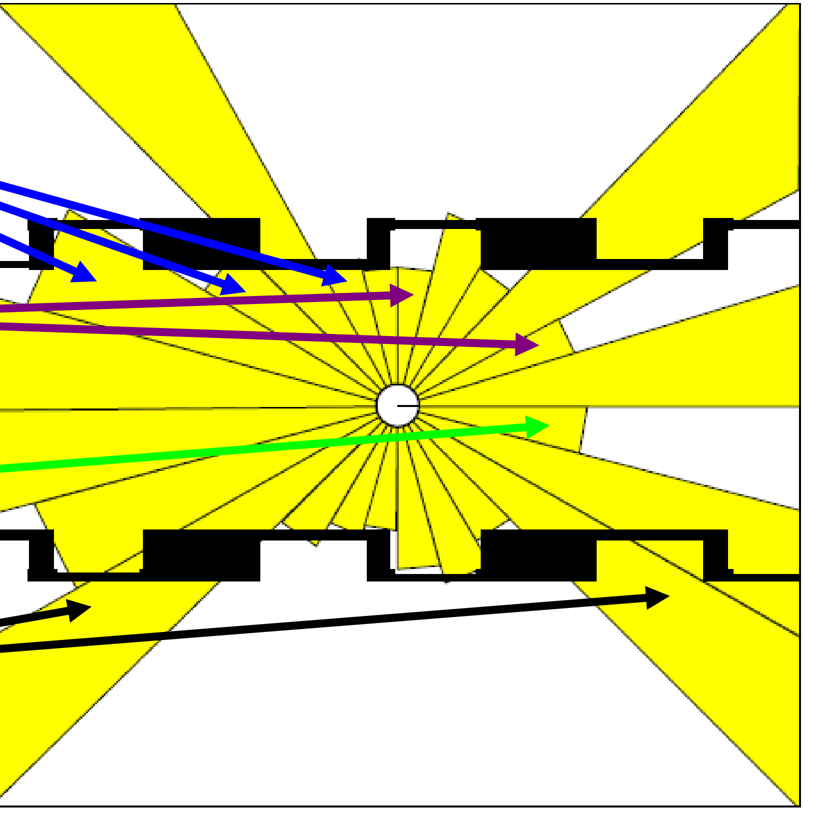
Beam-based Sensor Model



$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

Typical Measurement Errors of an Range Measurements

1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements

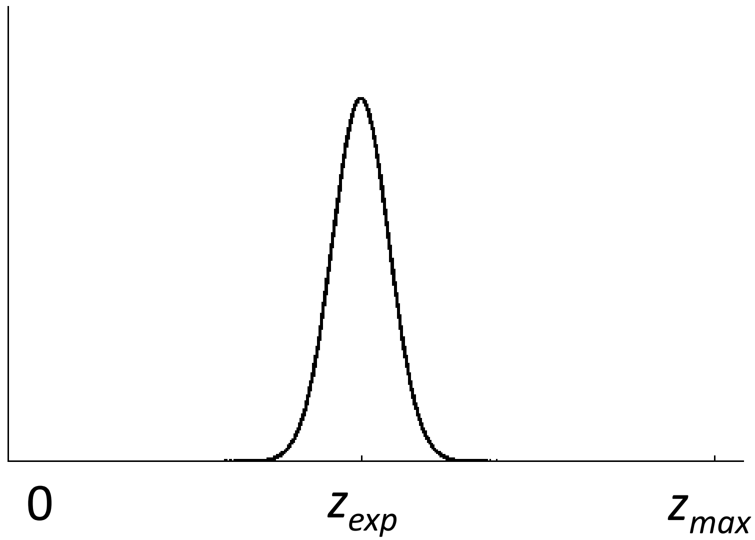


Proximity Measurement

- 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.

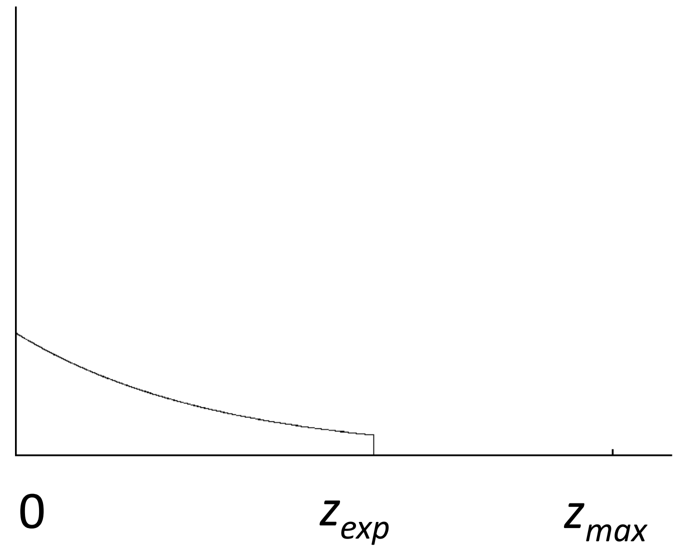
Beam-based Proximity Model

Measurement noise



$$P_{hit}(z \mid x, m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z-z_{\exp})^2}{b}}$$

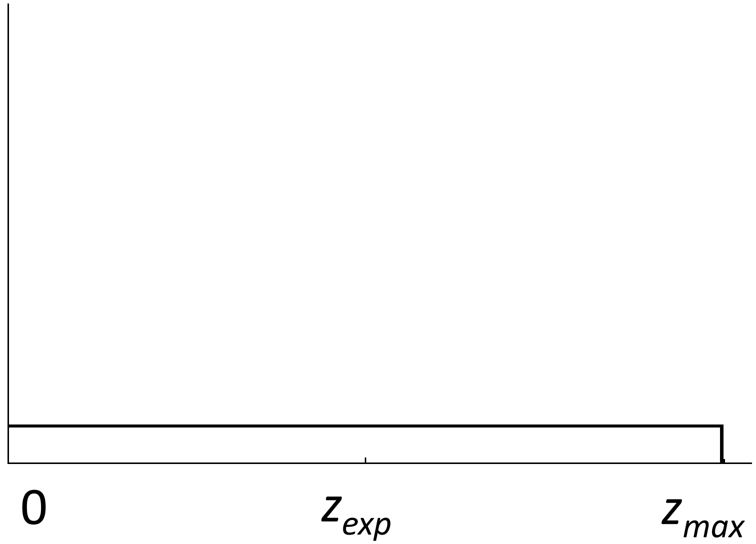
Unexpected obstacles



$$P_{\text{unexp}}(z \mid x, m) = \begin{cases} \eta\, \lambda\, e^{-\lambda z} & z < z_{\exp} \\ 0 & otherwise \end{cases}$$

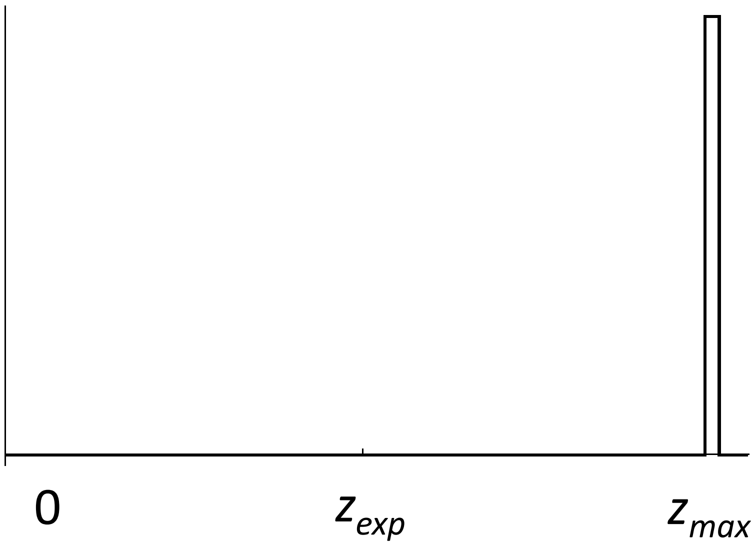
Beam-based Proximity Model

Random measurement



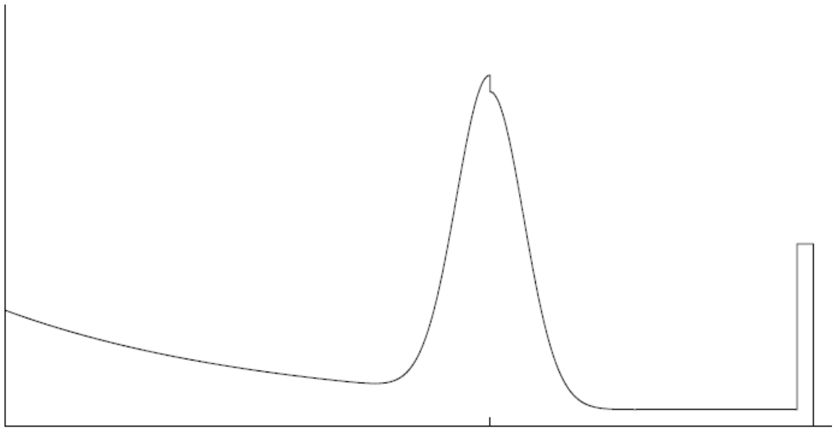
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Max range



$$P_{\max}(z \mid x, m) = \eta \frac{1}{z_{small}}$$

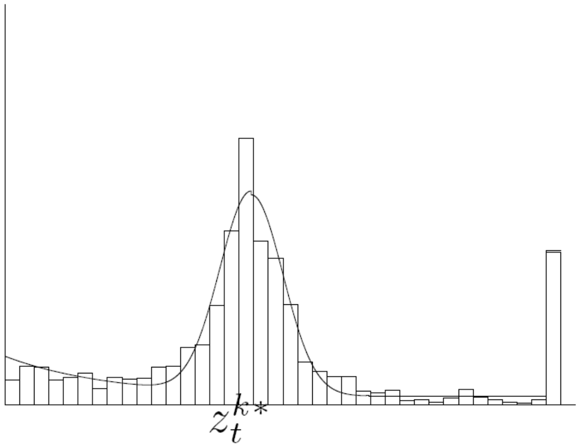
Resulting Mixture Density



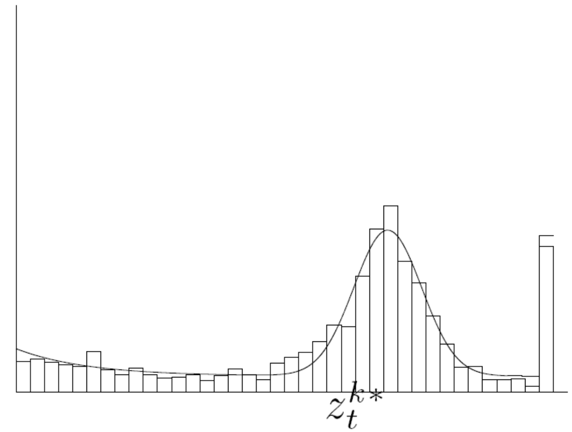
$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x, m) \\ P_{\text{unexp}}(z \mid x, m) \\ P_{\text{max}}(z \mid x, m) \\ P_{\text{rand}}(z \mid x, m) \end{pmatrix}$$

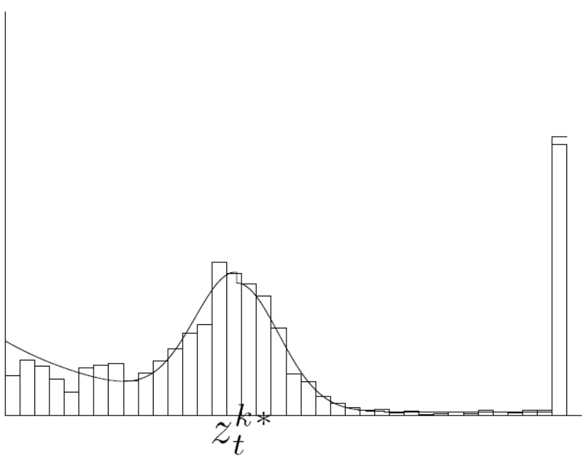
How can we determine the model parameters?

Approximation Results

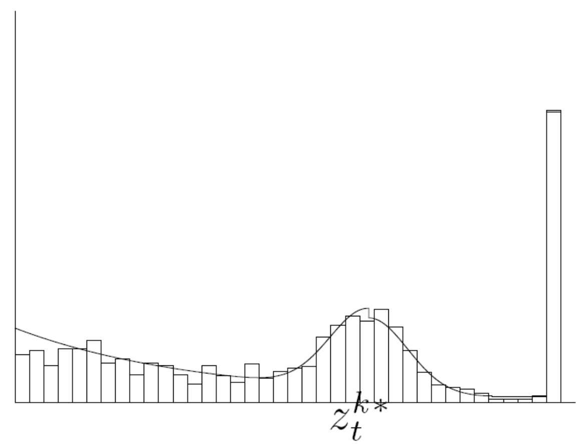


Laser





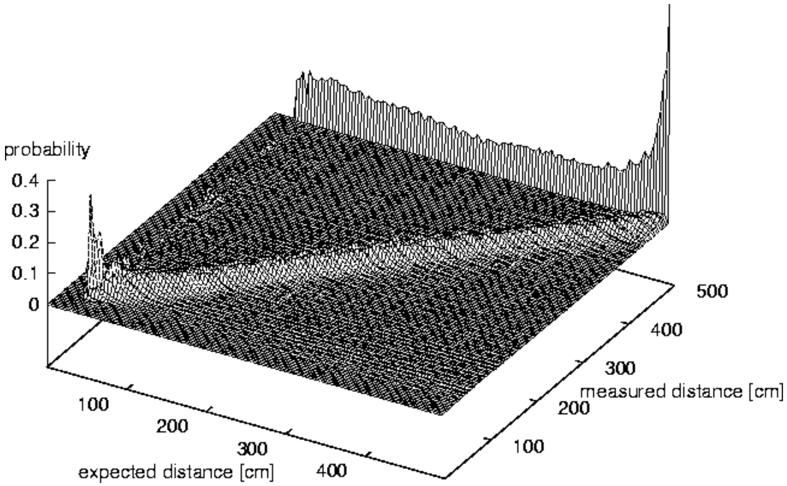
Sonar



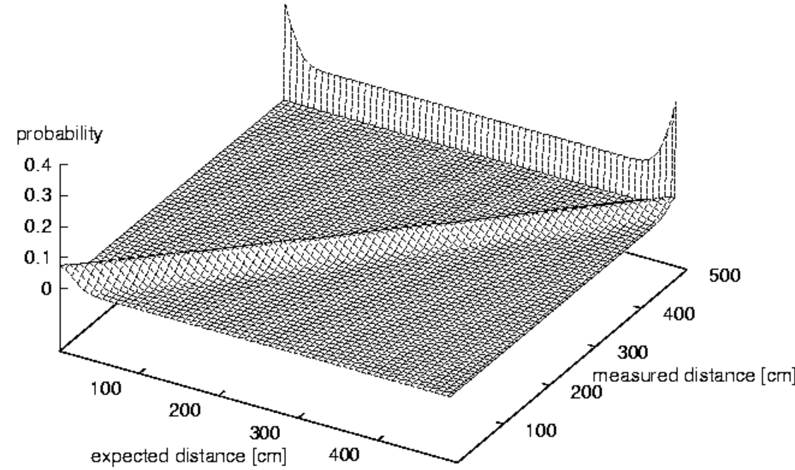
300cm

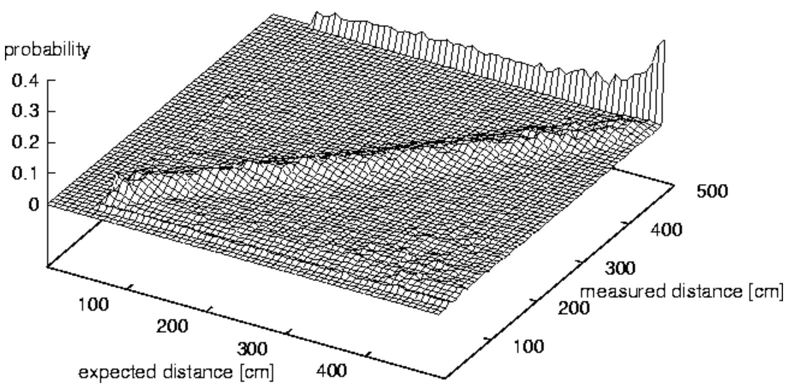
400cm

Approximation Results

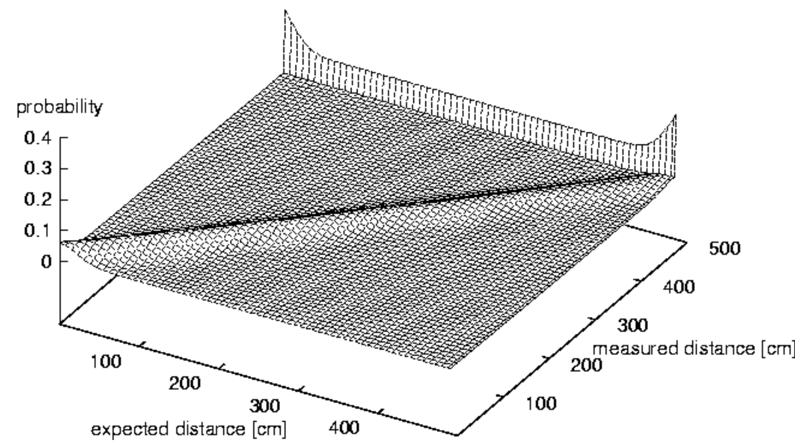


Laser



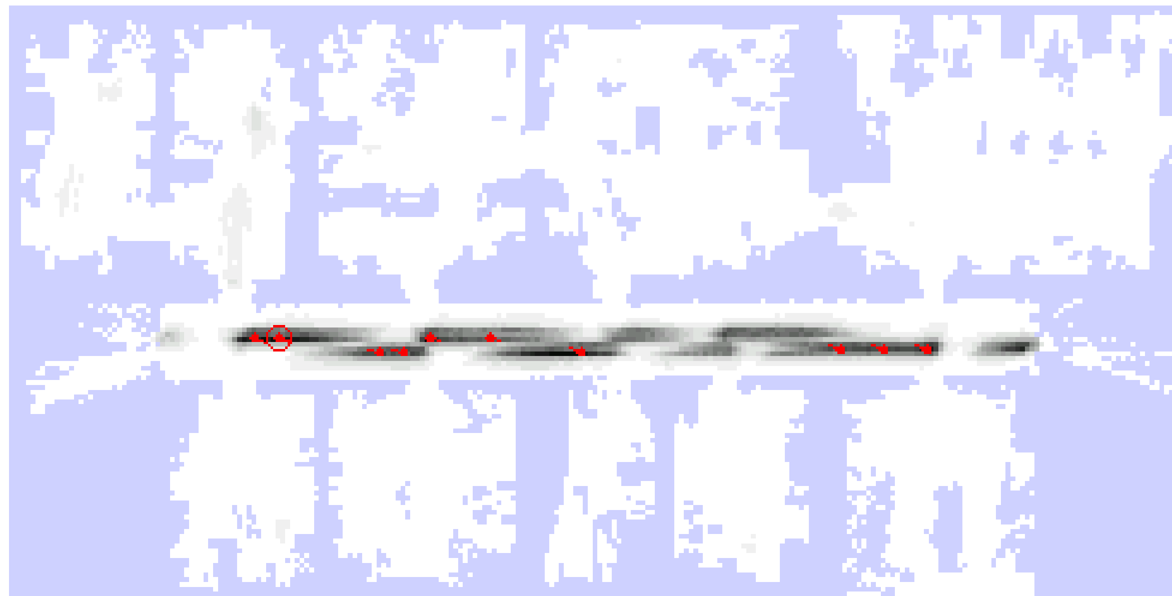


Sonar



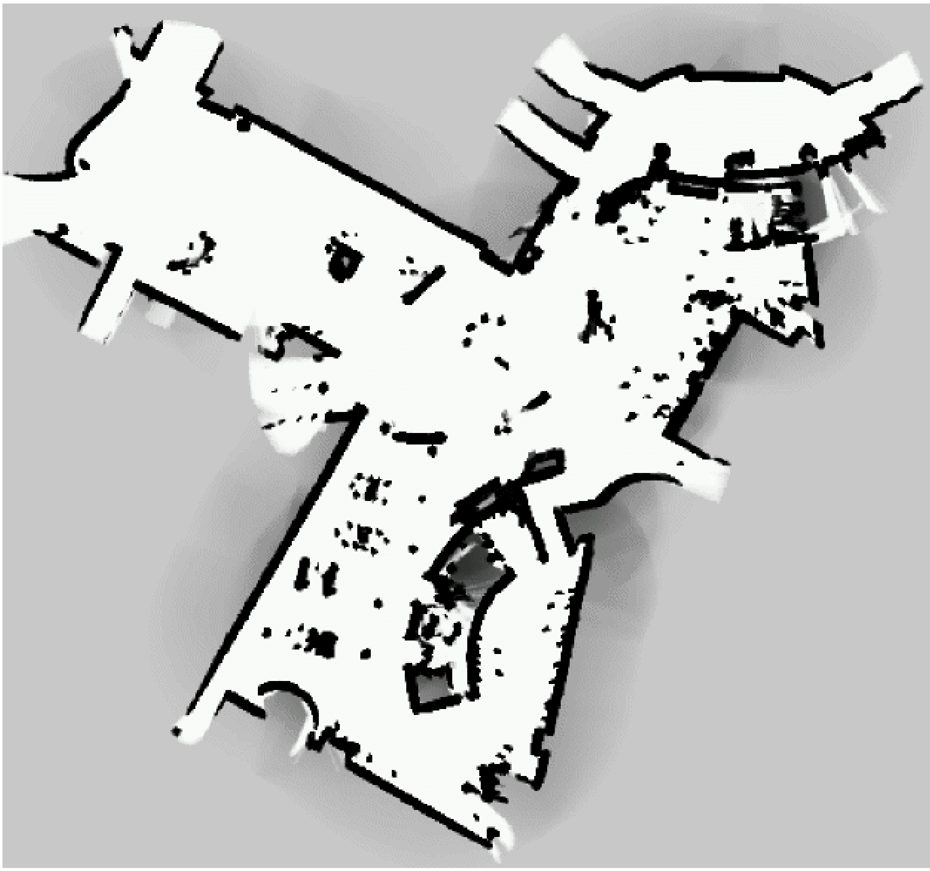
Example

z

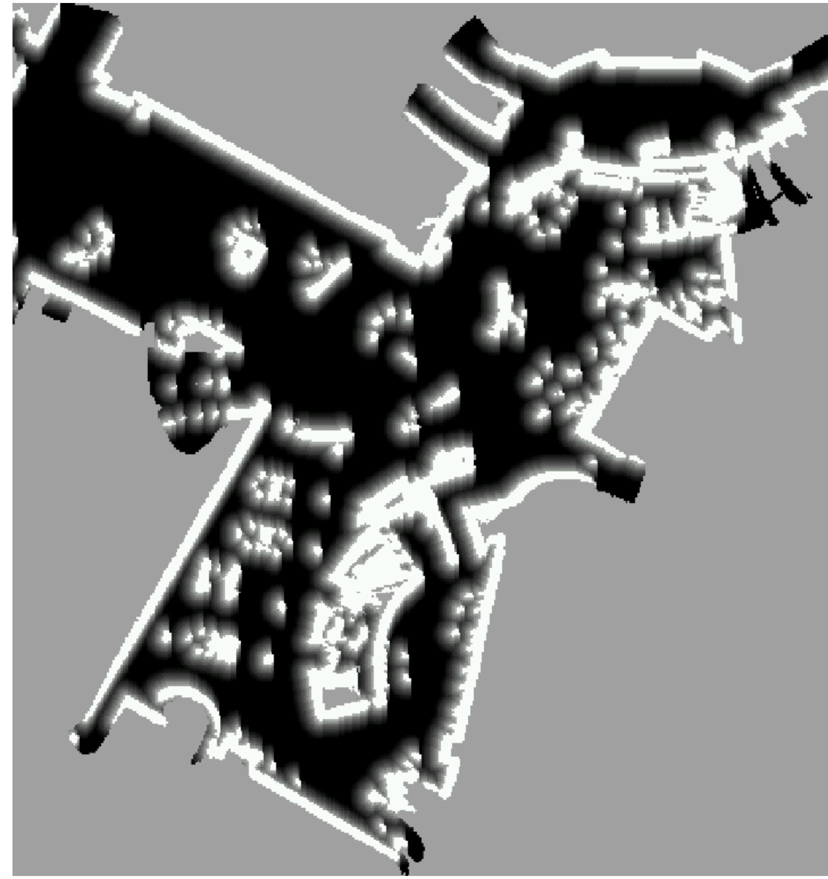


$P(z|x,m)$

San Jose Tech Museum



Occupancy grid map



Likelihood field

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.

Distance and Bearing

Probabilistic Model

$$z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$$

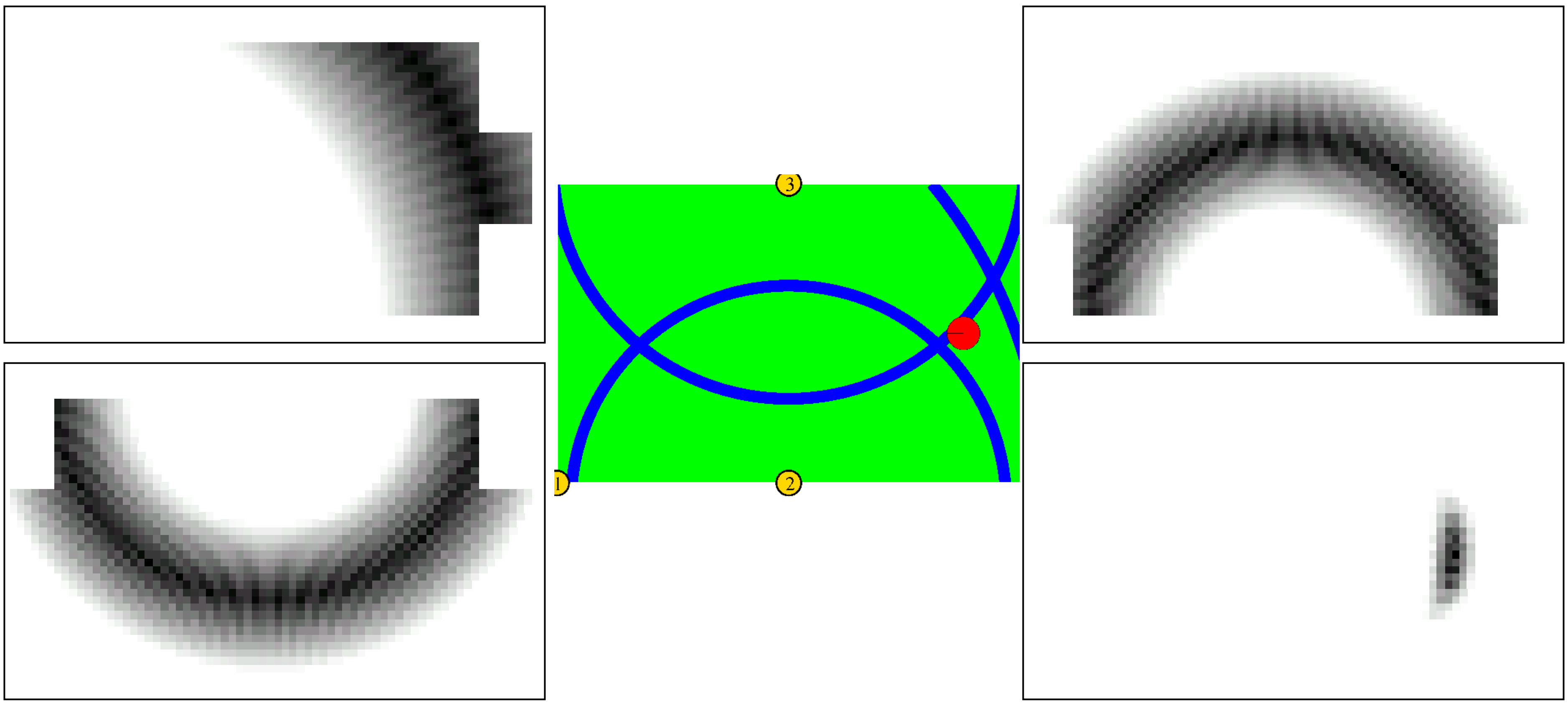
$$\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$$

$$\hat{a} = \mathrm{atan2}(m_y(i) - y, m_x(i) - x) - \theta$$

$$p_{\mathrm{det}} = \mathrm{prob}(\hat{d} - d, \varepsilon_d) \cdot \mathrm{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$$

$$z_{\mathrm{det}} p_{\mathrm{det}} + z_{\mathrm{fp}} P_{\mathrm{uniform}}(z \mid x, m)$$

Distributions



Laser Scanner Features (Line)

$$\alpha = \frac{1}{2}\boldsymbol{atan2}\left(-2\sum_{i=0}^{n}(\bar{y}-y_i)(\bar{x}-x_i)\ ,\sum_{i=0}^{n}(\bar{y}-y_i)^2-(\bar{x}-x_i)^2\right)$$

$$r = \bar{x}\cos(\alpha) + \bar{y}\sin(\alpha)$$

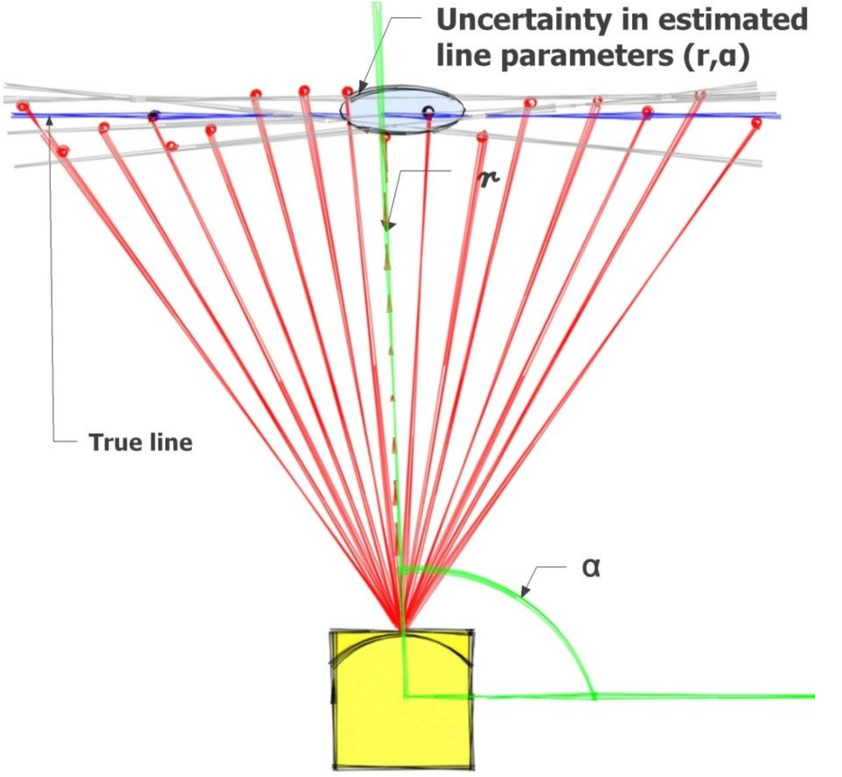
$$P_{\alpha r} = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_r^2 \end{bmatrix}$$

$$\sigma_\alpha^2 = \sum_{i=0}^{n}\left[\frac{\partial\alpha}{\partial\rho_i}\right]^2\sigma_{\rho_i}^2$$

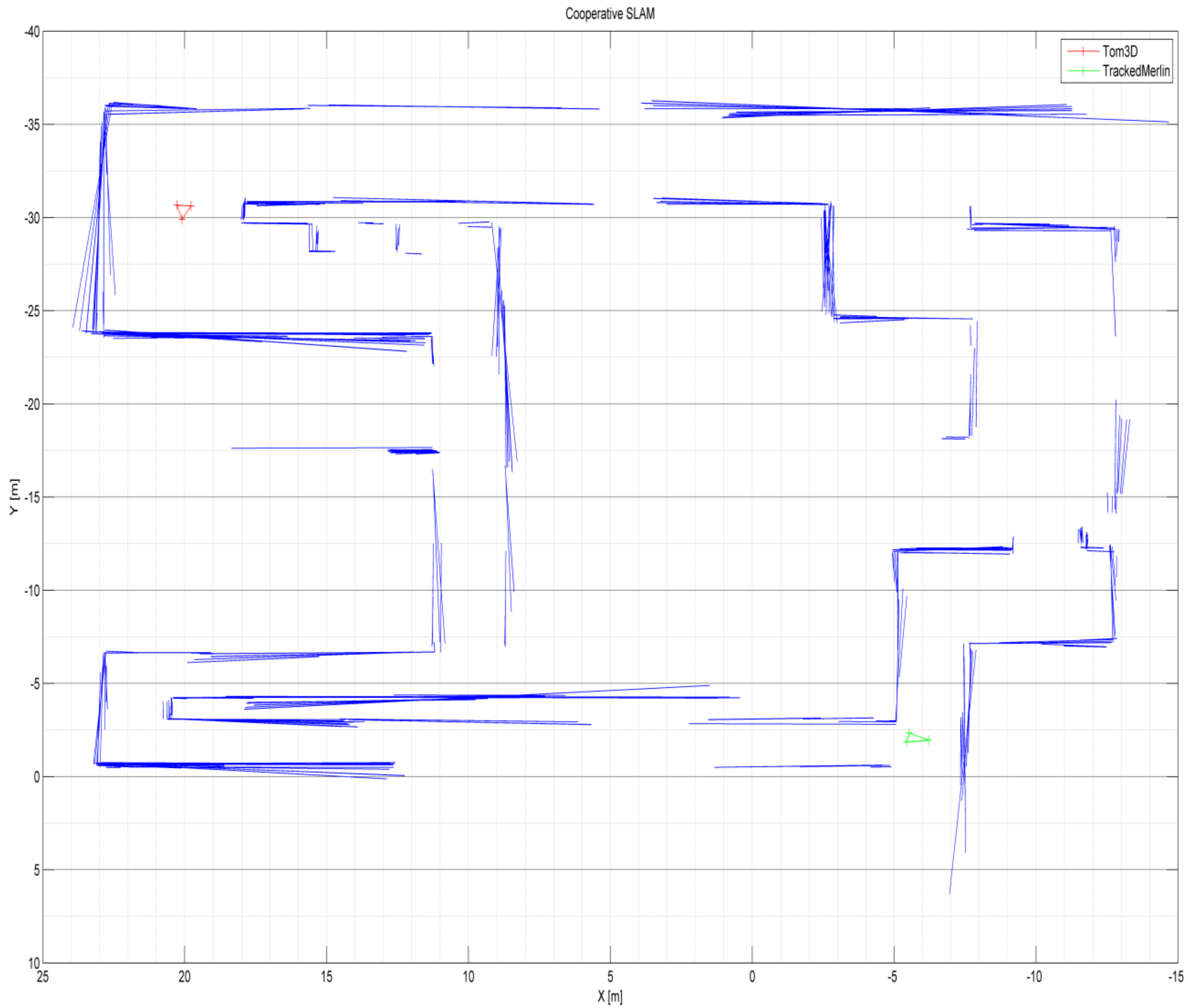
$$\sigma_r^2 = \sum_{i=0}^{n}\left[\frac{\partial r}{\partial\rho_i}\right]^2\sigma_{\rho_i}^2$$

$$\sigma_{\alpha r} = \sigma_{r\alpha} = \sum_{i=0}^{n}\left[\frac{\partial\alpha}{\partial\rho_i}\cdot\frac{\partial r}{\partial\rho_i}\right]\cdot\sigma_{\rho_i}^2$$

$$\rho\cos(\theta-\alpha) - r = 0$$

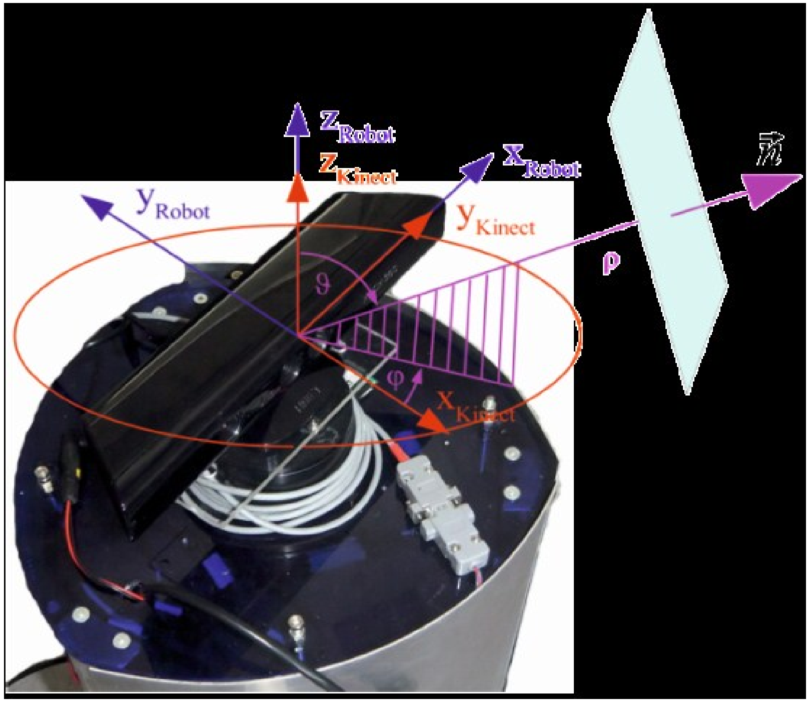


2D Geometric Feature based Map



Kinect Features (Plane)

$$p_x \cdot \sin(\theta) \cdot \cos(\varphi) + p_y \cdot \sin(\theta) \cdot \sin(\varphi) + p_z \cdot \cos(\theta) = \rho$$



Do until DetectedPlanes < 8 and TotalPoints > MinPoints

Randomly select (P_1, P_2, P_3) from a random circular region

Calculate plane from (P_1, P_2, P_3)

Transform the Calculate plane from $\mathbb{R}^3$ to Hough space

If local maxima is found in Hough space

Delete points corresponding to plane from input points

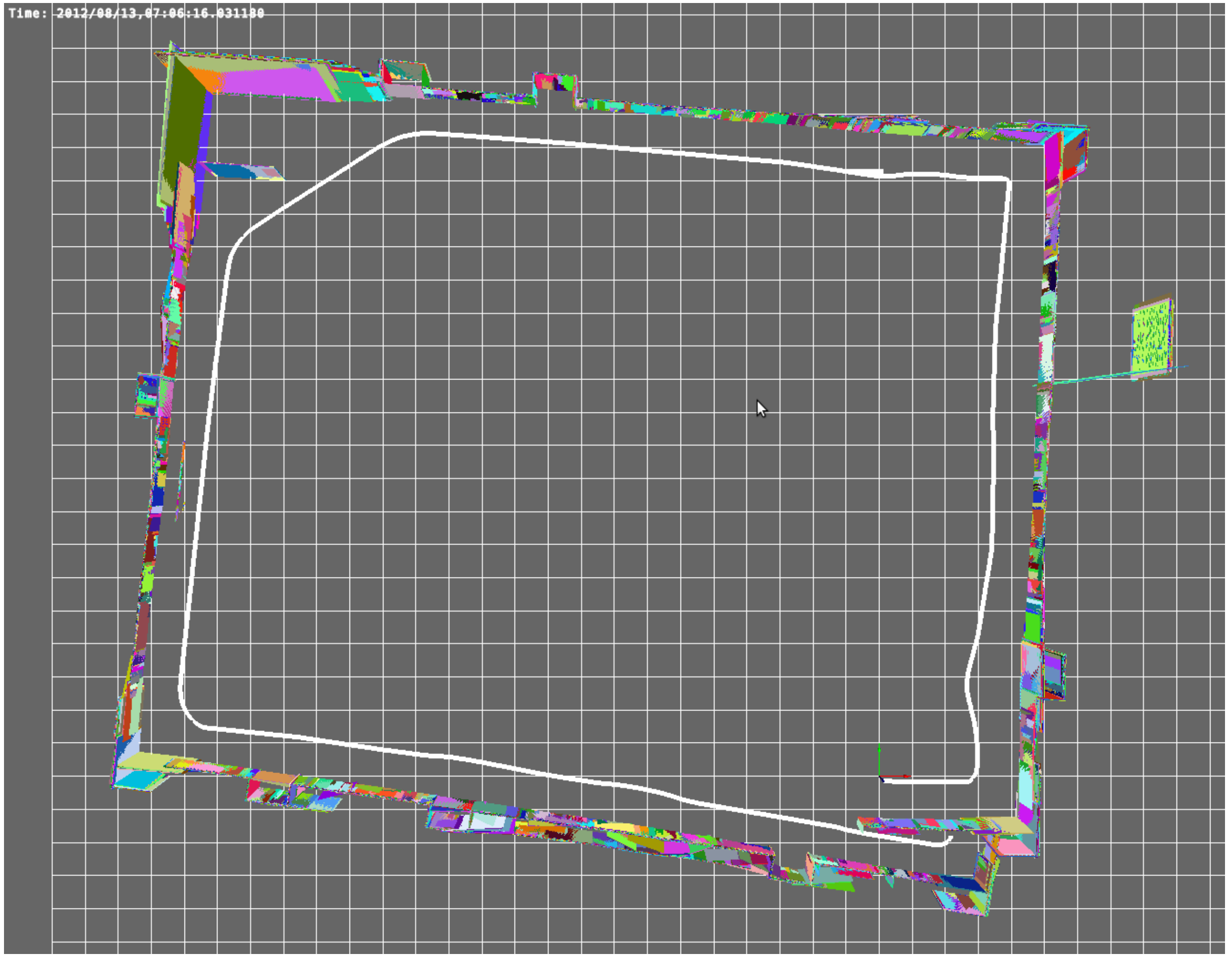
Calculate plane boundries

Reset Hough space

End if

End Do

3D Geometric Feature based Map



Summary of Sensor Models

- Explicitly modeling uncertainty in sensing is key to robustness.
- In many cases, good models can be found by the following approach:
  1. Determine parametric model of noise free measurement.
  2. Analyze sources of noise.
  3. Add adequate noise to parameters (eventually mix in densities for noise).
  4. Learn (and verify) parameters by fitting model to data.
  5. 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!

Summary

- Motion Models
  - Velocity based model (Dead-Reckoning)
  - Odometry based model (Wheel Encoders)
- Sensor Models
  - Beam model of range finders
  - Feature based sensor models
    - Camera
    - Laser scanner
    - Kinect

Questions