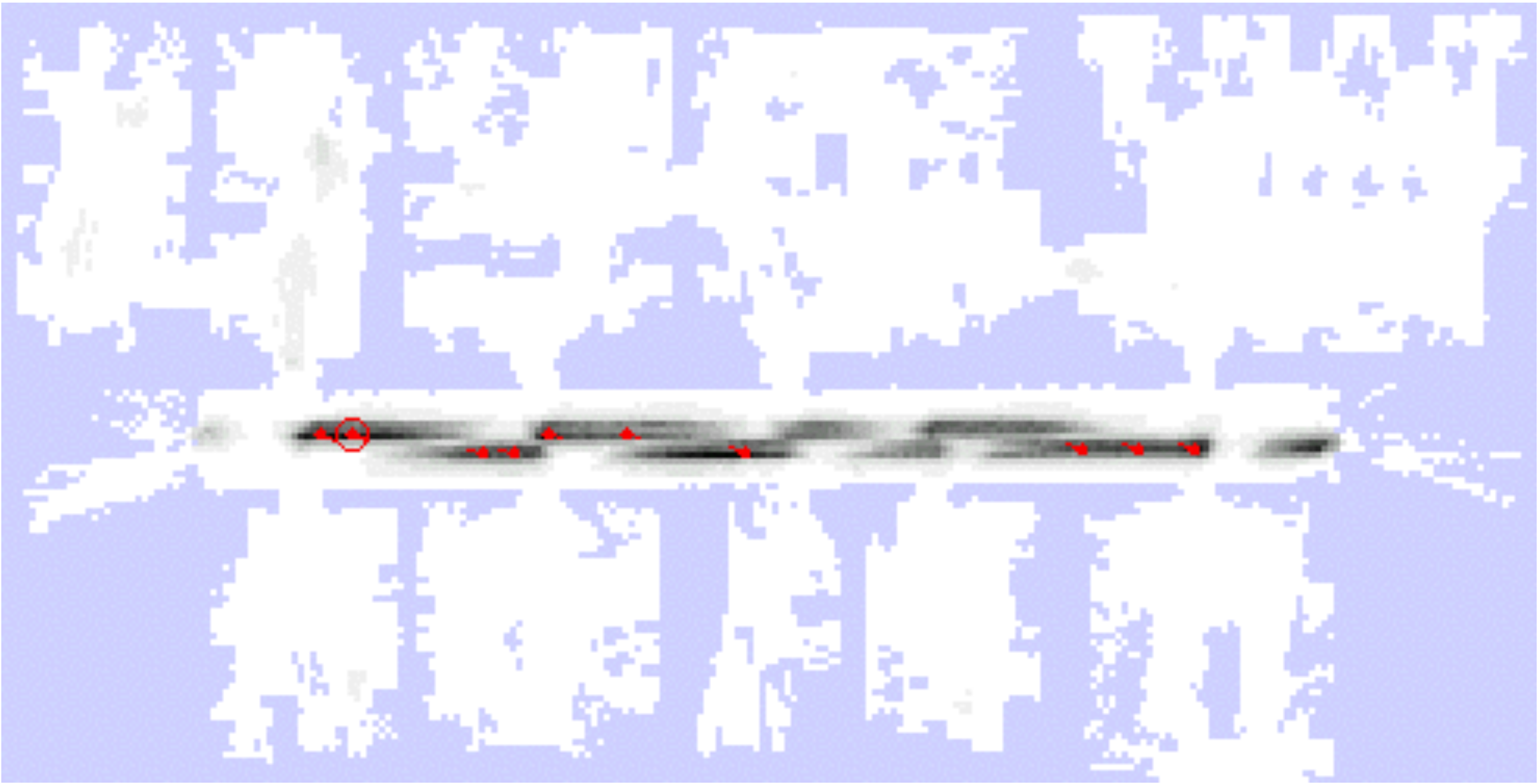
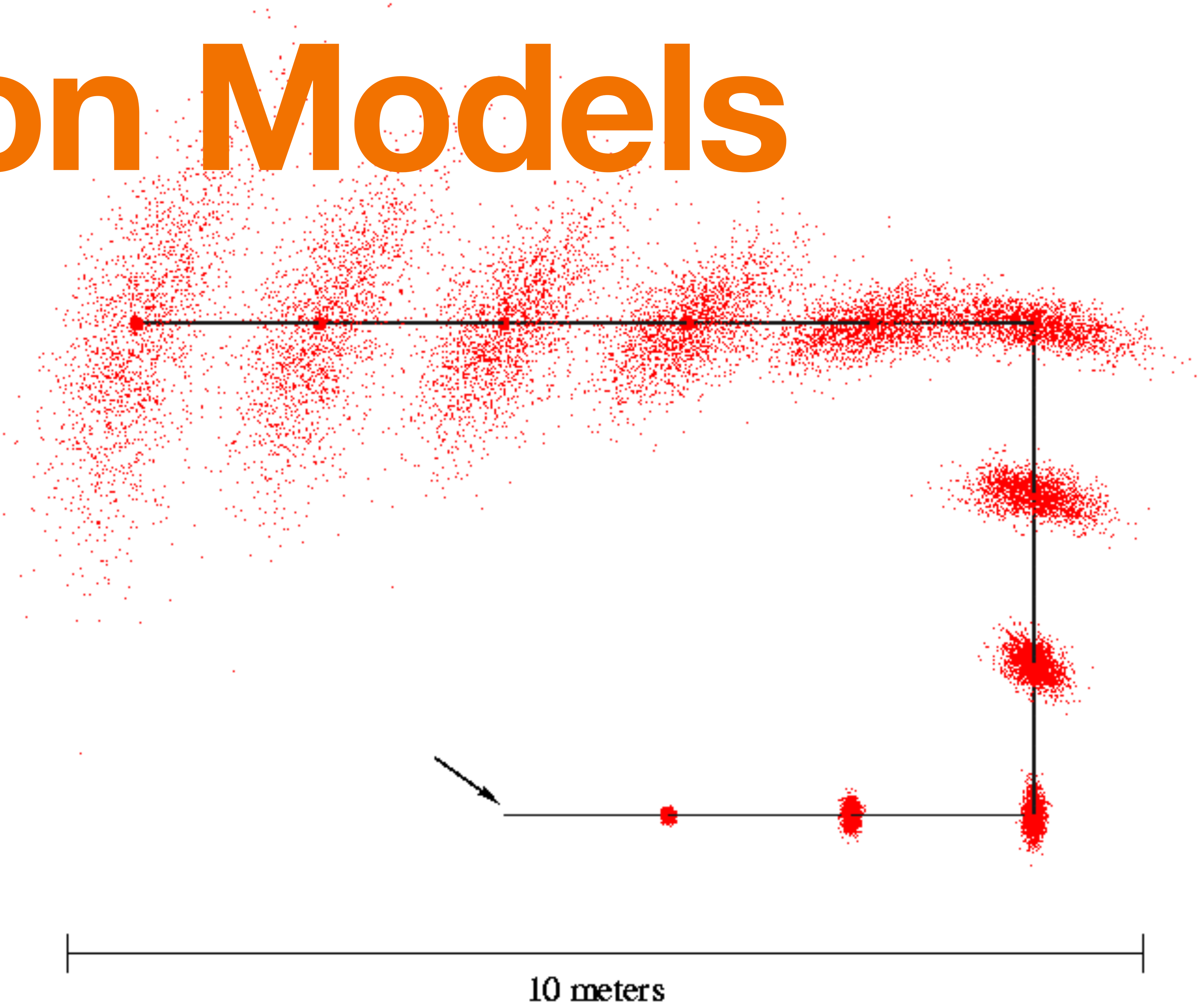
Lecture 17

Mobile Robotics - II - Sensor and Motion Models





from Probabilistic Robotics

Course logistics

- Quiz 8 was due today at noon.
- Project 5 was posted on 02/28 and is due on 03/20 (today).
- Project 6 will be posted on 03/20 and will be on 03/27.
- Group formation for P7 and Final Project by 03/20.
  - How is that going?

Previously

Joint and Conditional Probability

- $P(X = x \text{ and } Y = y) = P(x, y)$
- $P(x \mid y)$ is the probability of x given y

$$P(x \mid y) = \frac{P(x, y)}{P(y)}$$

$$P(x, y) = P(x \mid y) P(y)$$

- If X and Y are independent then

$$P(x, y) = P(x) P(y)$$

- If X and Y are independent then

$$P(x \mid y) = P(x)$$

Bayes Formula

$$P(x, y) = P(x \mid y) P(y) = P(y \mid x) P(x)$$

$$P(x \mid y) = \frac{P(y \mid x) P(x)}{P(y)} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Conditioning

- Bayes rule and background knowledge:

$$P(x \mid y, z) = \frac{P(y \mid x, z) P(x \mid z)}{P(y \mid z)}$$

$$P(x \mid y) = \int P(x \mid y, z) P(z \mid y) dz$$

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots z_n) = \frac{P(z_n \mid x, z_1, \ldots z_{n-1}) P(x \mid z_1, \ldots z_{n-1})}{P(z_n \mid z_1, \ldots z_{n-1})}$$

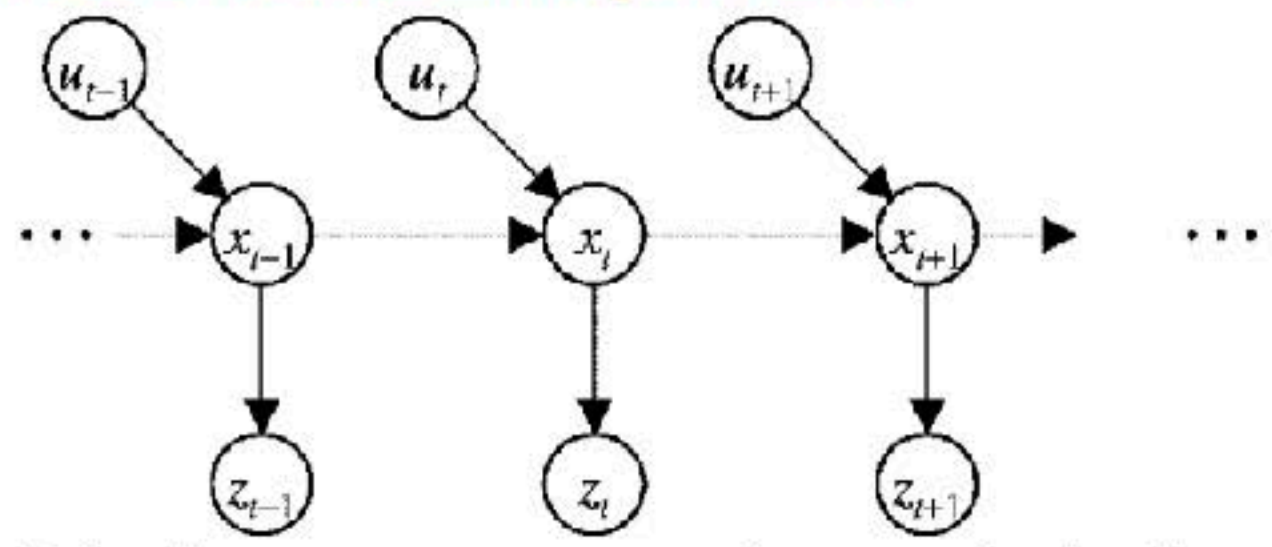
Markov assumption: z_n is conditionally independent of $z_1, \ldots, z_{n-1}$ given x.

$$P(x \mid z_1, \ldots z_n) = \frac{P(z_n \mid x) P(x \mid z_1, \ldots z_{n-1})}{P(z_n \mid z_1, \ldots z_{n-1})}$$

$$= \eta P(z_n \mid x) P(x \mid z_1, \ldots z_{n-1})$$

$$= \eta_{1 \ldots n} \prod_{i=1 \ldots n} P(z_i \mid x) P(x)$$

Markov Assumption



$$P(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

$$P(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Bayes Filters

z = observation
u = action
x = state

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

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

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

Probabilistic Motion Models

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

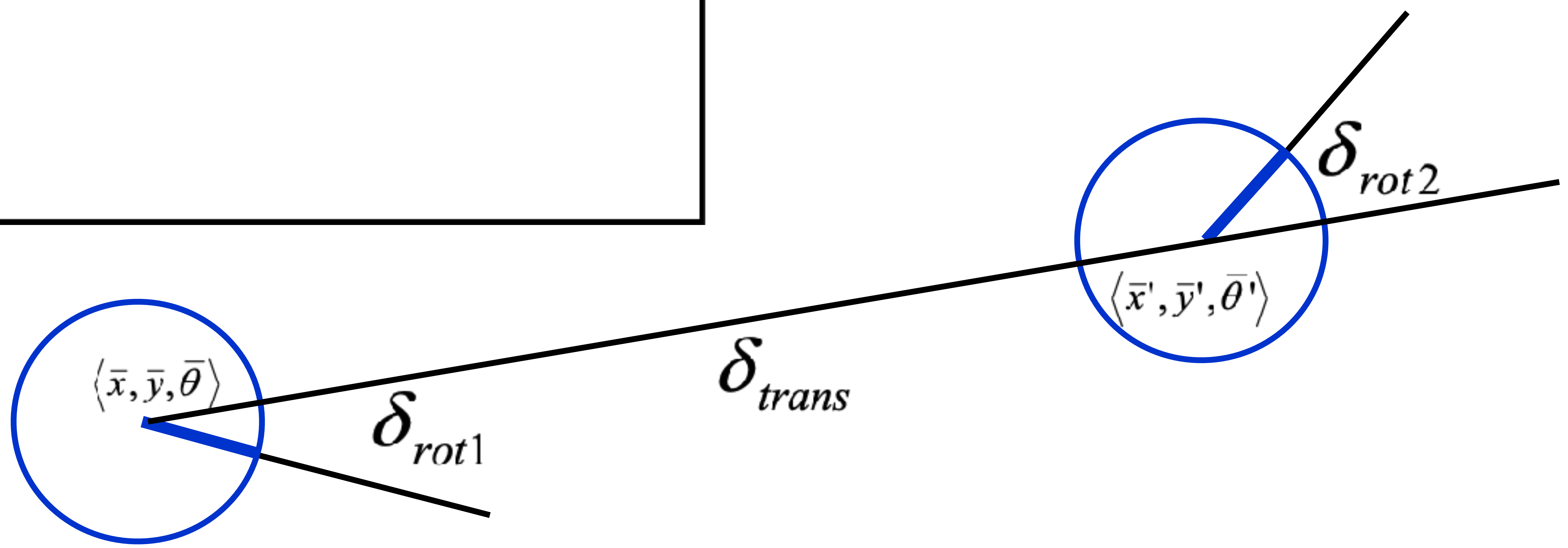
Probabilistic Kinematics

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

$$\delta_{trans} =$$

$$\delta_{rot1} =$$

$$\delta_{rot2} =$$



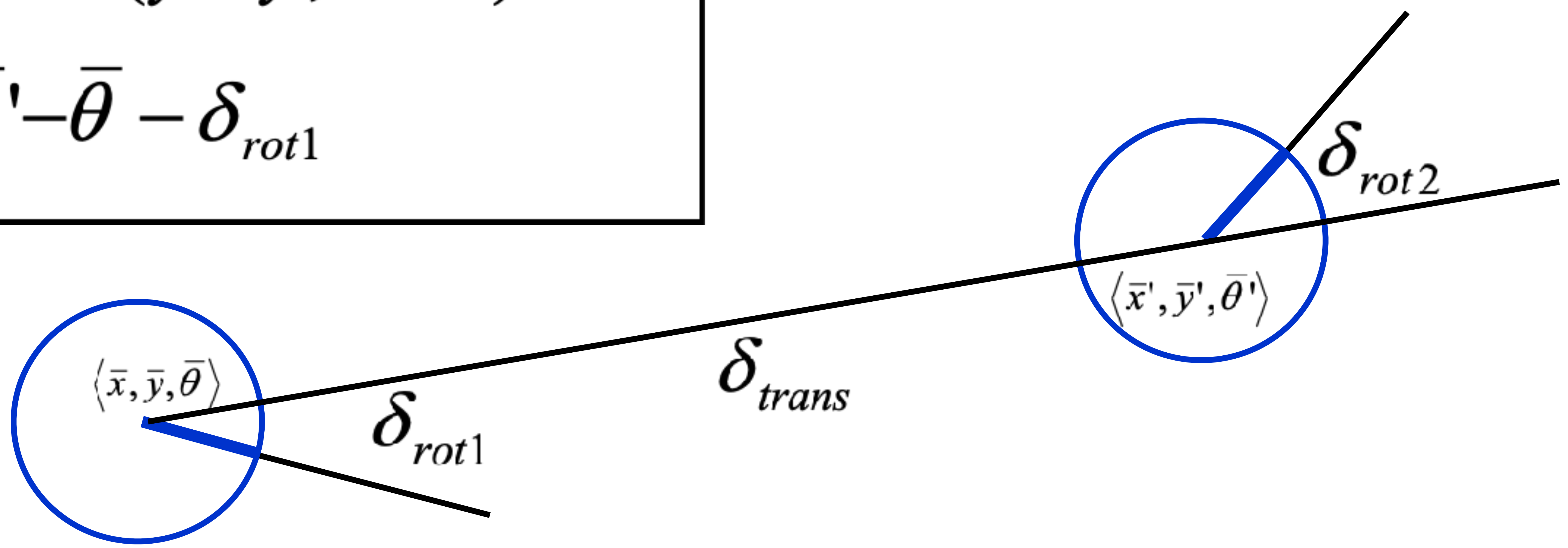
Probabilistic Kinematics

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

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

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

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

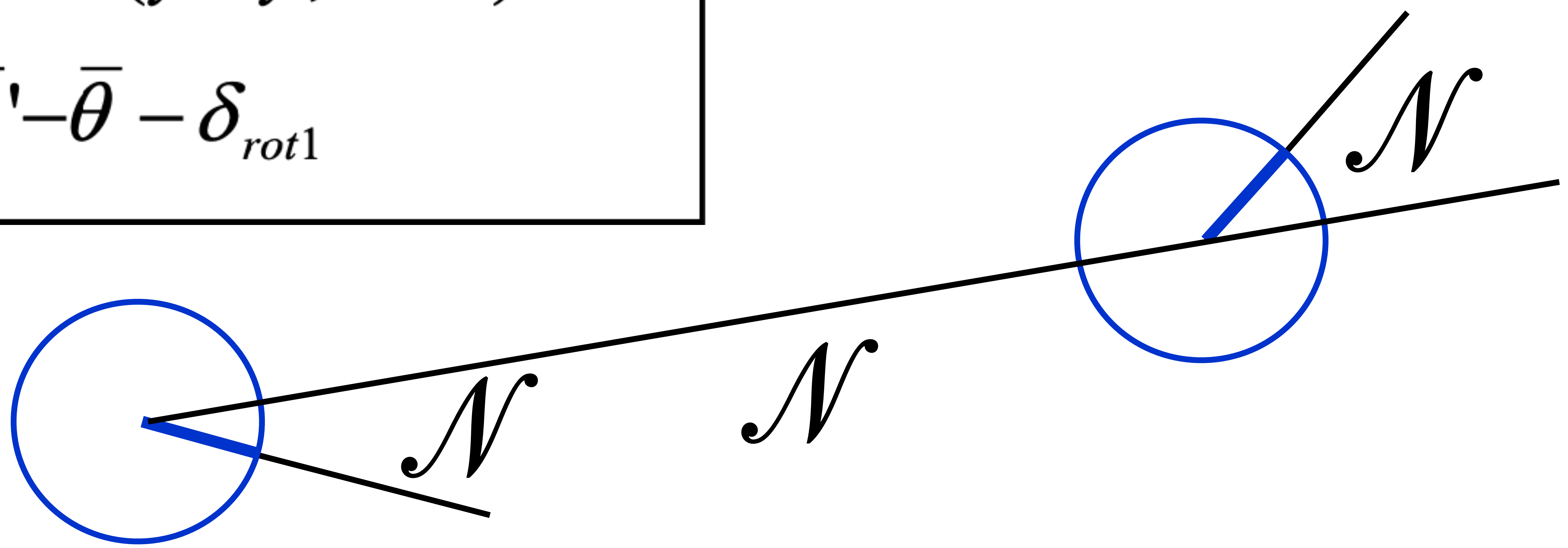


Noise Model for Motion

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

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

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



Noise Model for Motion

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

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

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

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

Odometry Motion Model

$$p(x_t \mid x_{t-1}, u_t)$$

Odometry Motion Model

$$p(x_t \mid x_{t-1}, u_t)$$

Algorithm **motion_model_odometry** (u, x, x'):

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

Odometry Motion Model

$$p(x_t \mid x_{t-1}, u_t)$$

Algorithm **motion_model_odometry** (u, x, x'):

1. $\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$
2. $\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
3. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$

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

4. $\hat{\delta}_{trans} = \sqrt{(x - x')^2 + (y - y')^2}$
5. $\hat{\delta}_{rot1} = \text{atan2}(y' - y, x' - x) - \theta$
6. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$

$$x = < x, y, \theta >$$
$$x' = < x', y', \theta' >$$

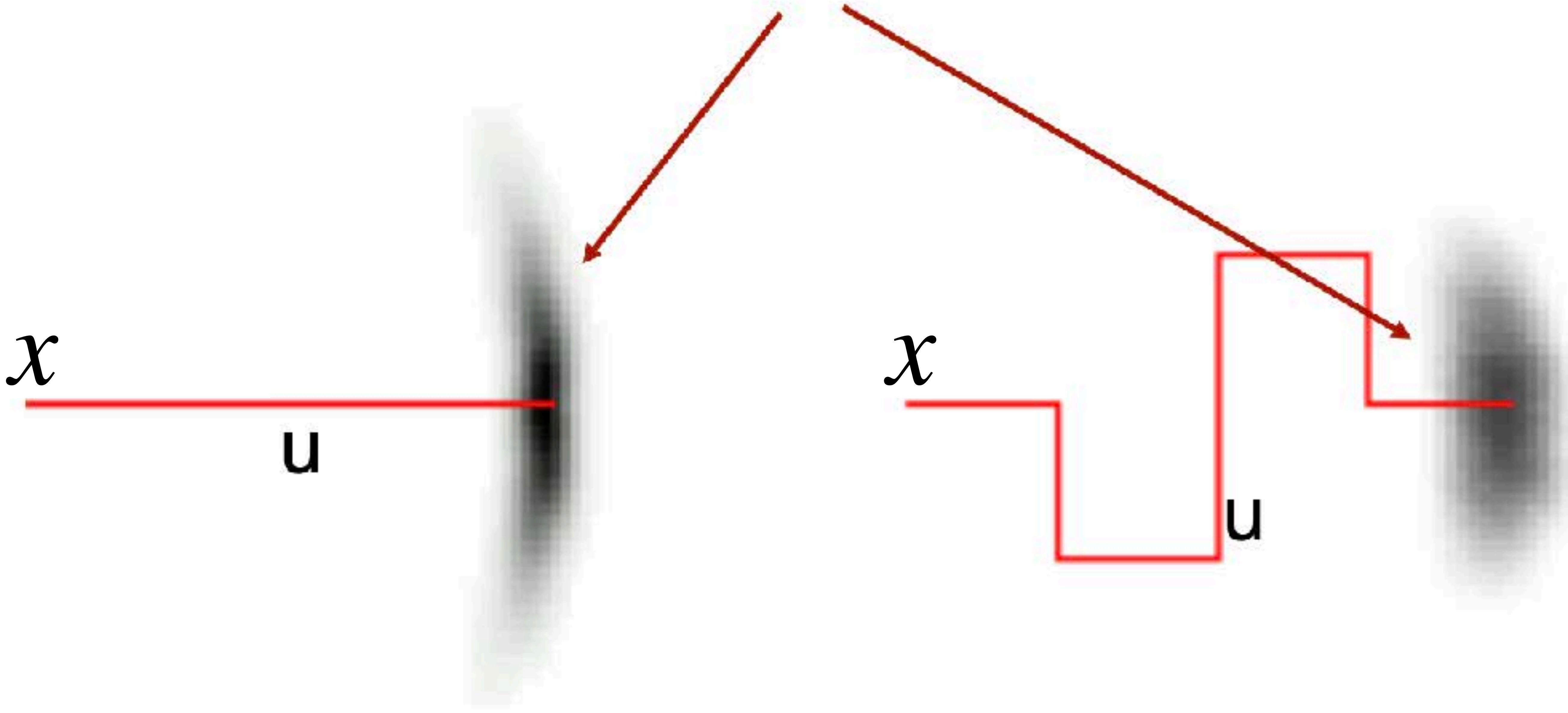
Finding the posterior

7. $p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 \hat{\delta}^2_{rot1} + \alpha_2 \hat{\delta}^2_{trans})$
8. $p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}^2_{trans} + \alpha_4 (\hat{\delta}^2_{rot1} + \hat{\delta}^2_{rot2}))$
9. $p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 \hat{\delta}^2_{rot2} + \alpha_2 \hat{\delta}^2_{trans})$
10. Return $p_1 * p_2 * p_3$

Odometry Motion Model

$$p(x_t \mid x_{t-1}, u_t)$$

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



This is a projected illustration ignoring the θ

Sample Odometry Motion Model

Sample for x_t

$$p(x_t \mid x_{t-1}, u_t)$$

Sample Odometry Motion Model

Sample for x_t

$$p(x_t \mid x_{t-1}, u_t)$$

1. Algorithm **sample_motion_model** (u, x):

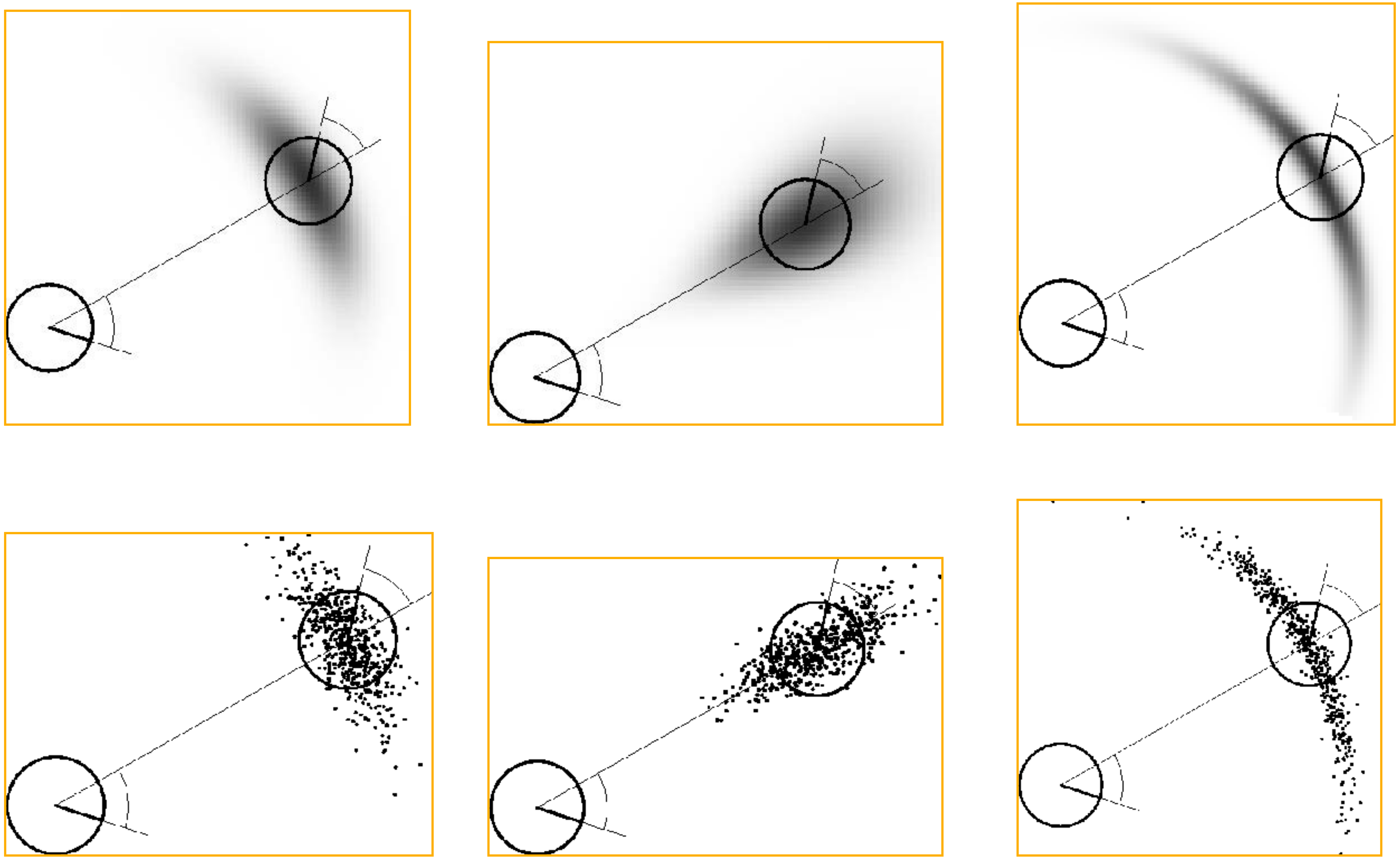
   $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$

1. $\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 |\delta_{rot1}| + \alpha_2\, \delta_{trans})$
2. $\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3\, \delta_{trans} + \alpha_4\, (|\delta_{rot1}| + |\delta_{rot2}|))$
3. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2\, \delta_{trans})$

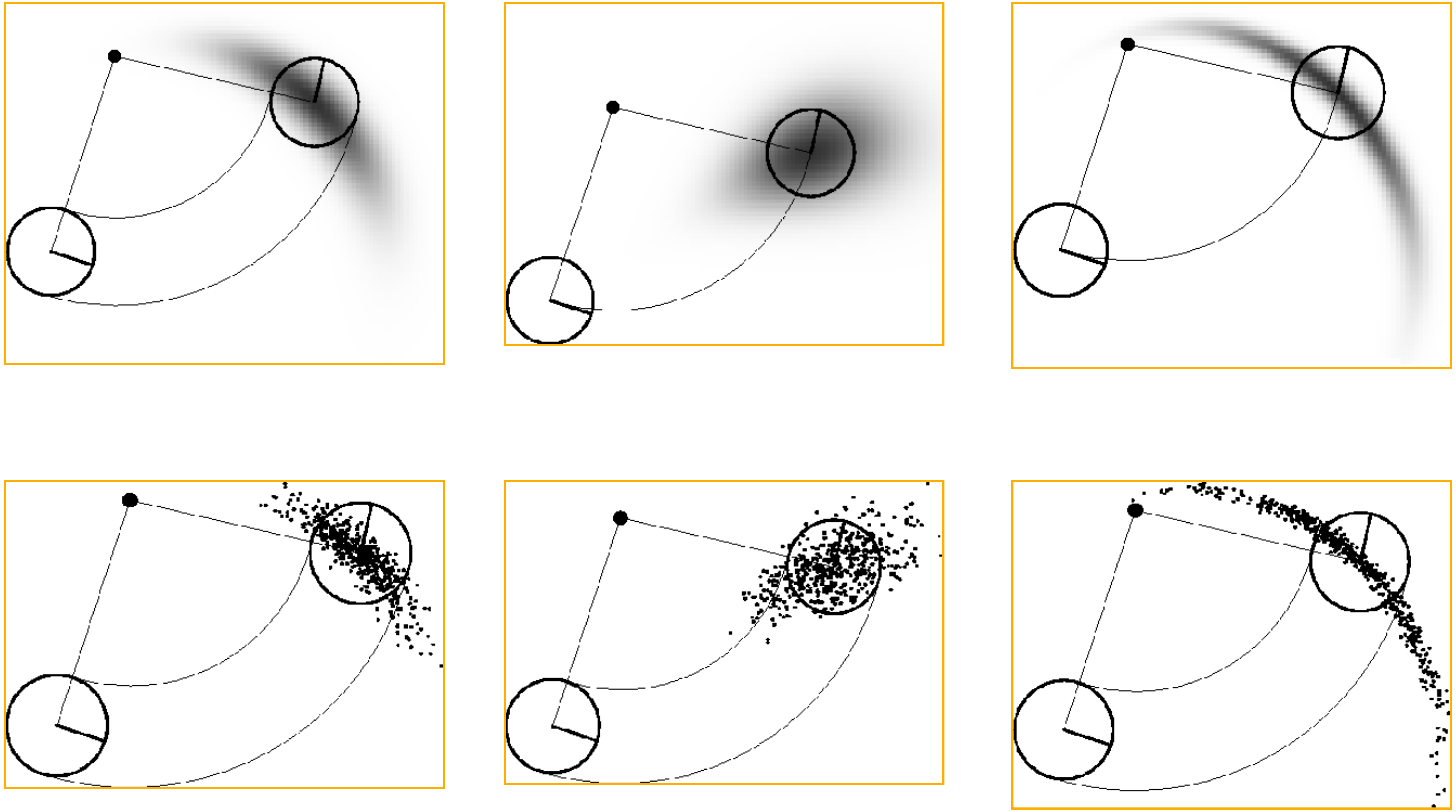
4. $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
5. $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$
6. $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

7. Return $\langle x', y', \theta' \rangle$

Examples (odometry based)

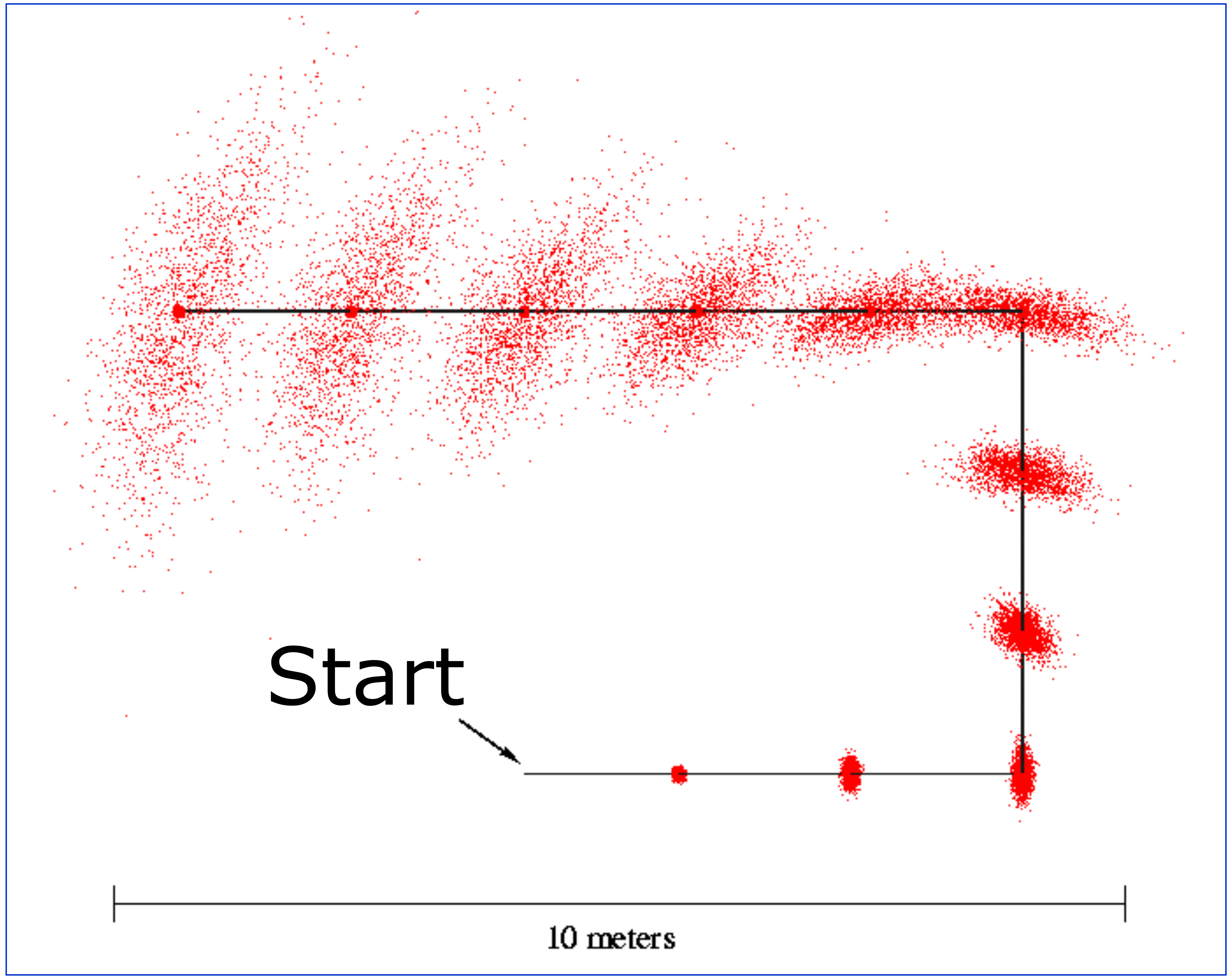


Examples (velocity based)



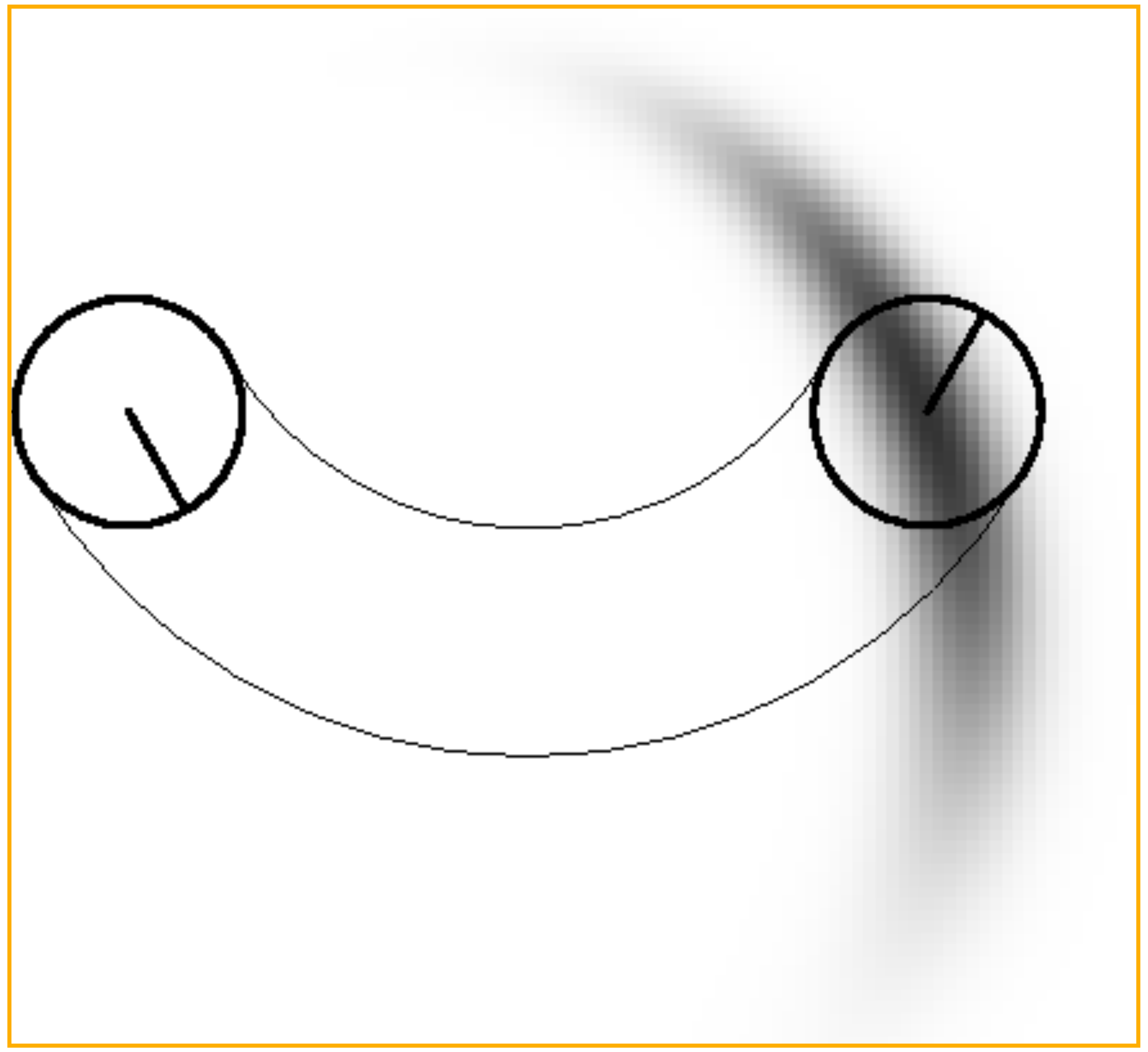
Slide borrowed from Dieter Fox

Sample-based Motion

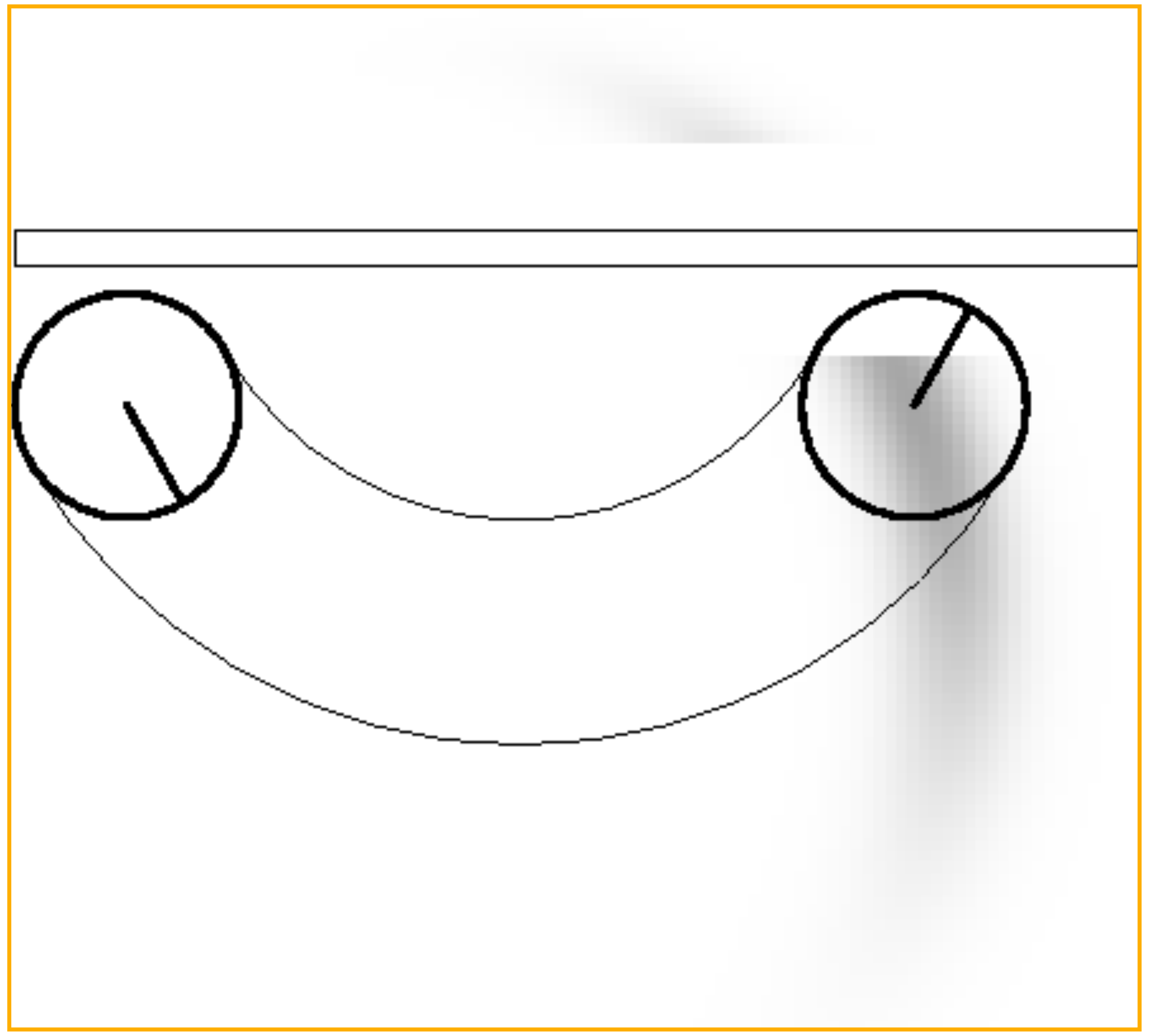


Slide borrowed from Dieter Fox

Motion Model with Map



$$P(x \mid u, x')$$



$$P(x \mid u, x', m) \approx P(x \mid m)\, P(x \mid u, x')$$

- When does this approximation fail?

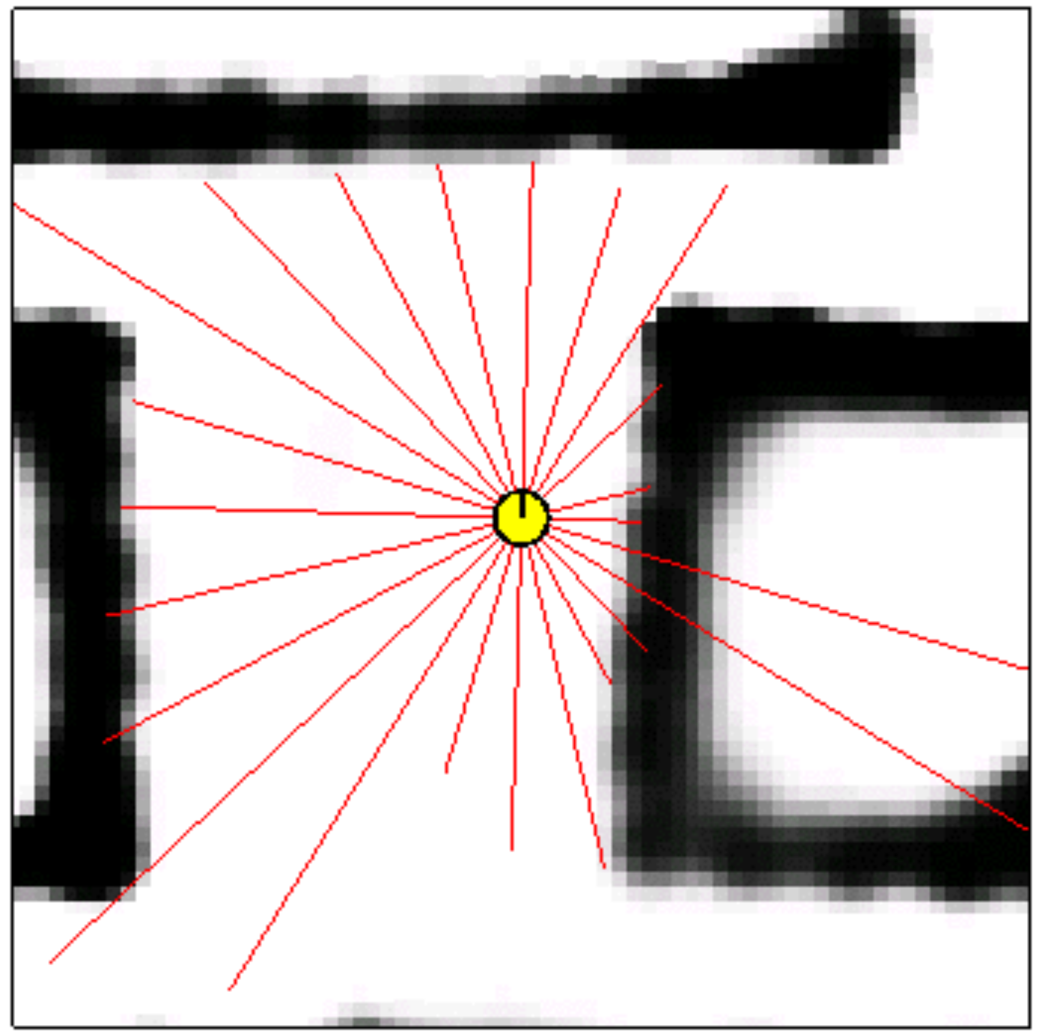
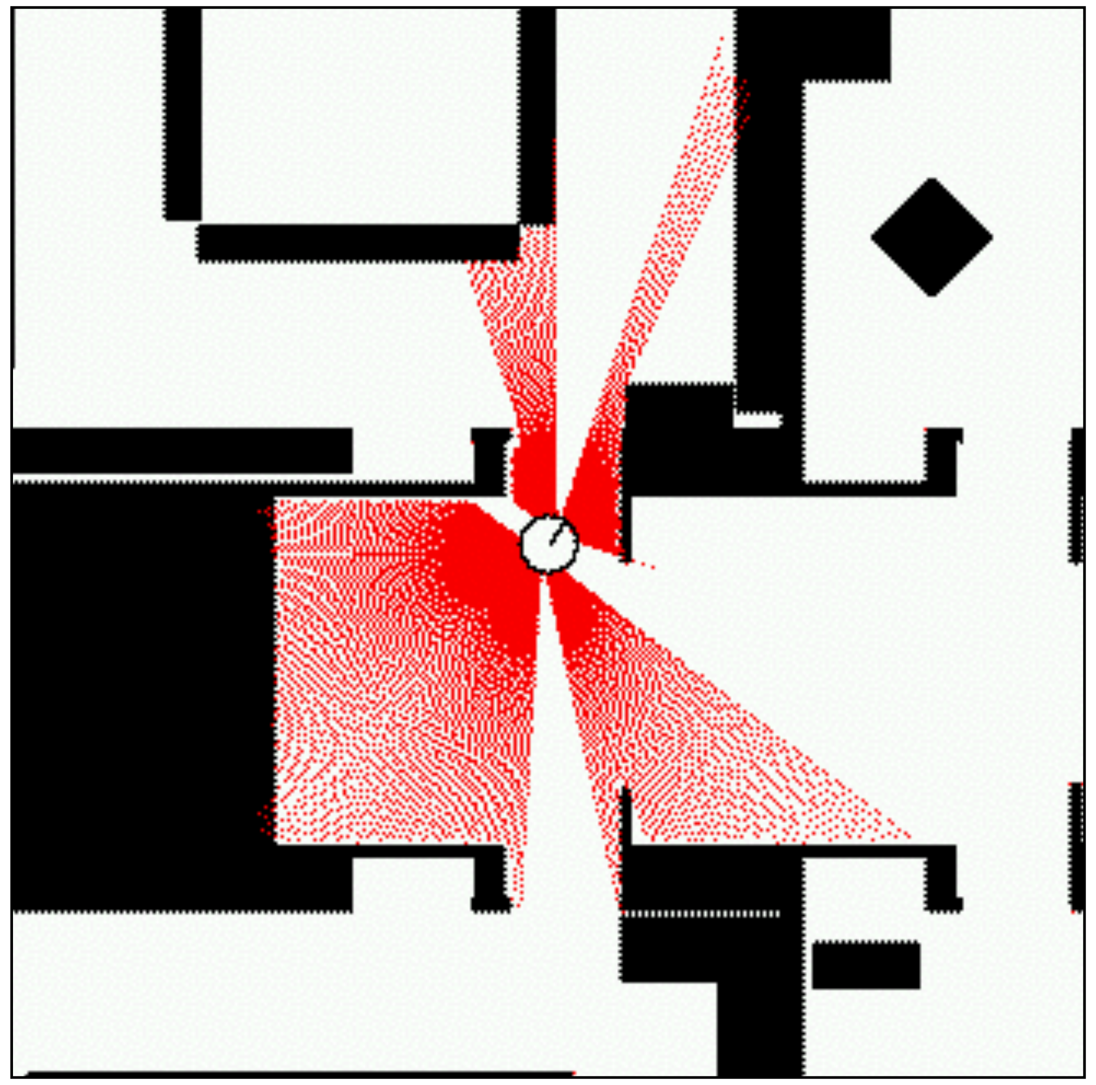
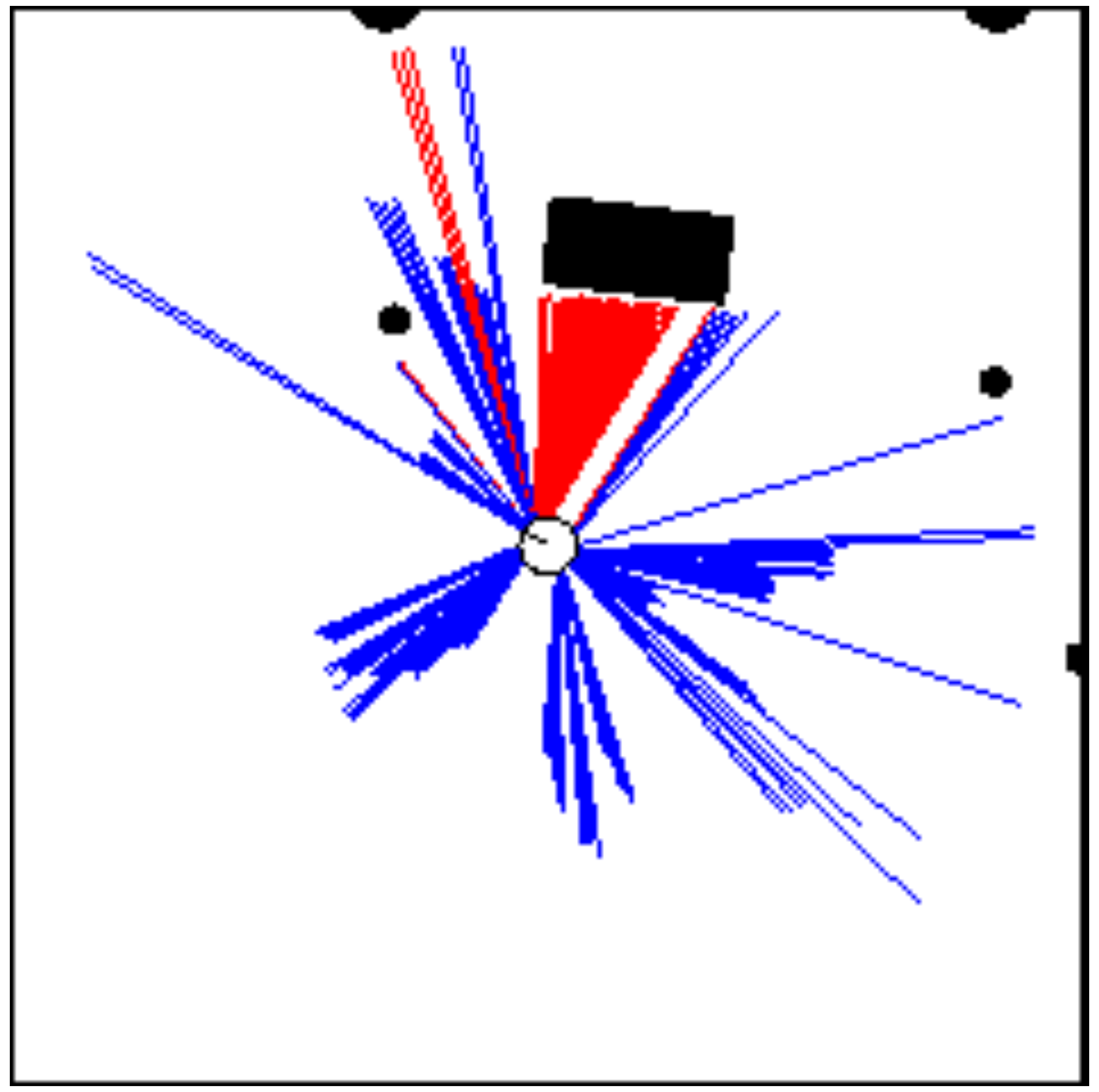
Probabilistic Sensor Models

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

Sensors for Mobile Robots

- Contact sensors: Bumpers, touch sensors
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
  - Encoders, torque
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, tof, phase)
  - Infrared (intensity)
- Visual sensors: Cameras, depth cameras
- Satellite-style sensors: GPS, MoCap

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.

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

Beam-based Sensor Model

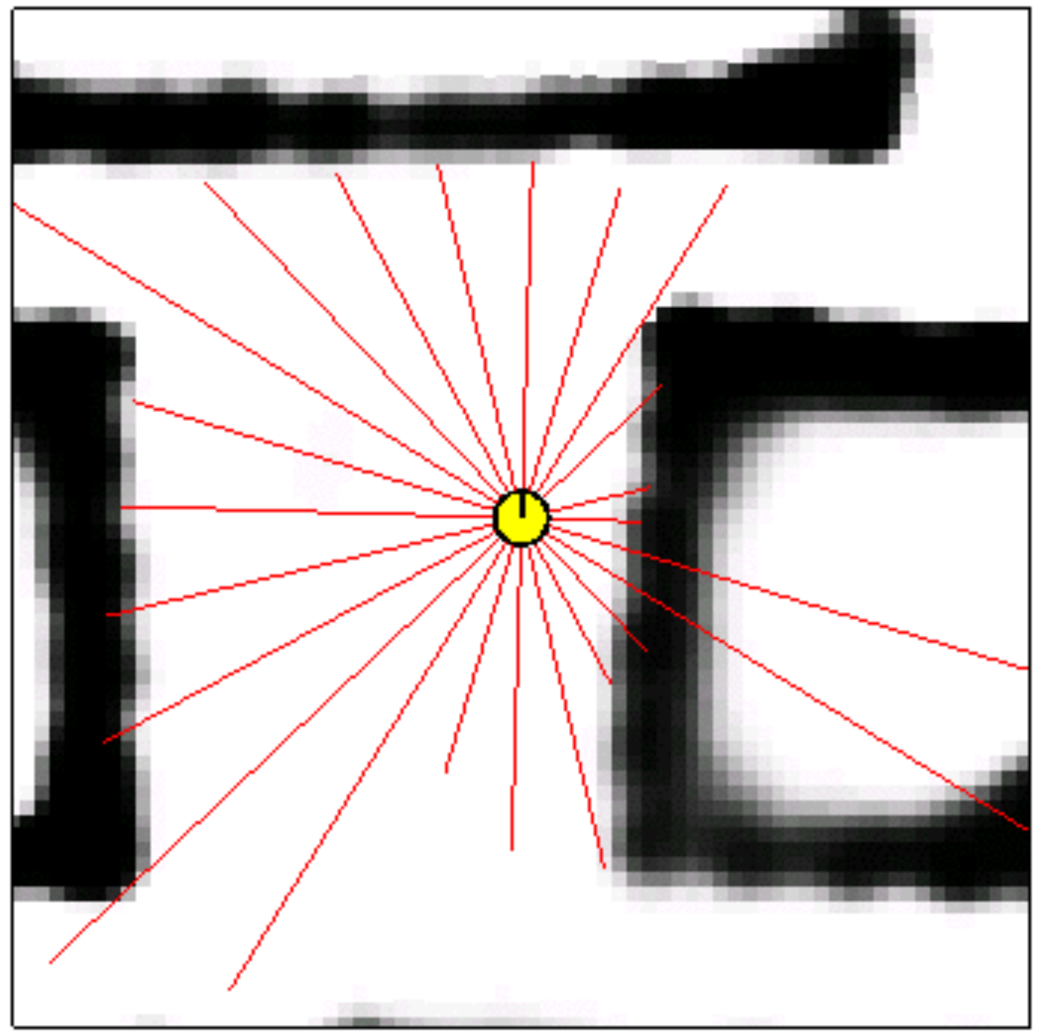
- Scan z consists of K measurements.

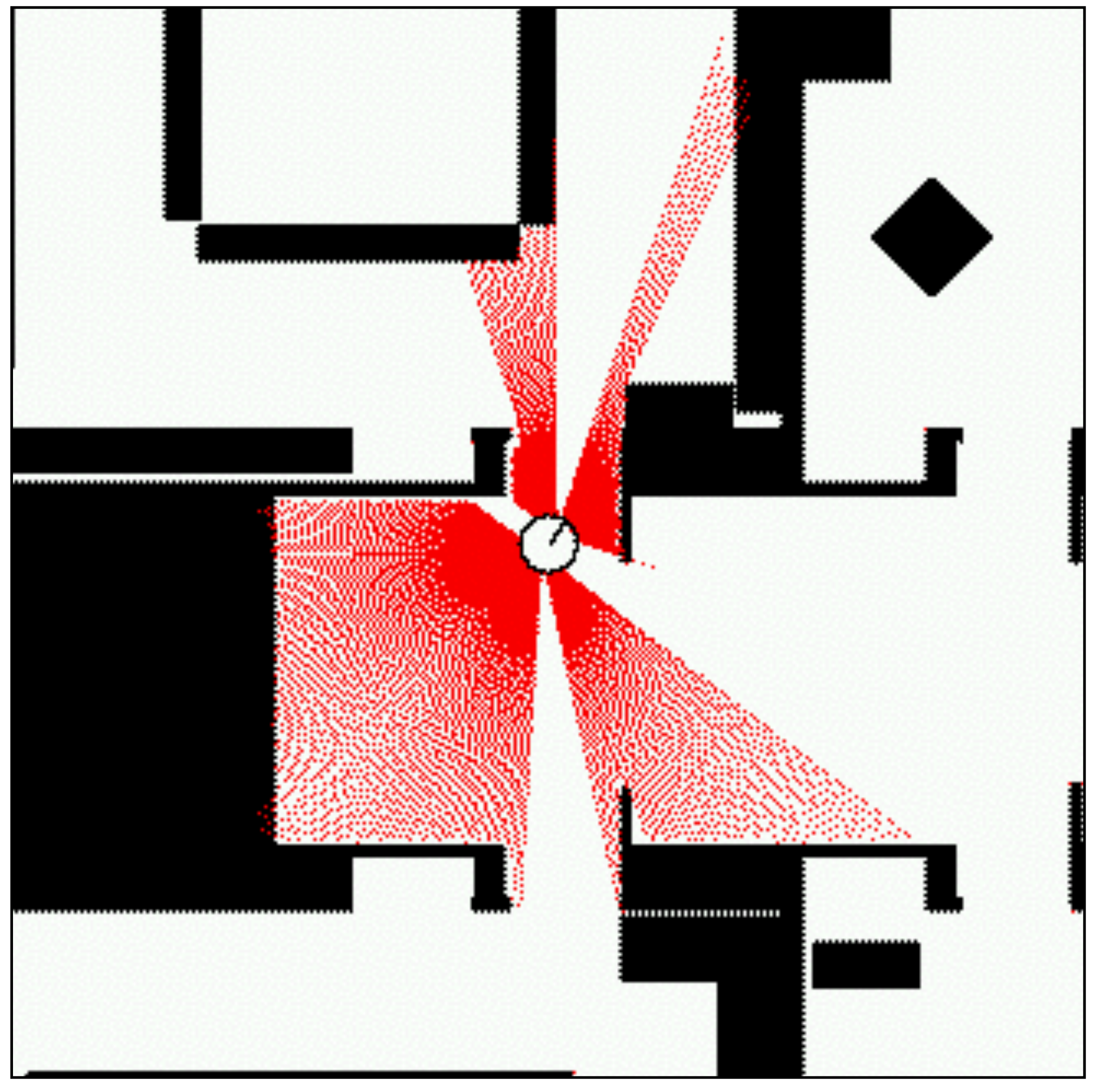
$$z = \{z_1, z_2, \ldots, z_K\}$$

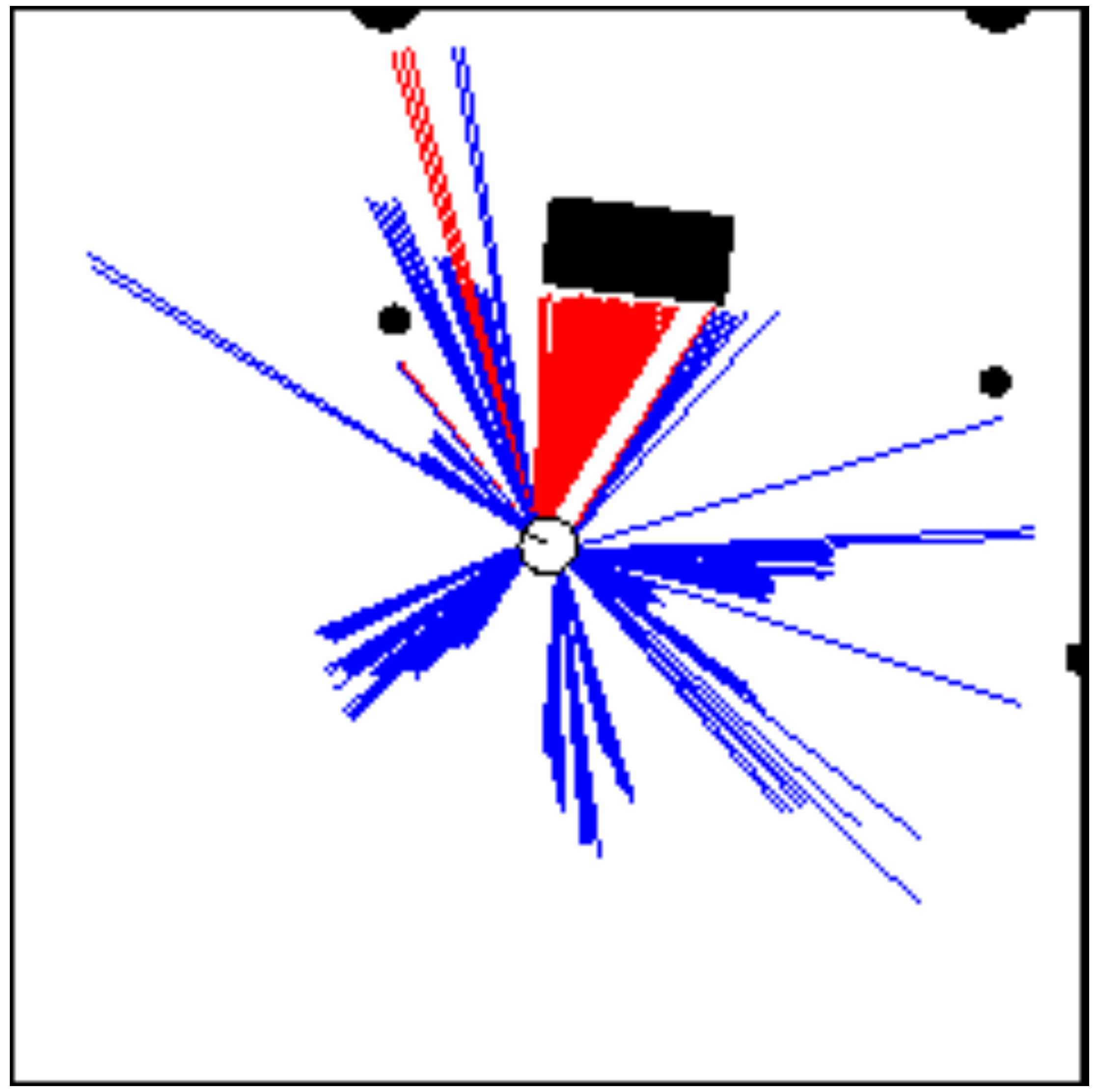
- Individual measurements are independent given the robot position and a map.

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

Beam-based Sensor Model







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

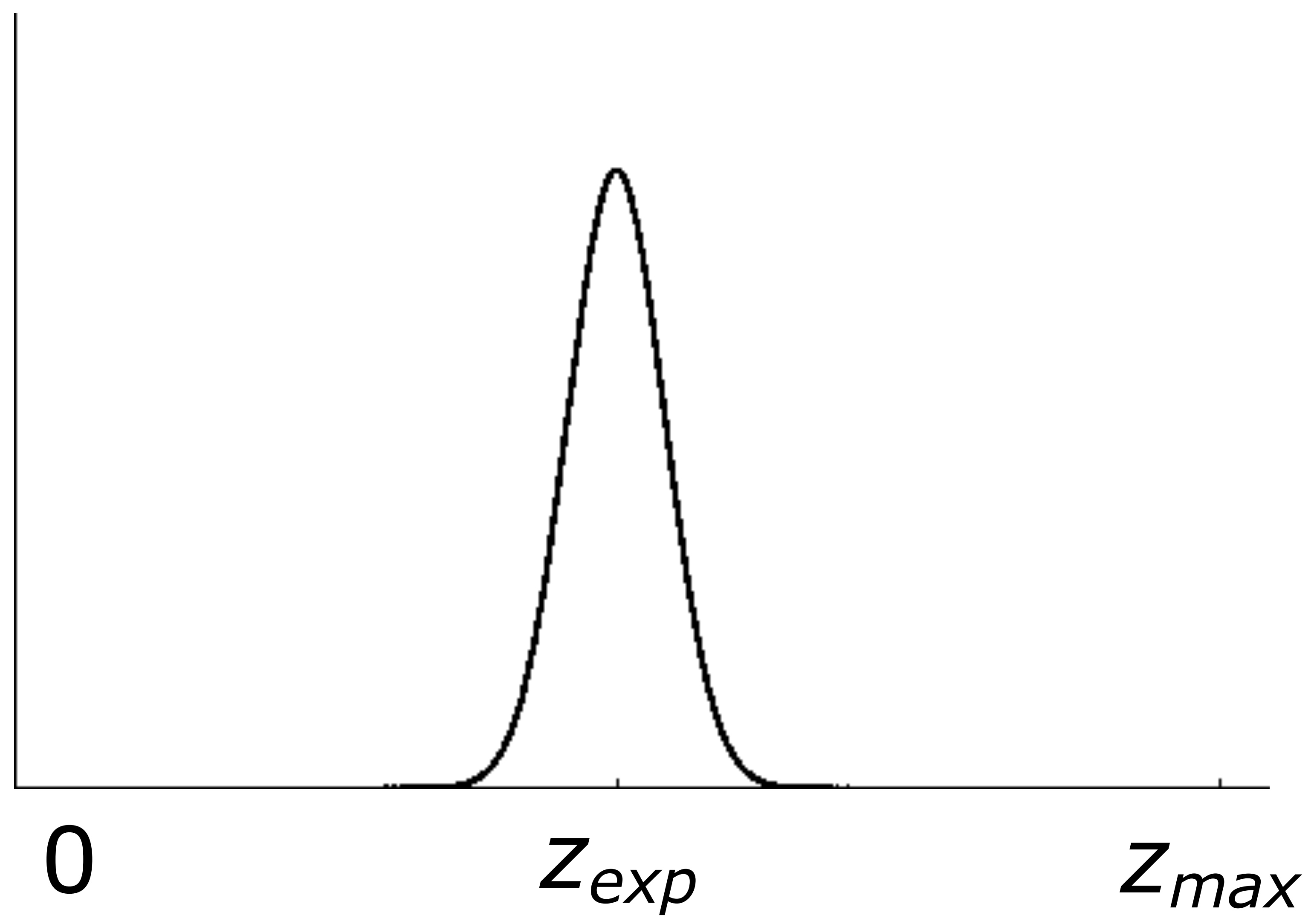
Slide borrowed from Dieter Fox

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.

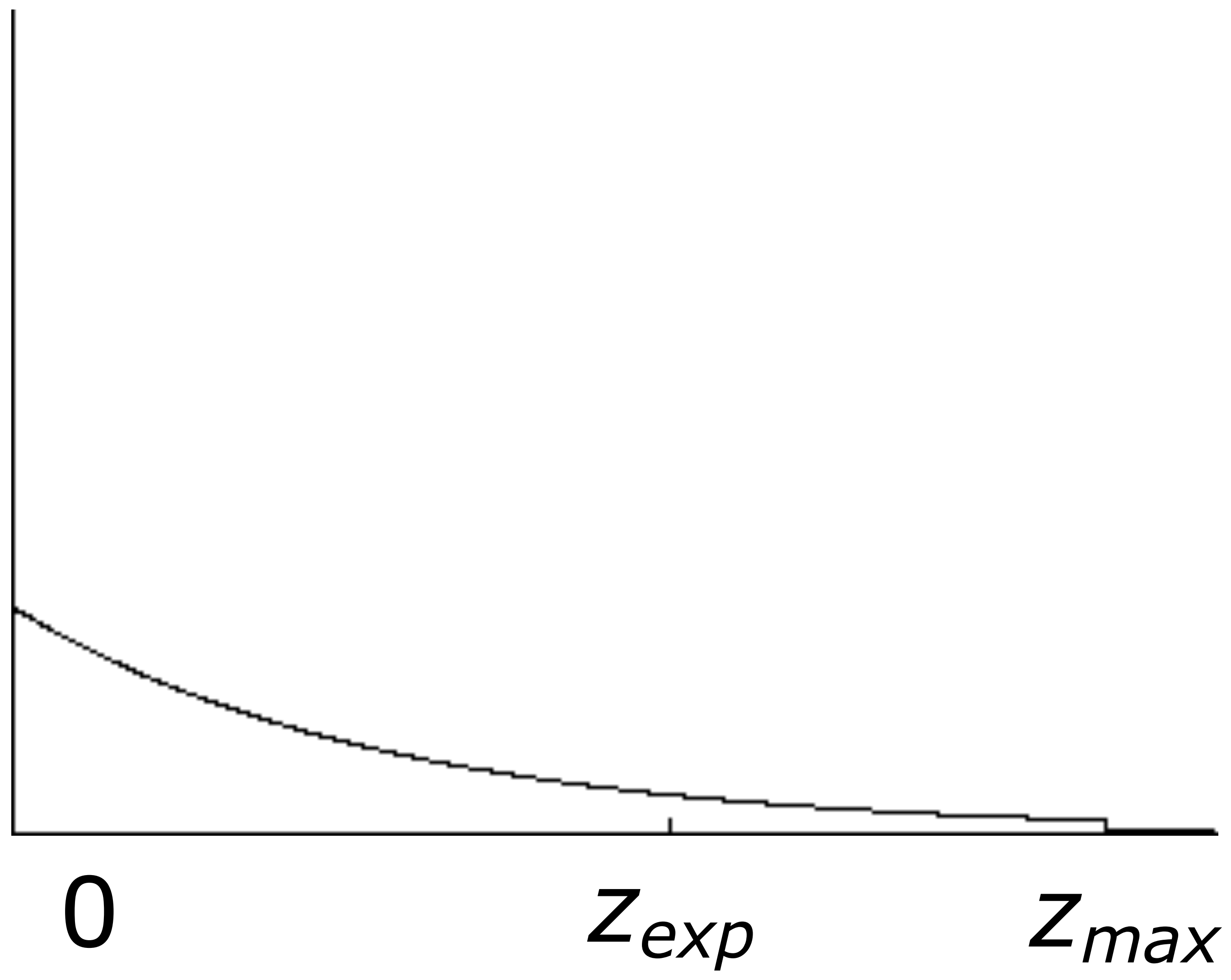
Beam-based Proximity Model

Measurement noise



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

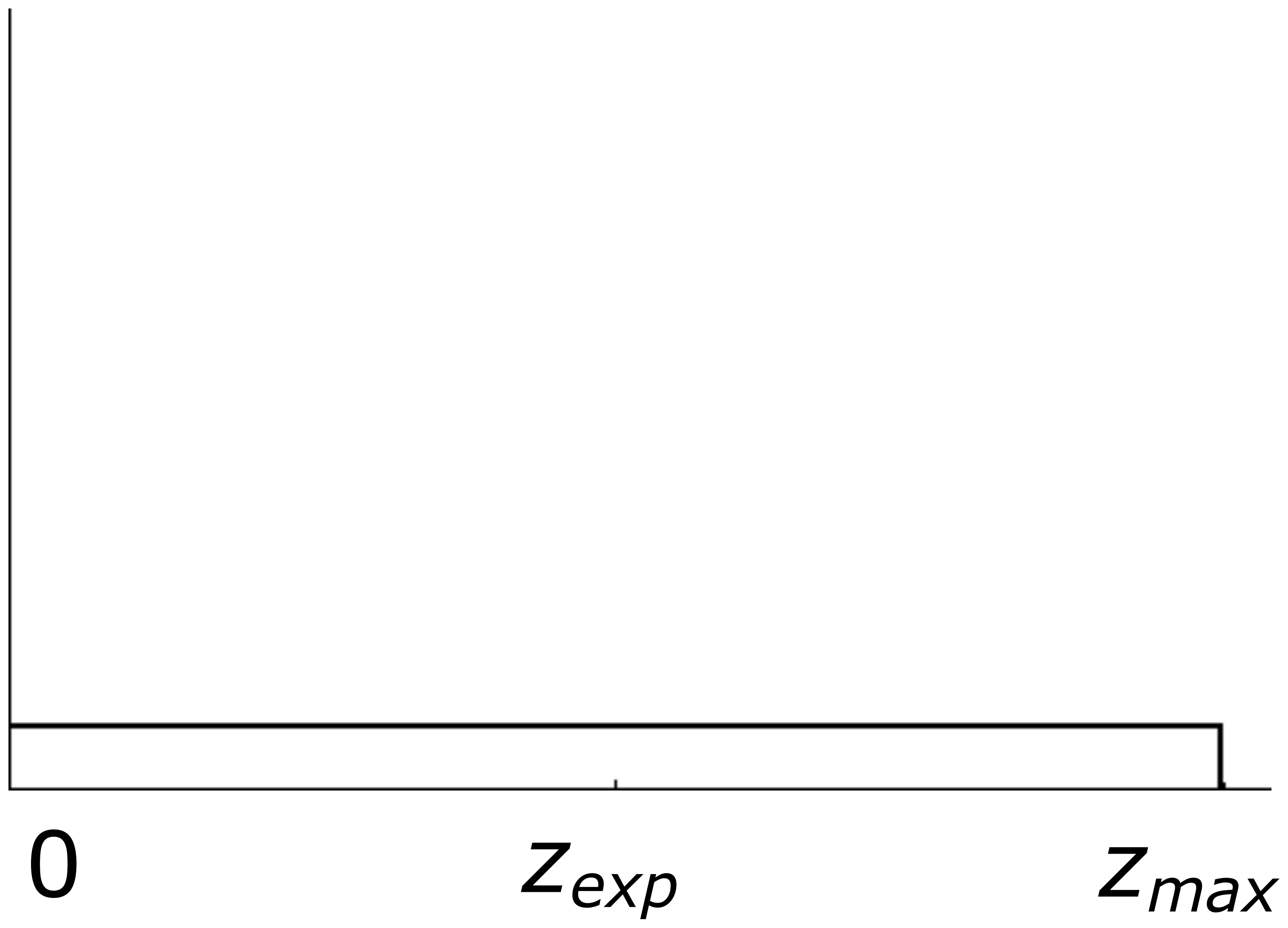
Unexpected obstacles



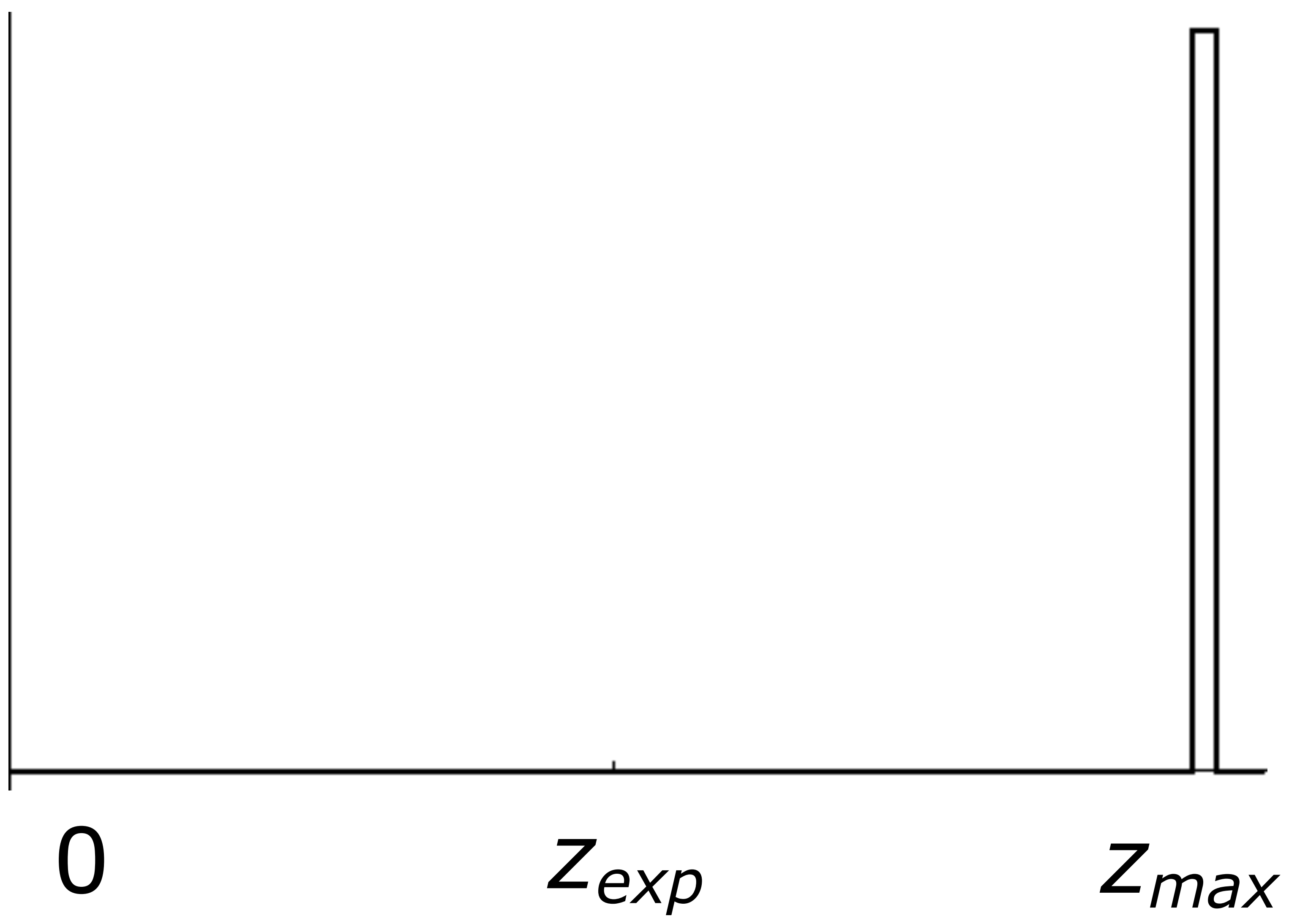
$$P_{\text{unexp}}(z \mid x,m) = \eta \, \lambda \, e^{-\lambda z}$$

Beam-based Proximity Model

Random measurement



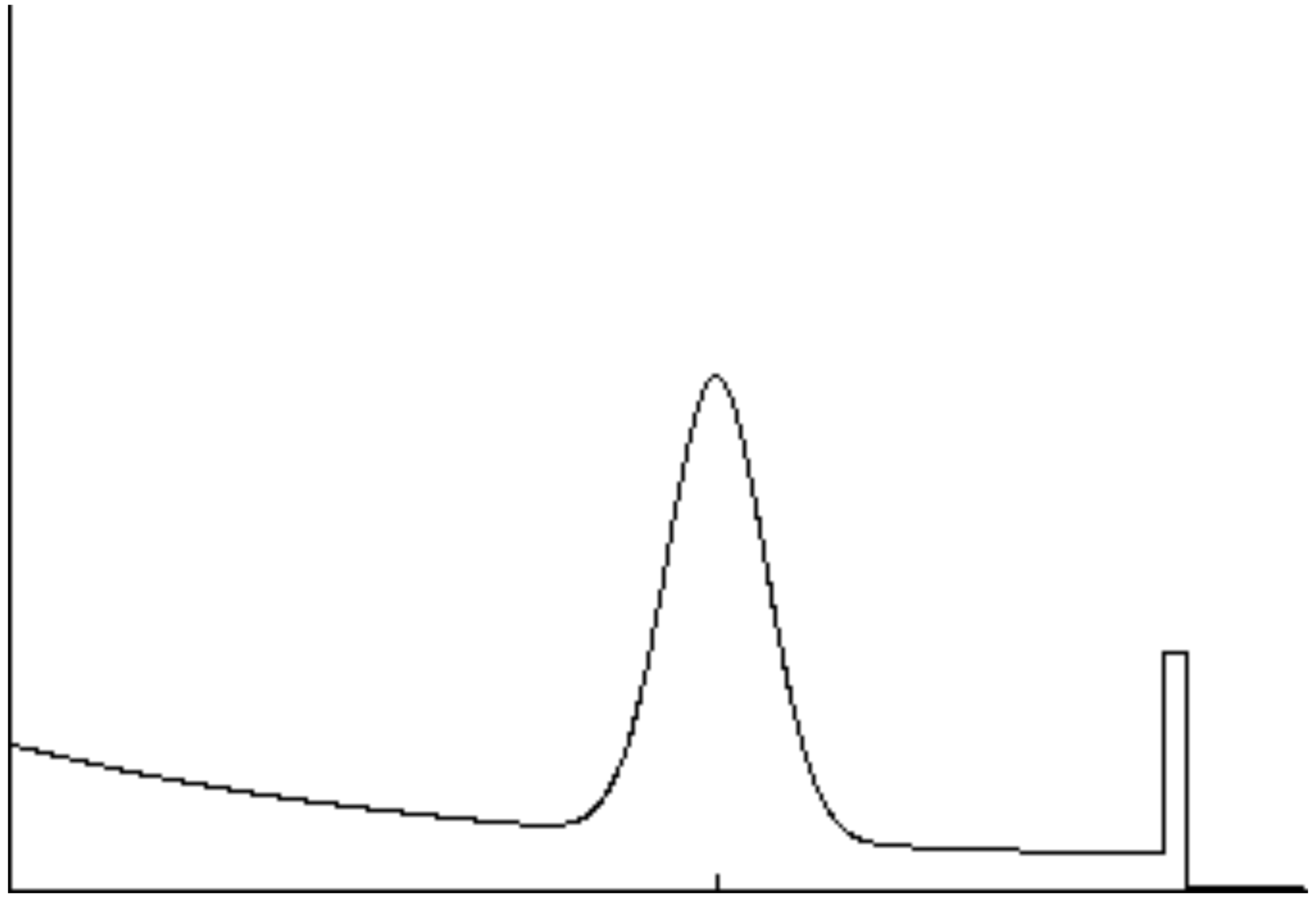
Max range



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

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

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

- Maximize log likelihood of the data z:

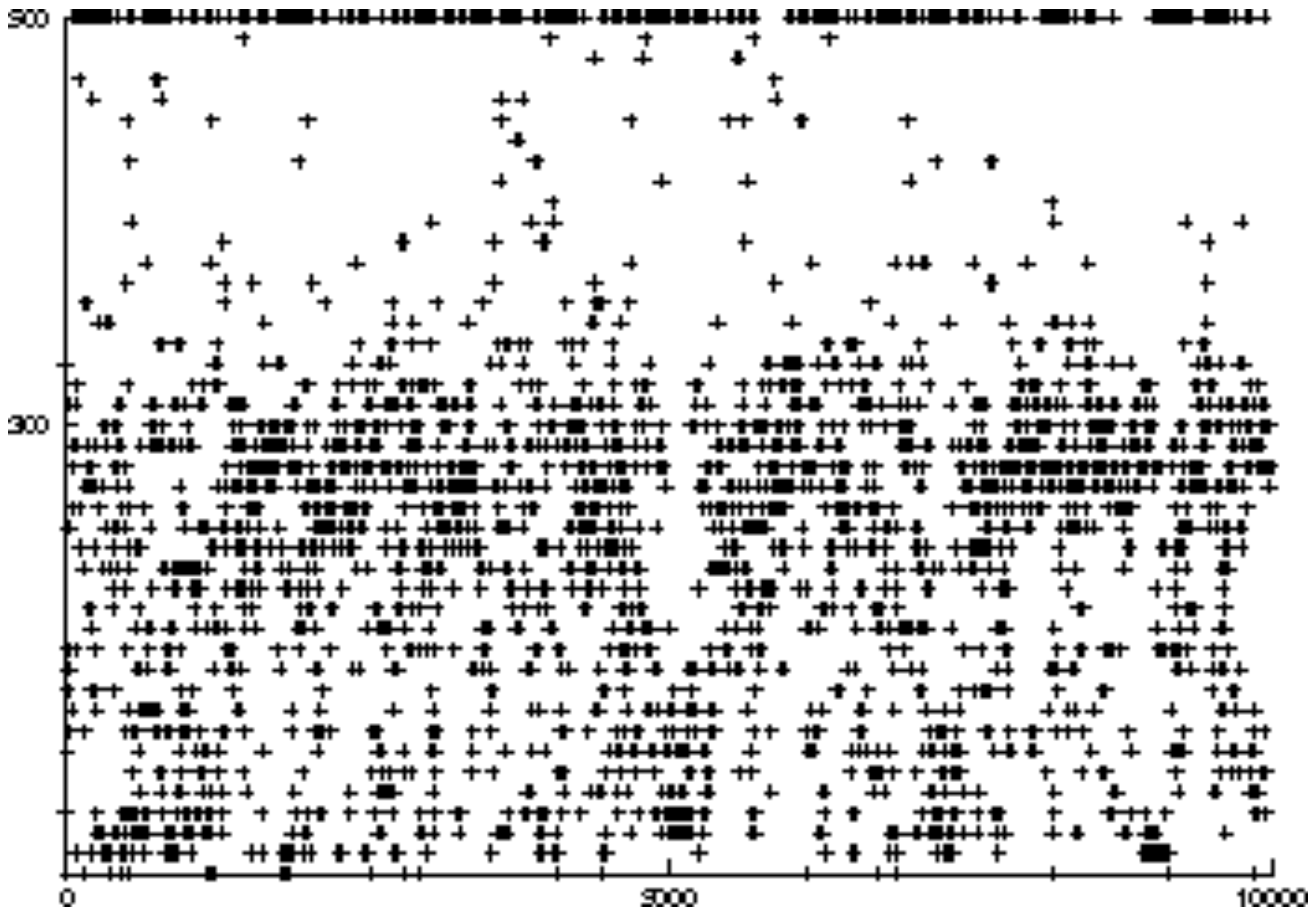
$$P(z \mid z_{\exp})$$

- Search parameter space.

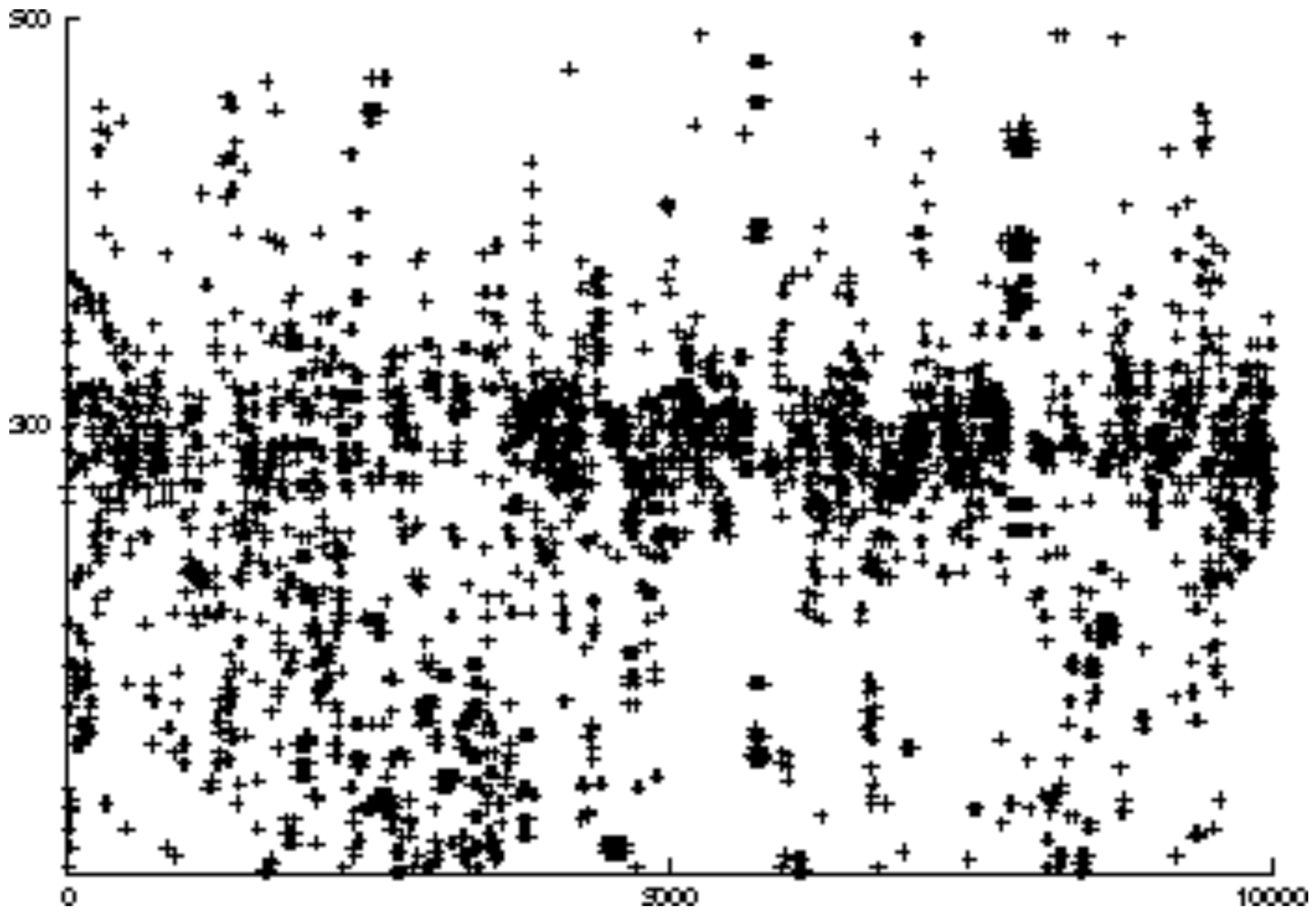
- EM to find mixture parameters
  - Assign measurements to densities.
  - Estimate densities using assignments.
  - Reassign measurements.

Raw Sensor Data

Measured distances for expected distance of 300 cm.



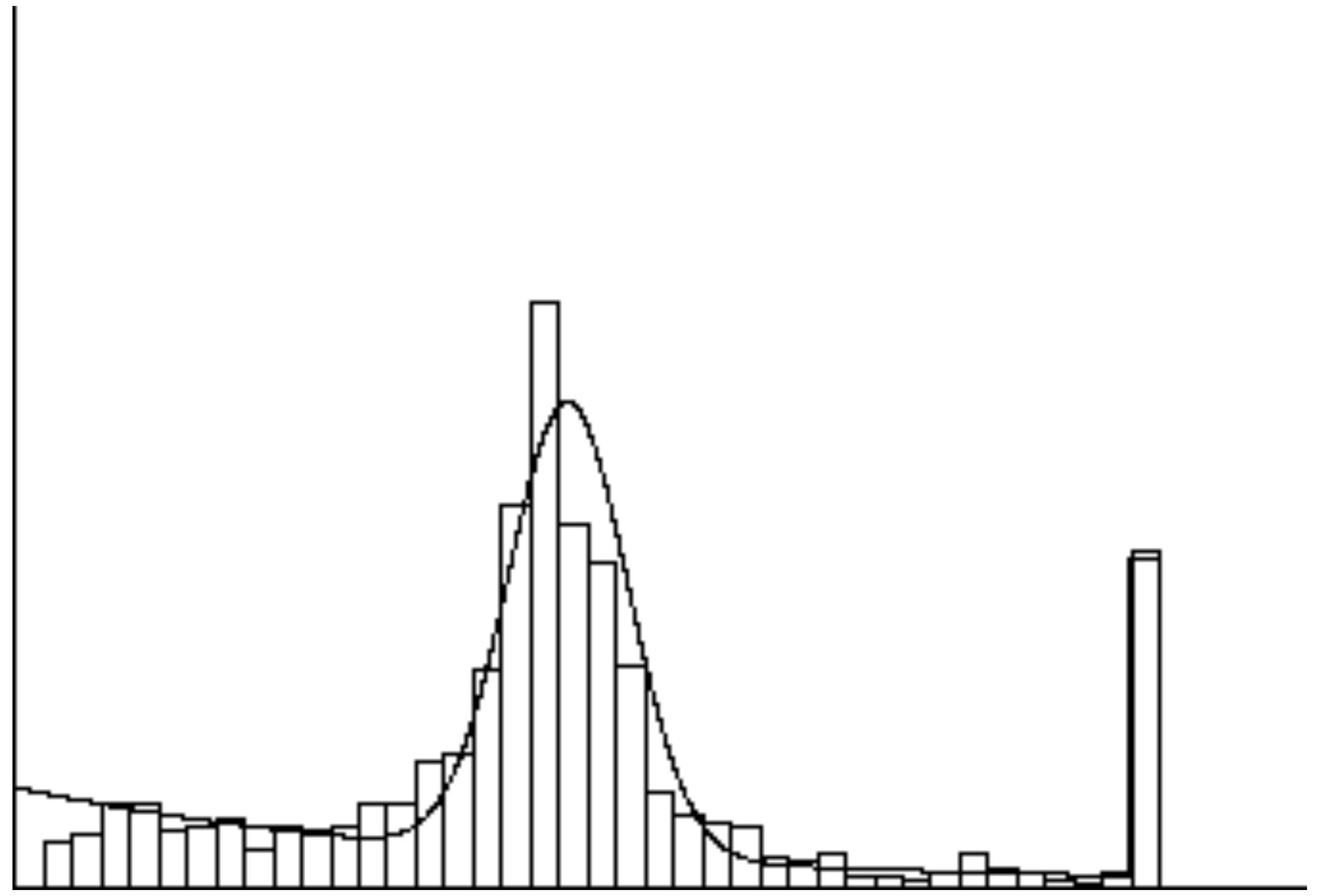
Sonar



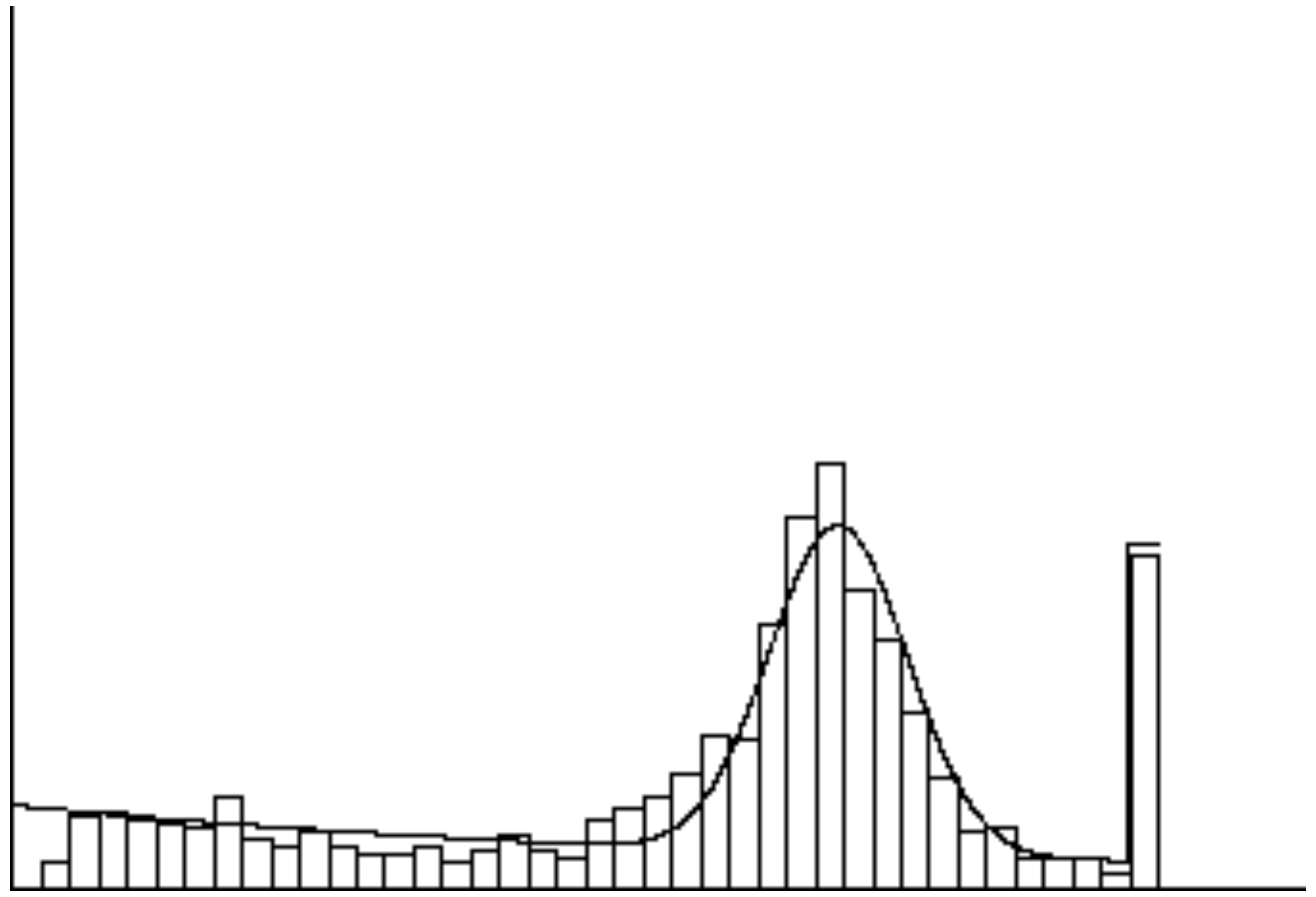
Laser

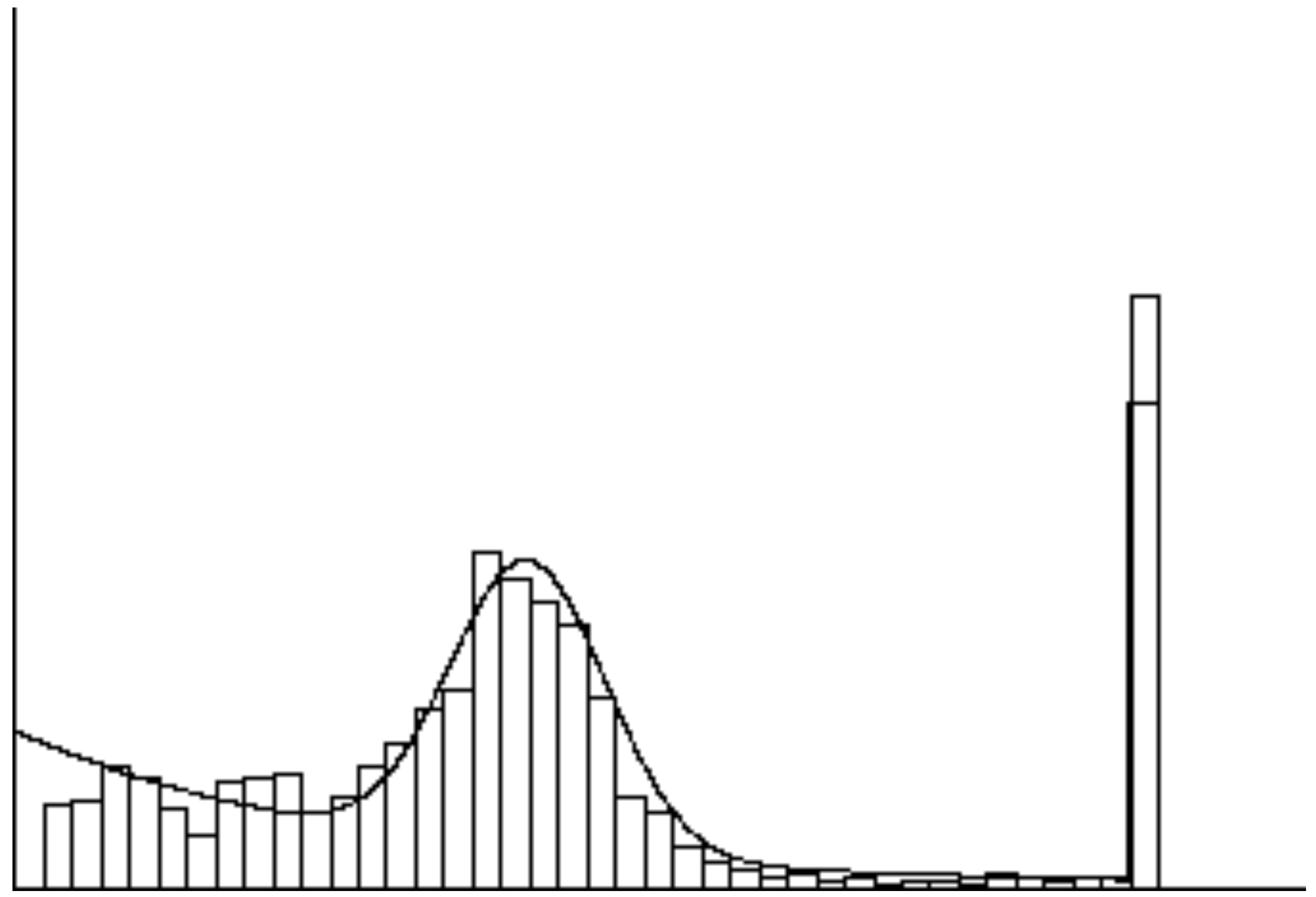
Slide borrowed from Dieter Fox

Approximation Results

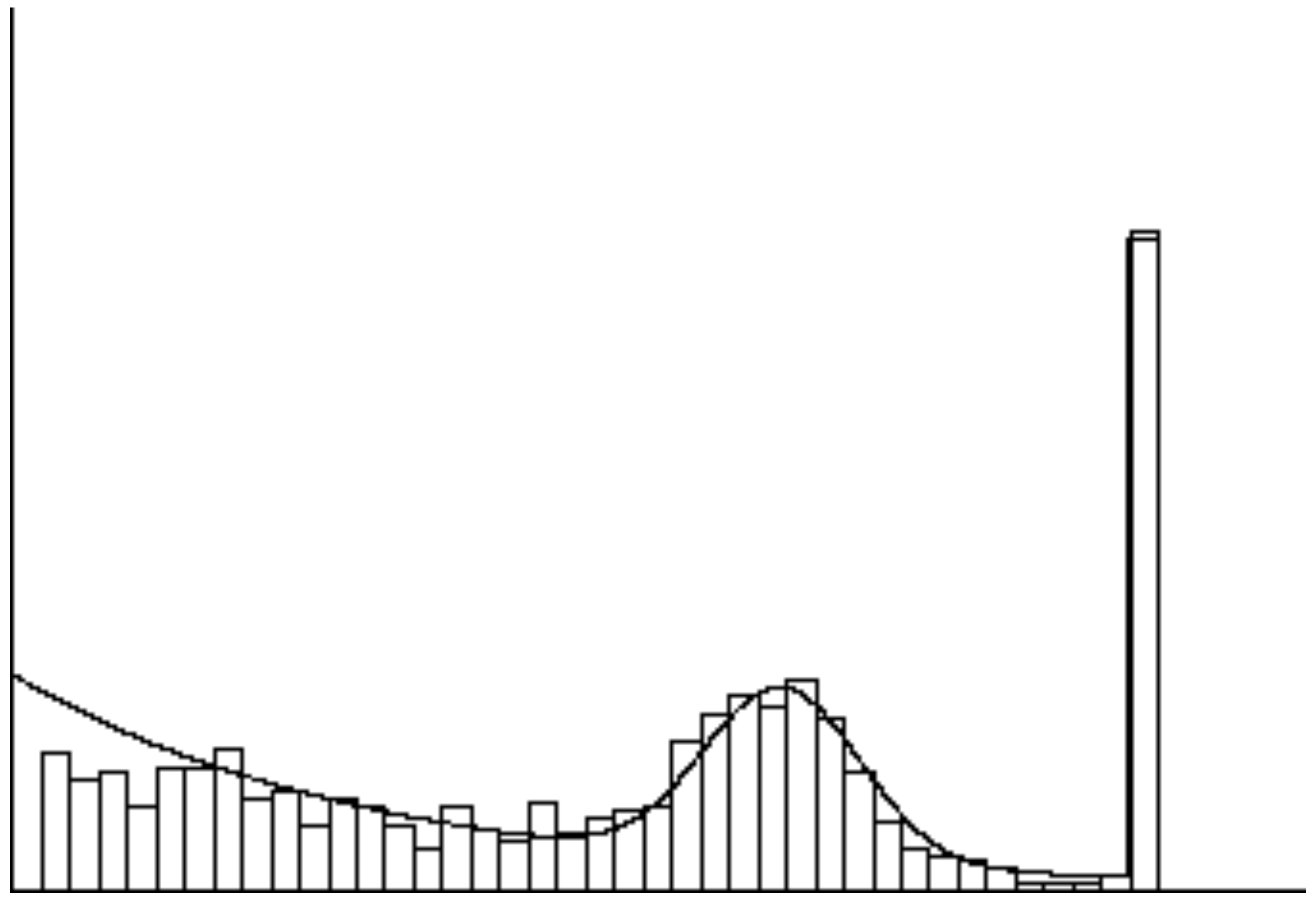


Laser





Sonar

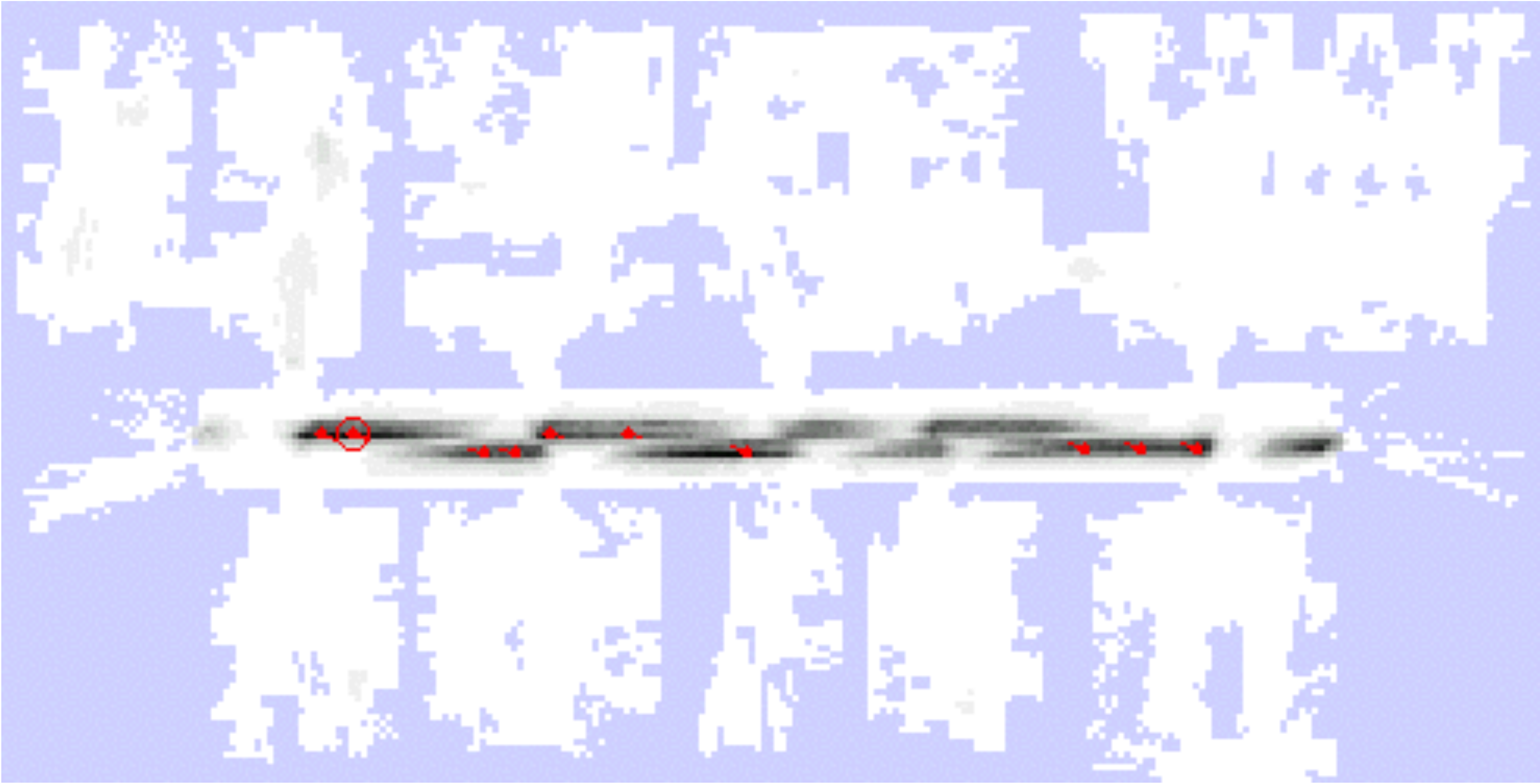


300cm

400cm

Example

z



$P(z|x,m)$

Summary Beam-based Model

- Assumes independence between beams.
  - Justification?
  - Overconfident!
- Models physical causes for measurements.
  - Mixture of densities for these causes.

- Implementation
  - Learn parameters based on real data.
  - Different models can be learned for different angles at which the sensor beam hits the obstacle.
  - Determine expected distances by ray-tracing.
  - Expected distances can be pre-processed.

Next Lecture
Mobile Robotics - III - Kalman

Final Projects from Previous Term

https://rpm-lab.github.io/CSCI5551-Fall23-S2/final_presentations/