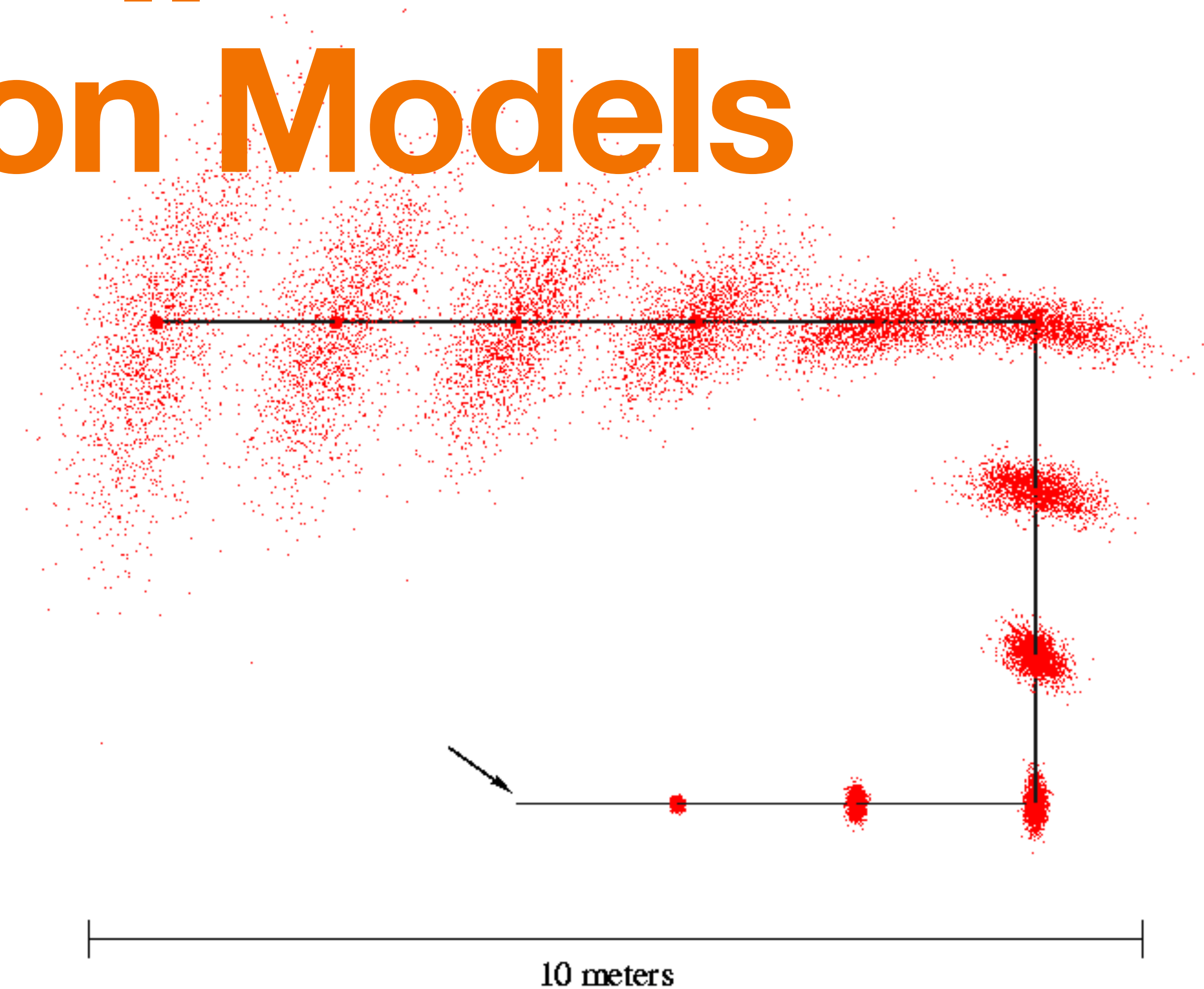
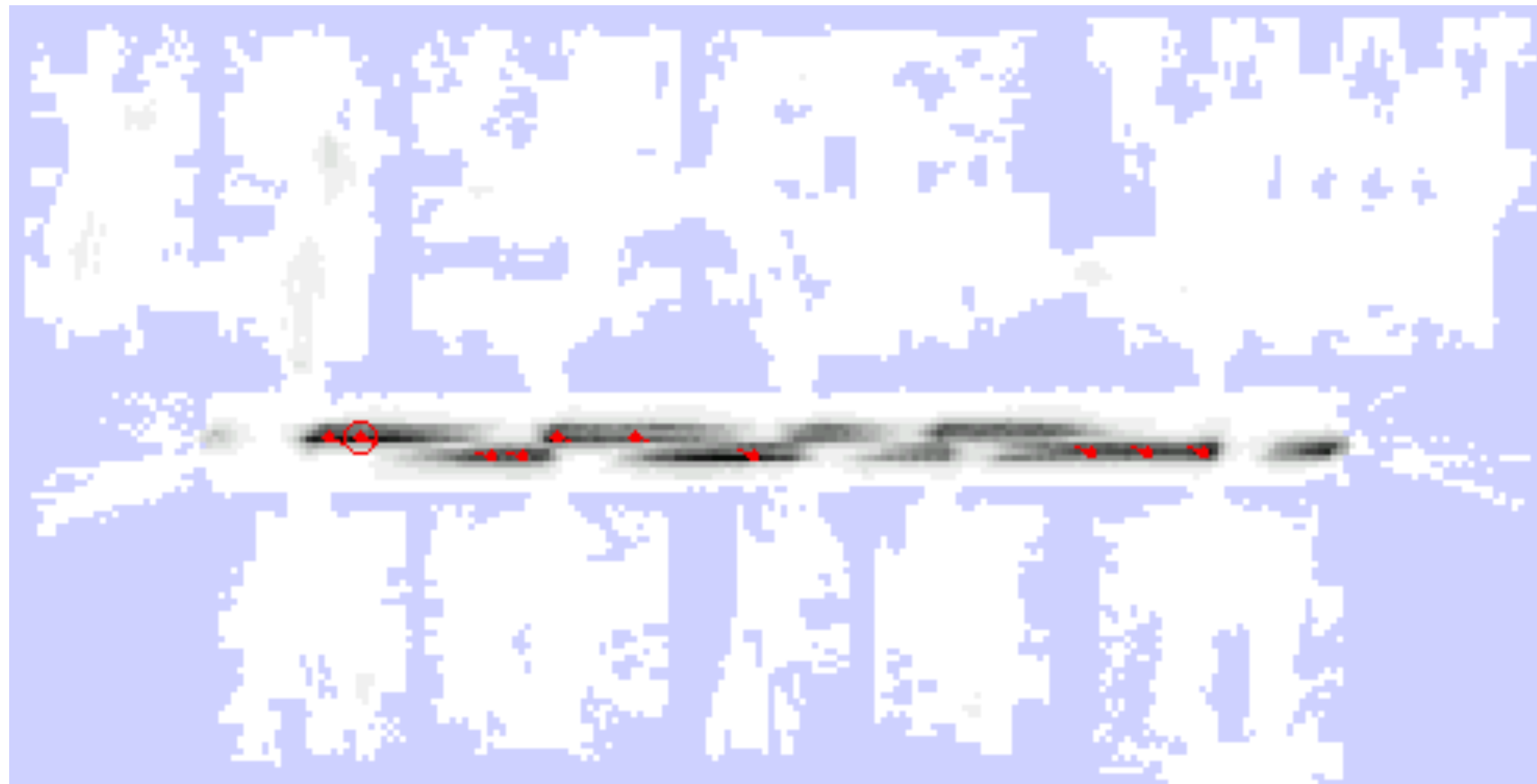


Lecture 17

Mobile Robotics - II - Sensor and Motion Models



Course logistics

- Quiz 8 was due today at noon.
- Project 6 was posted on 03/24 and will be on 04/02 (Wednesday).
- Group formation for P7 and Final Project.
 - There are a few students who are yet to form the teams!
 - Let us know by email and we are happy to help.



Previously

Joint and Conditional Probability

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

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

$$P(x, y) = P(x|y)P(y)$$
- If X and Y are independent then

$$P(x, y) = P(x)P(y)$$
- If X and Y are independent then

$$P(x|y) = P(x)$$

Bayes Formula

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

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

Conditioning

- Bayes rule and background knowledge:

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

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

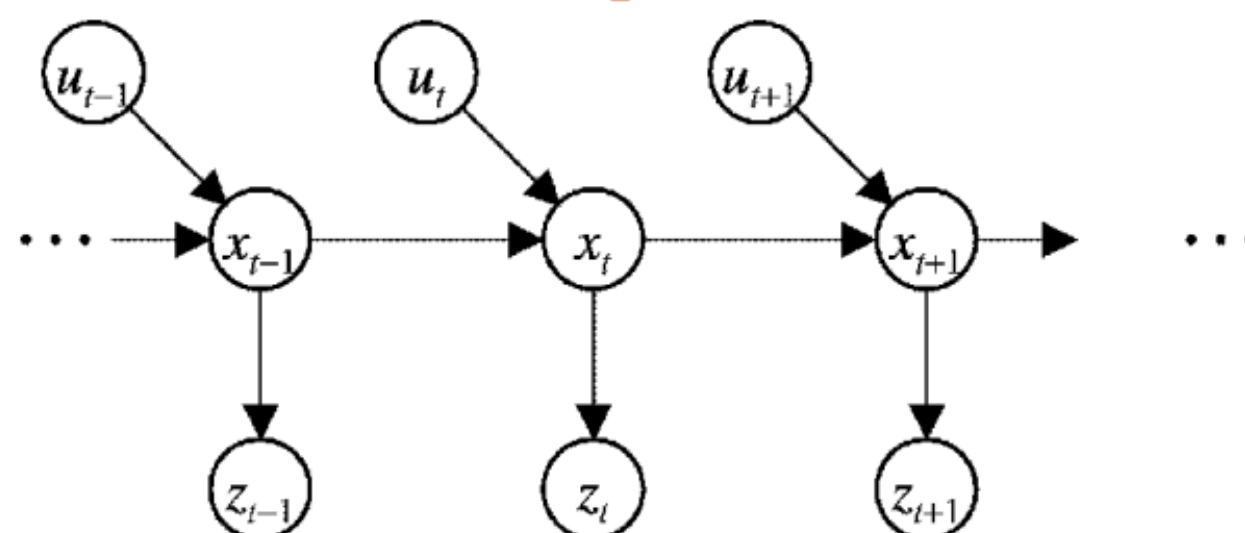
Recursive Bayesian Updating

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

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

$$\begin{aligned} P(x|z_1, \dots, z_n) &= \frac{P(z_n|x)P(x|z_1, \dots, z_{n-1})}{P(z_n|z_1, \dots, z_{n-1})} \\ &= \eta P(z_n|x)P(x|z_1, \dots, z_{n-1}) \\ &= \eta_{1..n} \prod_{i=1..n} P(z_i|x)P(x) \end{aligned}$$

Markov Assumption



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

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

Bayes Filters

z = observation
 u = action
 x = state

$$Bel(x_t) = P(x_t | u_1, z_1, \dots, u_t, z_t)$$

Bayes $= \eta P(z_t | x_t, u_1, z_1, \dots, u_t) P(x_t | u_1, z_1, \dots, u_t)$


Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_1, \dots, u_t)$

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

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

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

Probabilistic Motion Models

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


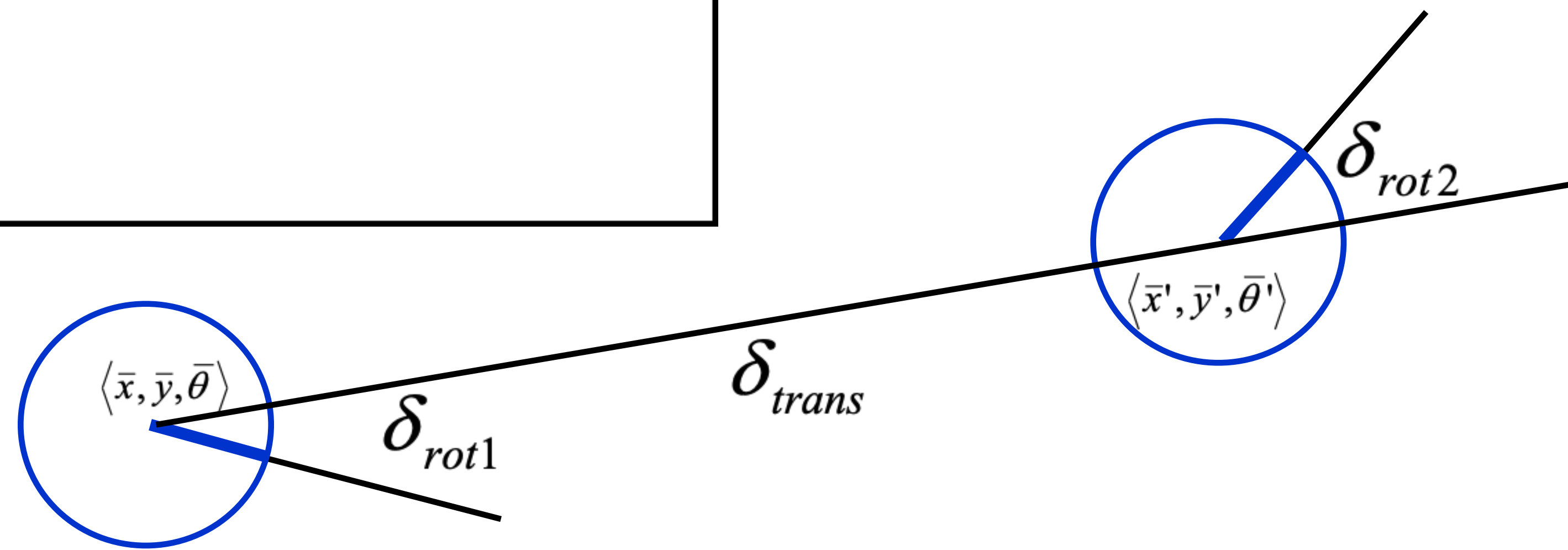
Probabilistic Kinematics

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

$$\delta_{trans} =$$

$$\delta_{rot1} =$$

$$\delta_{rot2} =$$



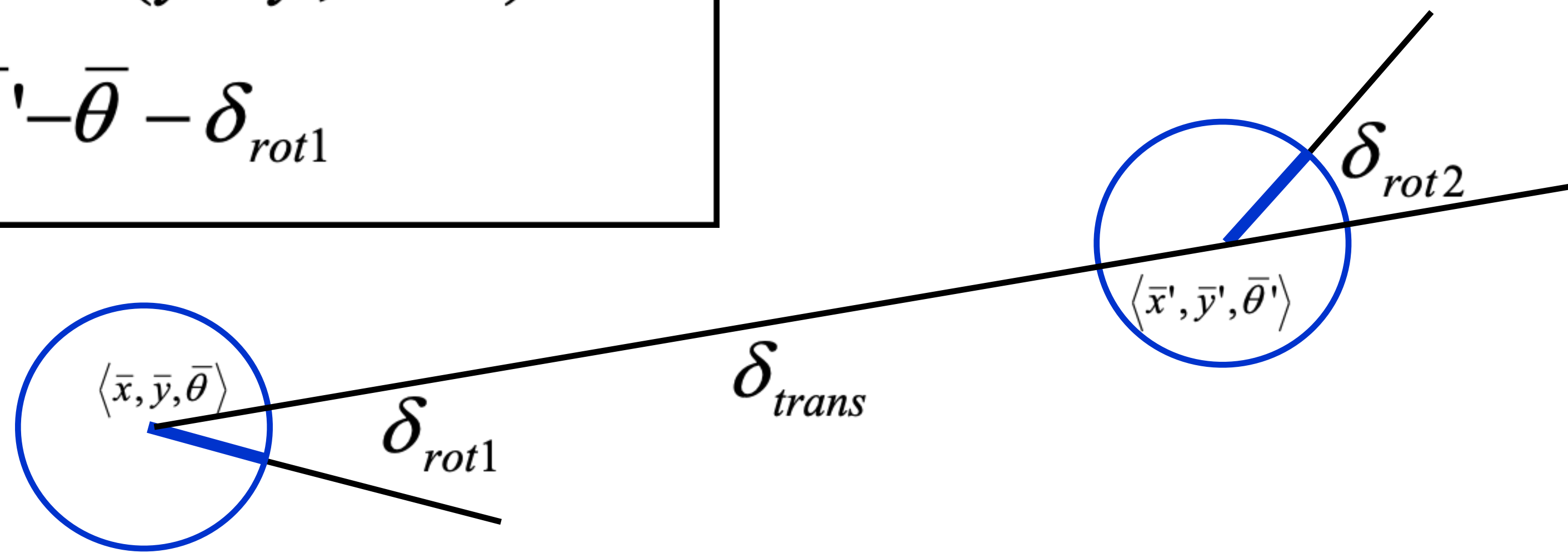
Probabilistic Kinematics

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

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

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

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

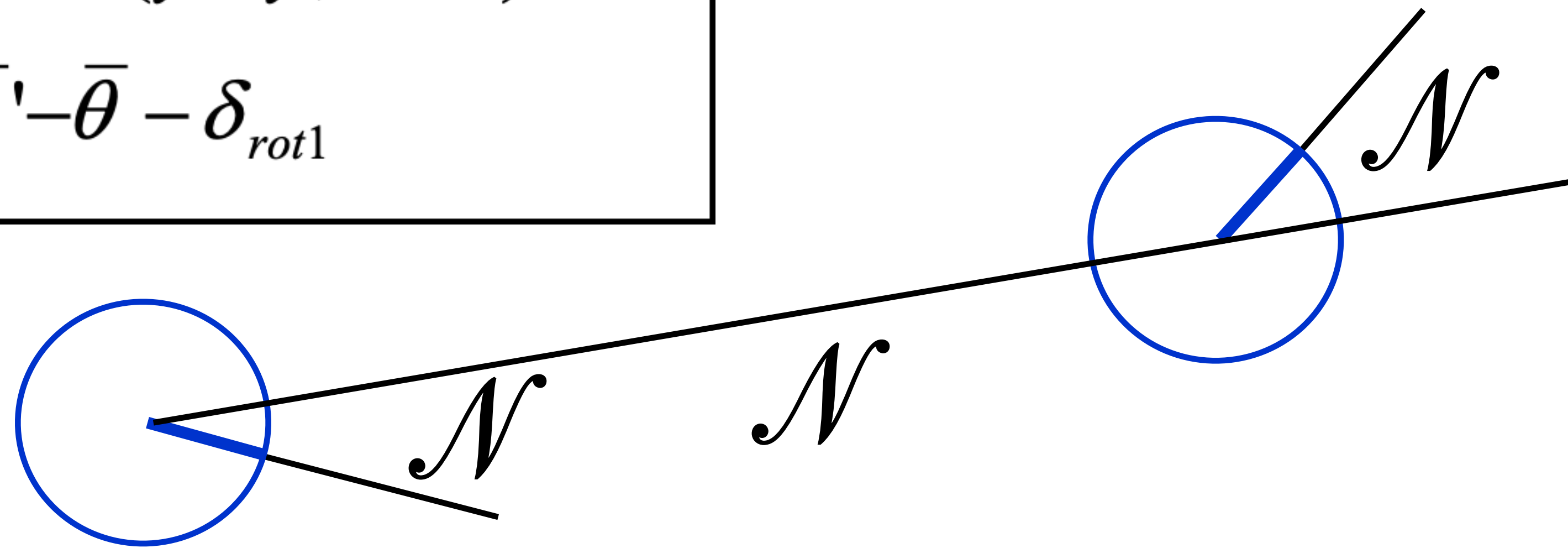


Noise Model for Motion

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

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

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



Noise Model for Motion

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

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

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

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



Odometry Motion Model

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



Odometry Motion Model

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

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. $p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 \hat{\delta}_{rot1}^2 + \alpha_2 \hat{\delta}_{trans}^2)$

8. $p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}_{trans}^2 + \alpha_4 (\hat{\delta}_{rot1}^2 + \hat{\delta}_{rot2}^2))$

9. $p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 \hat{\delta}_{rot2}^2 + \alpha_2 \hat{\delta}_{trans}^2)$

10. Return $p_1 * p_2 * p_3$



Odometry Motion Model

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

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. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$

$$x = \langle x, y, \theta \rangle$$
$$x' = \langle x', y', \theta' \rangle$$

Finding the posterior

7. $p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 \hat{\delta}_{rot1}^2 + \alpha_2 \hat{\delta}_{trans}^2)$

8. $p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}_{trans}^2 + \alpha_4 (\hat{\delta}_{rot1}^2 + \hat{\delta}_{rot2}^2))$

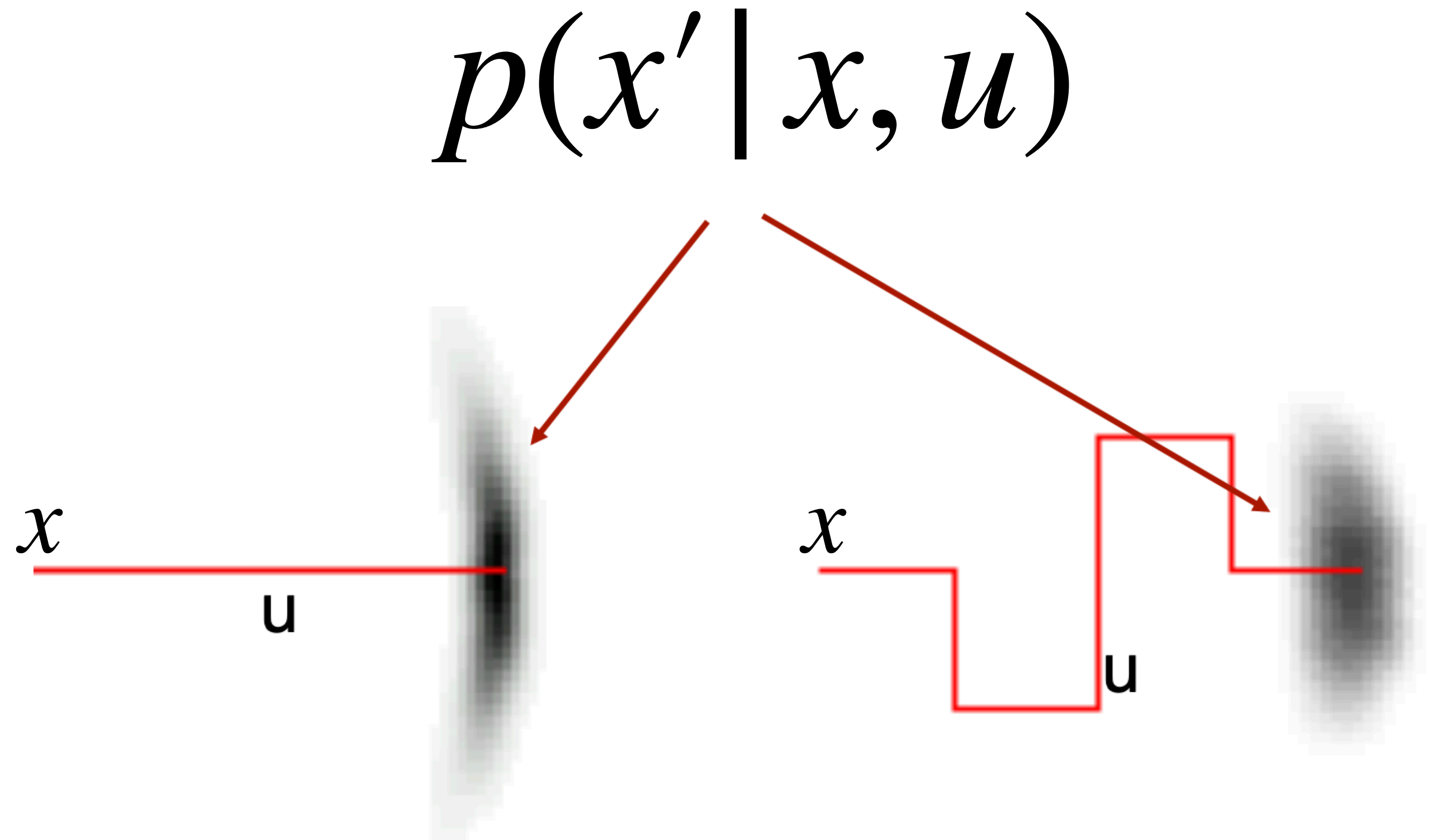
9. $p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 \hat{\delta}_{rot2}^2 + \alpha_2 \hat{\delta}_{trans}^2)$

10. Return $p_1 * p_2 * p_3$



Odometry Motion Model

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



This is a projected illustration ignoring the θ

Sample Odometry Motion Model

Sample for x_t

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



Sample Odometry Motion Model

Sample for x_t

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

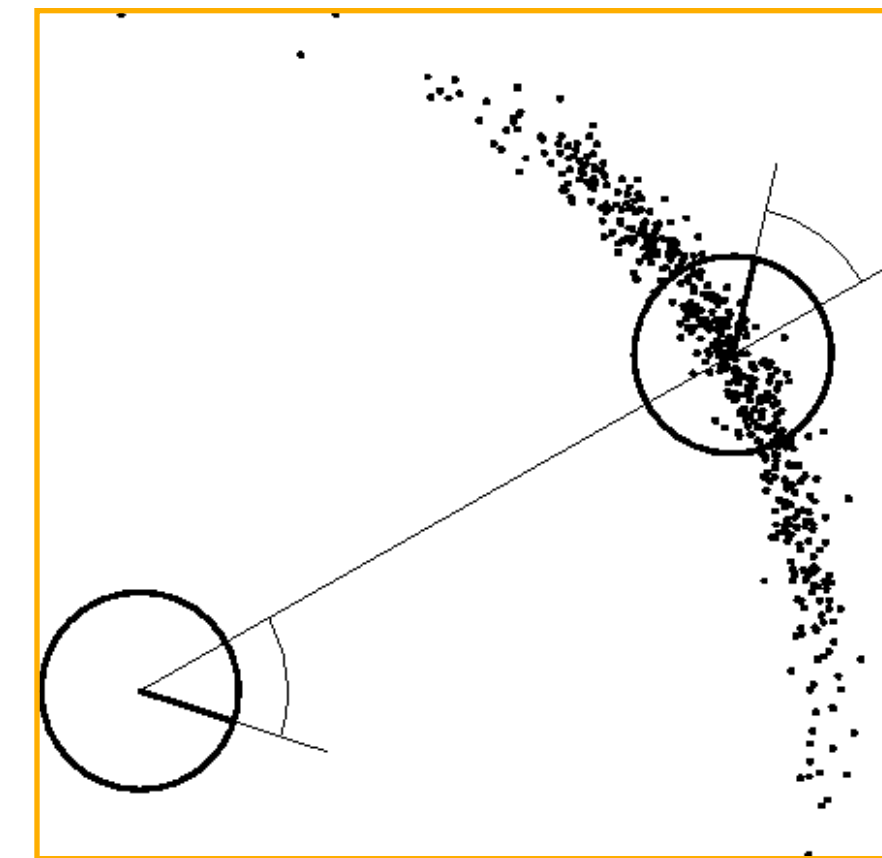
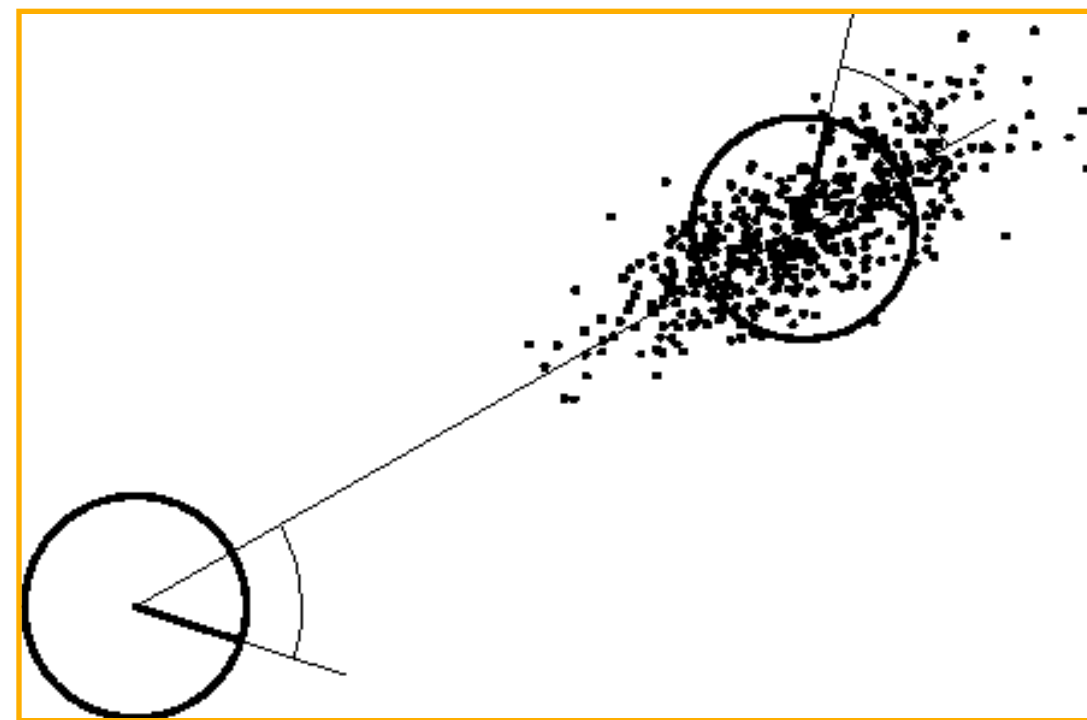
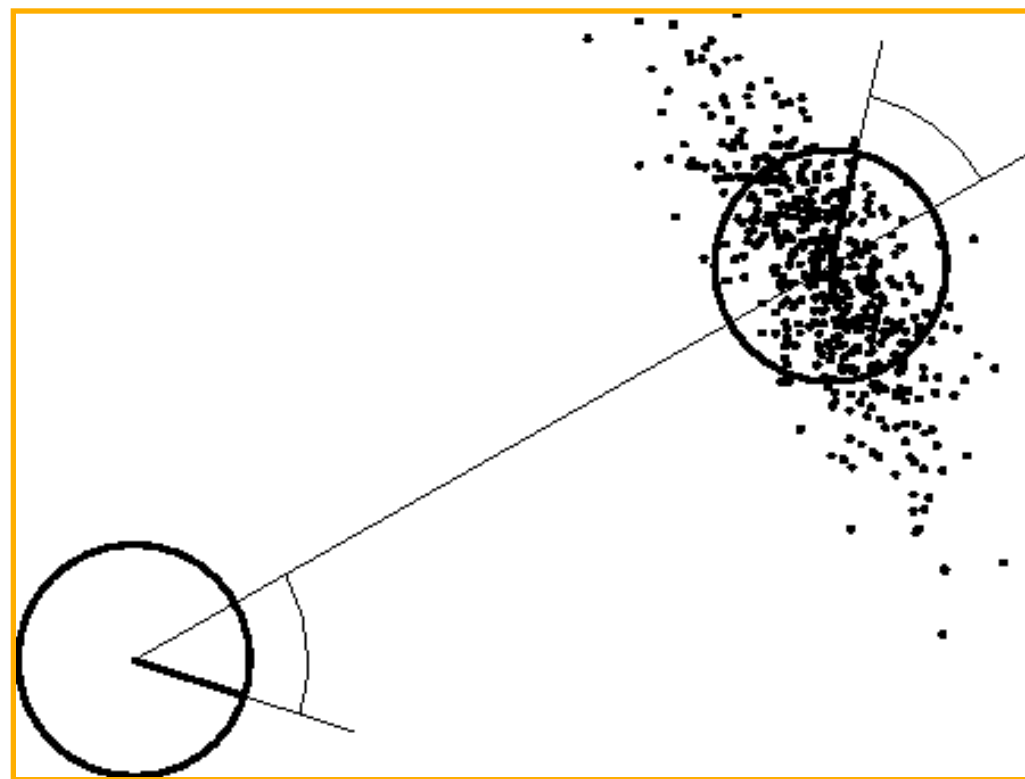
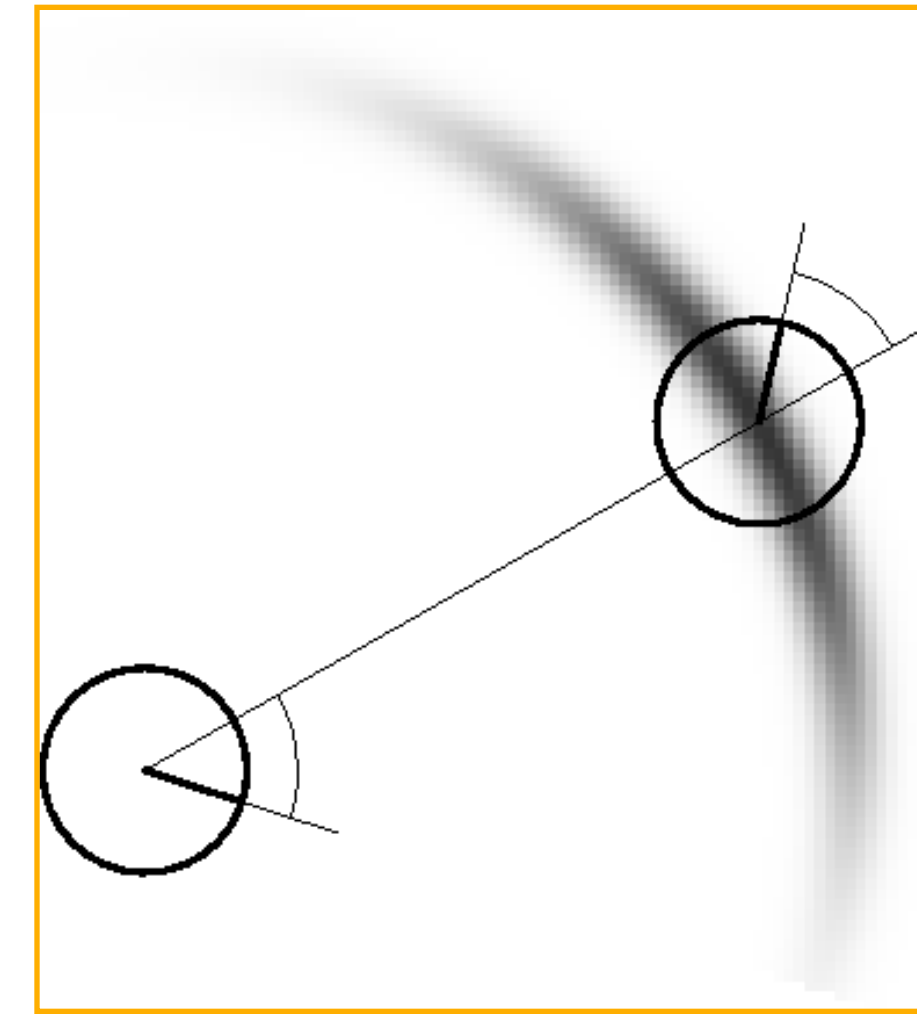
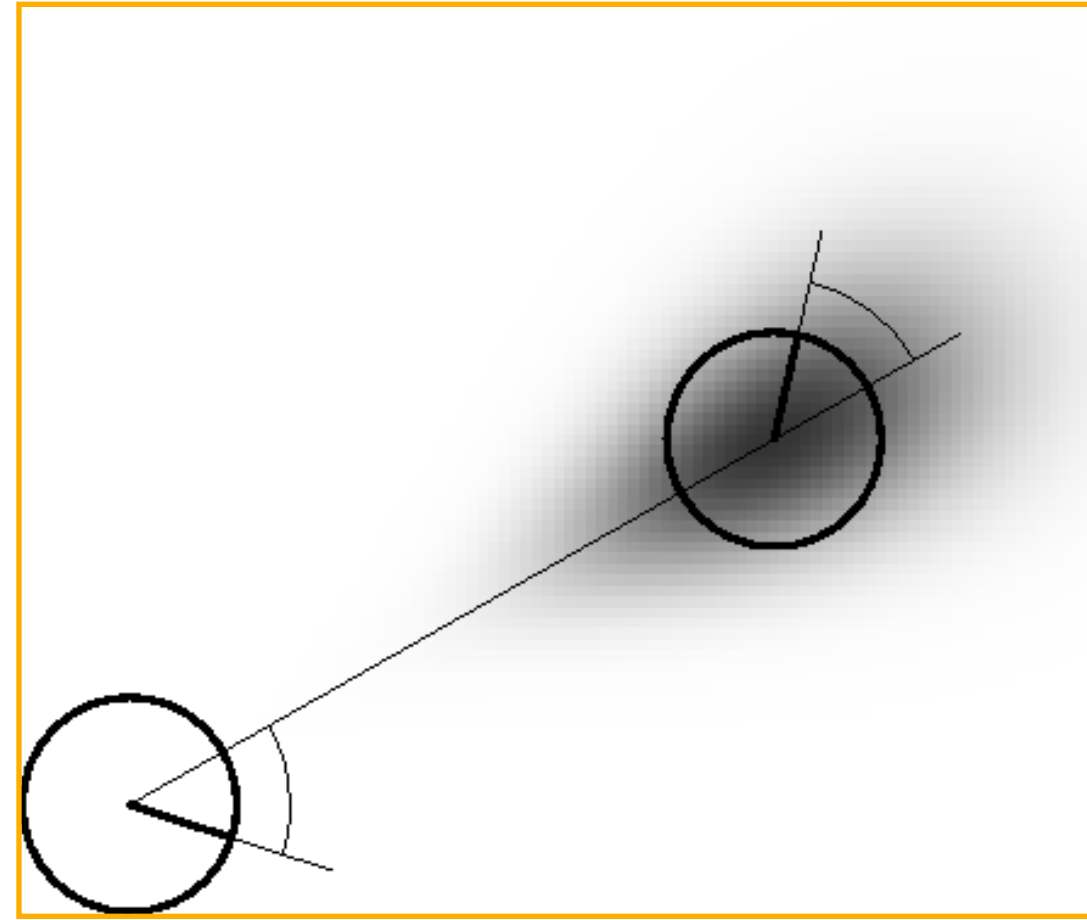
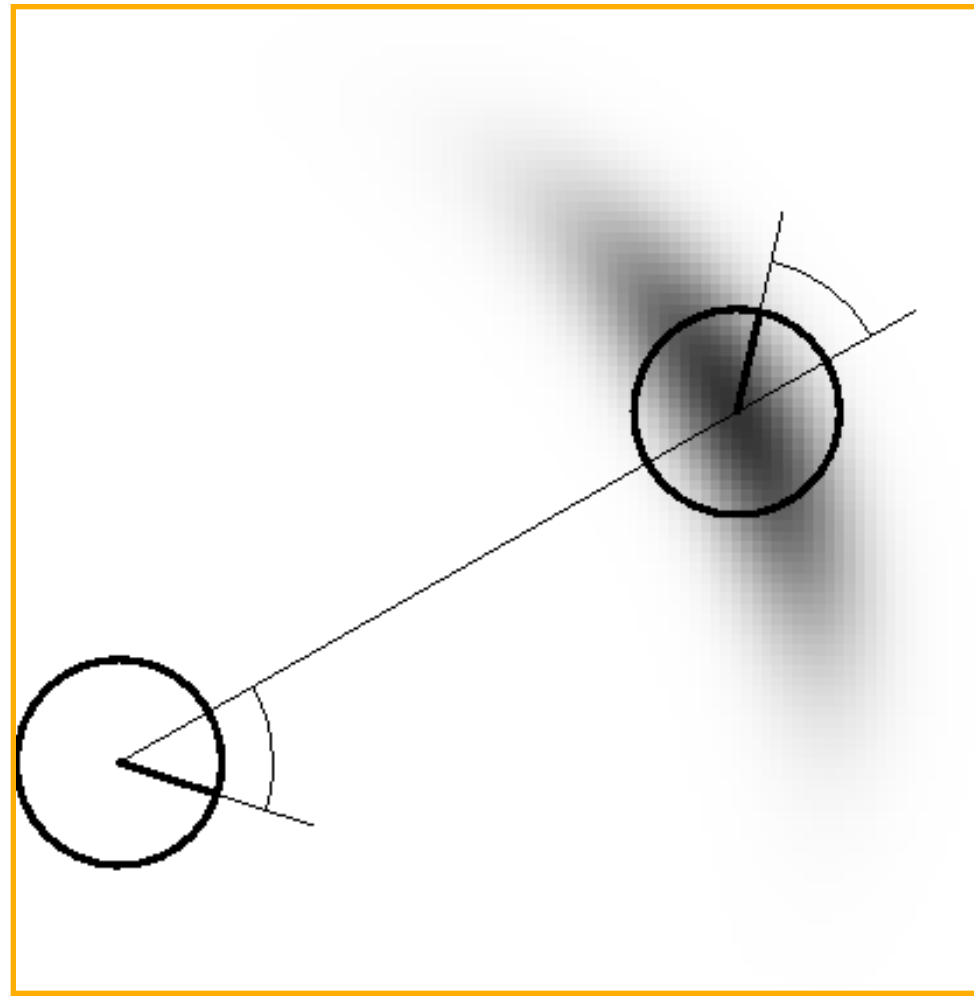
1. Algorithm **sample_motion_model** (u, x):

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

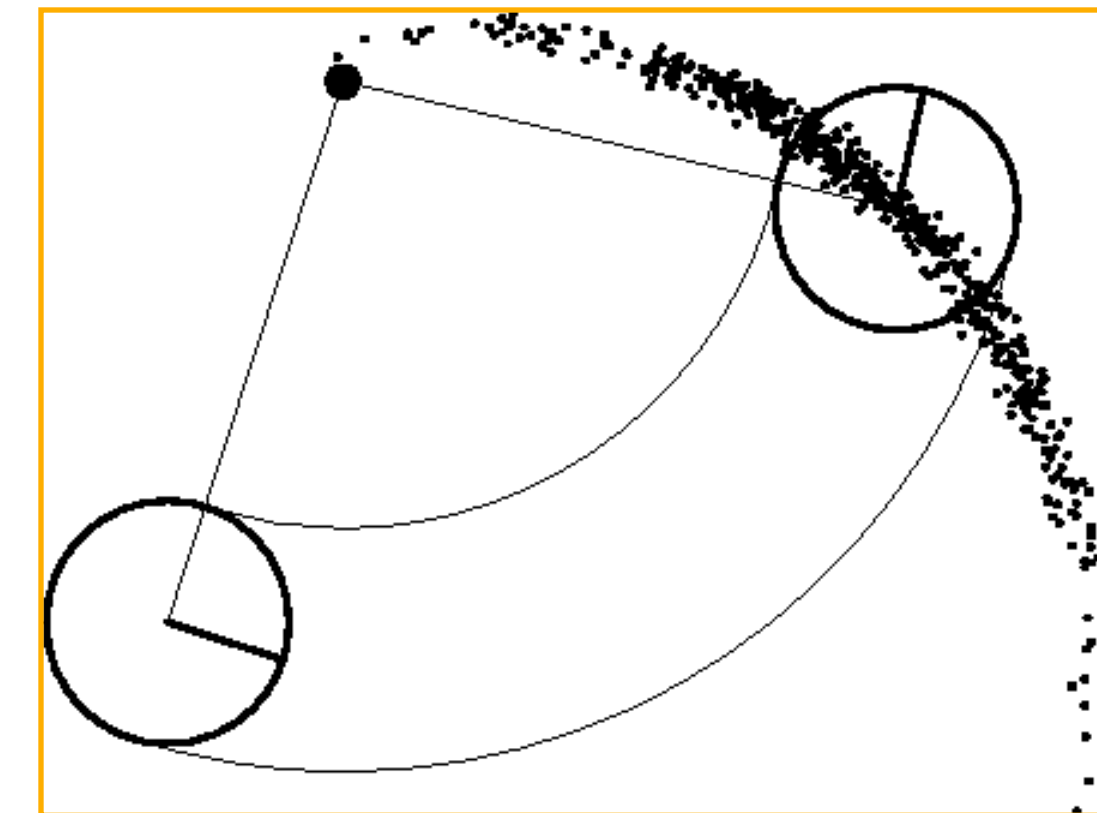
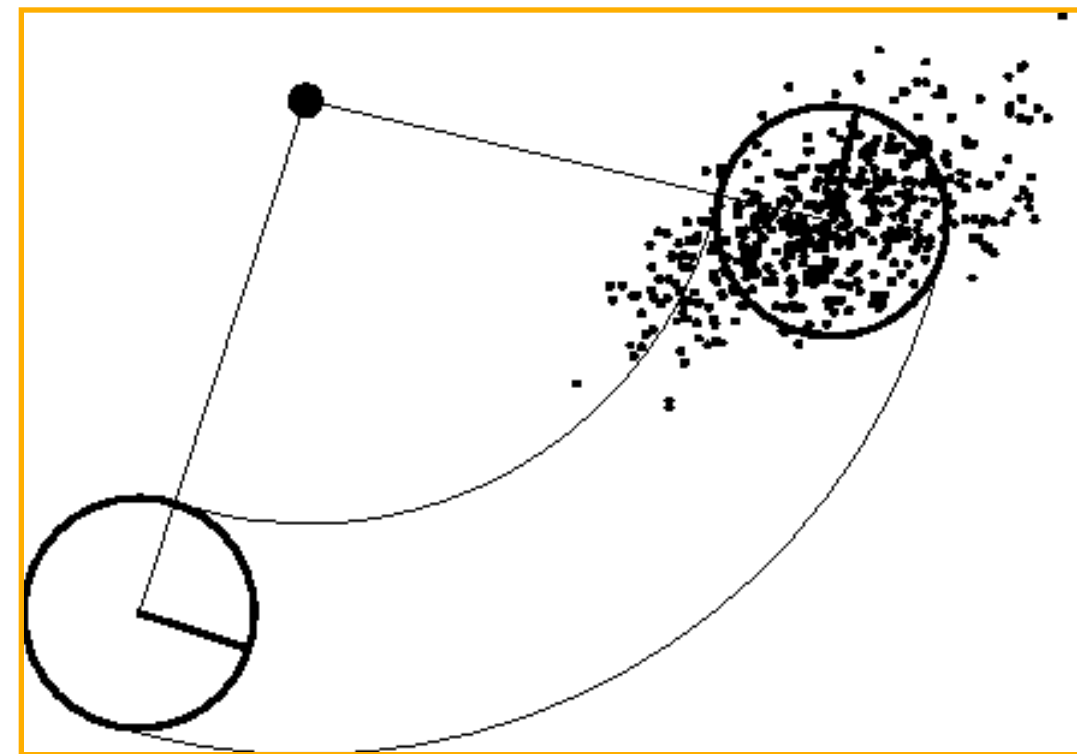
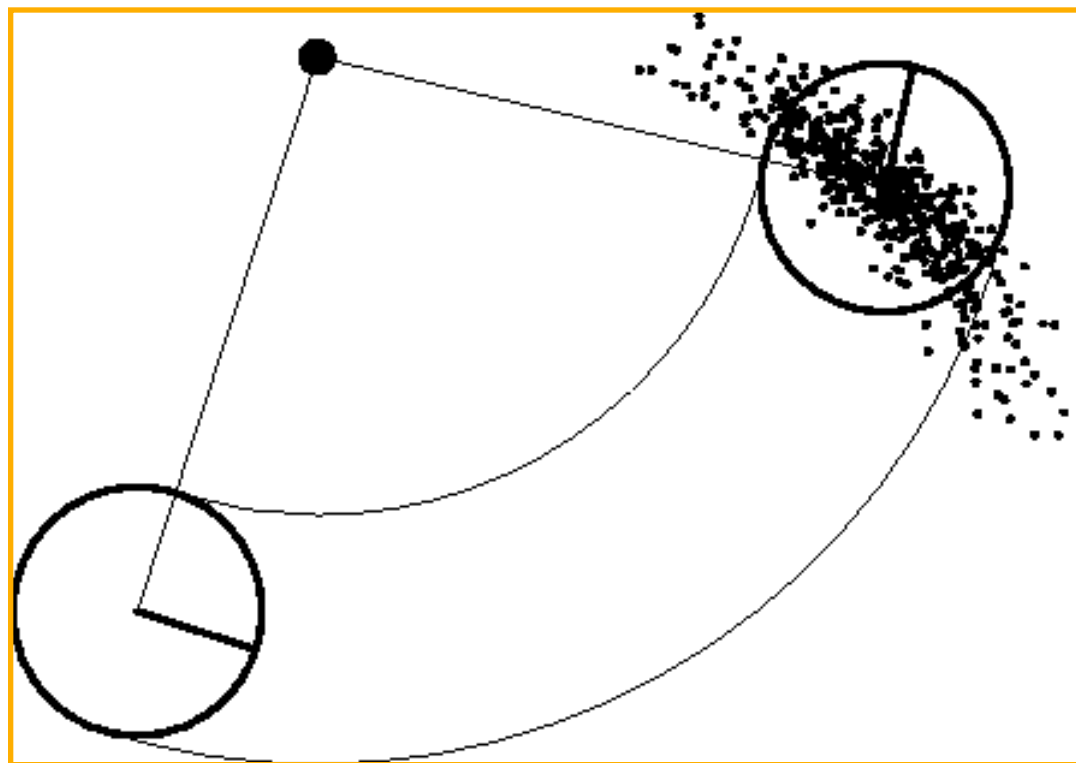
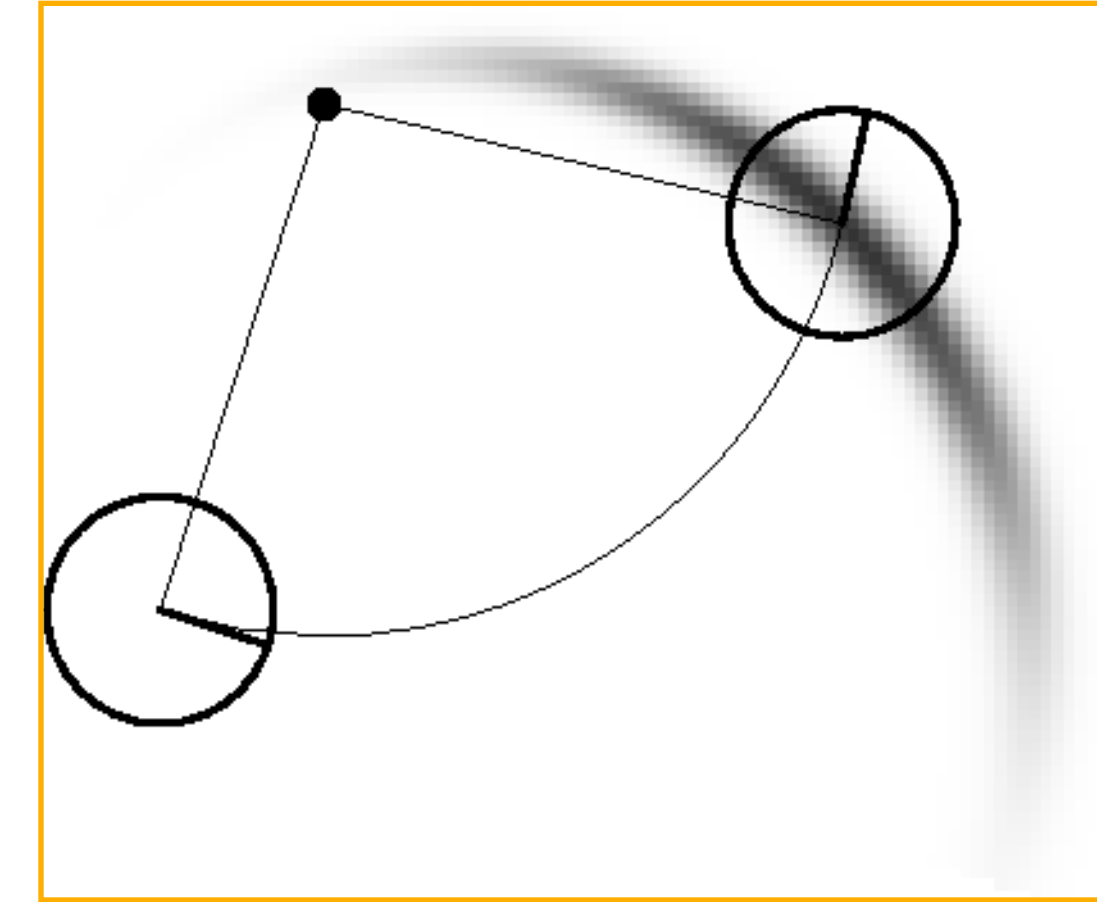
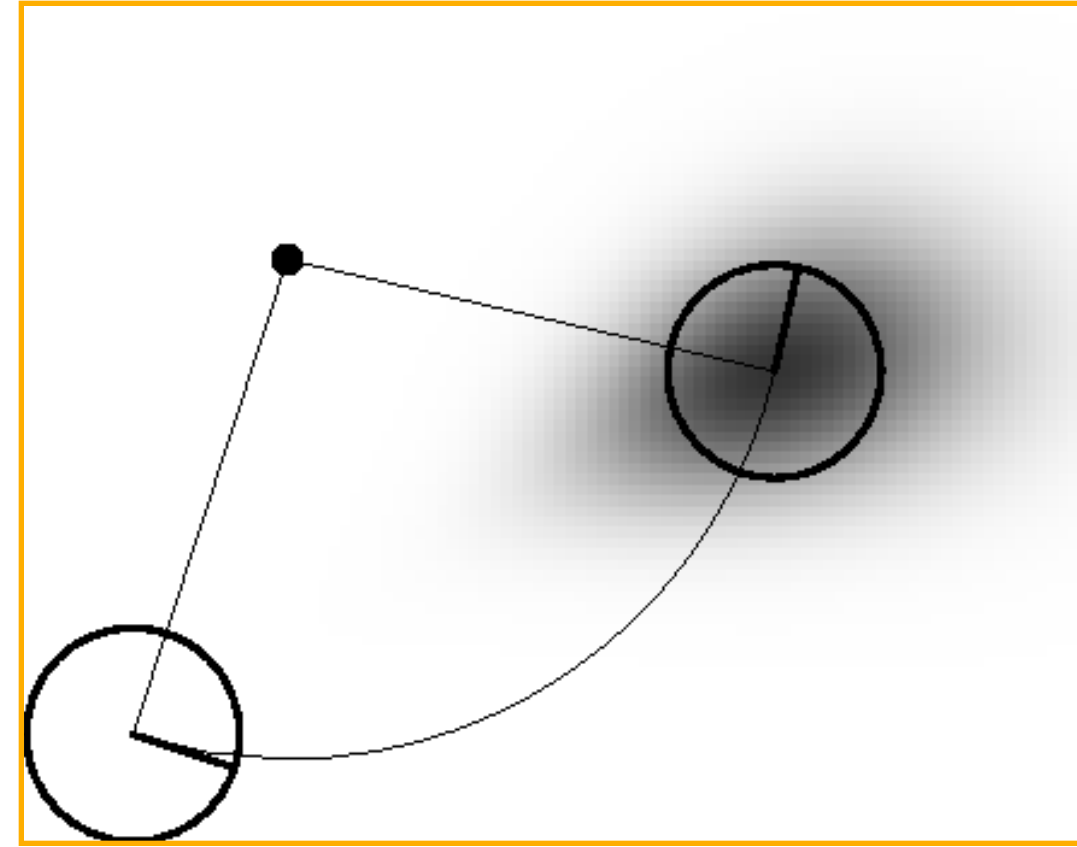
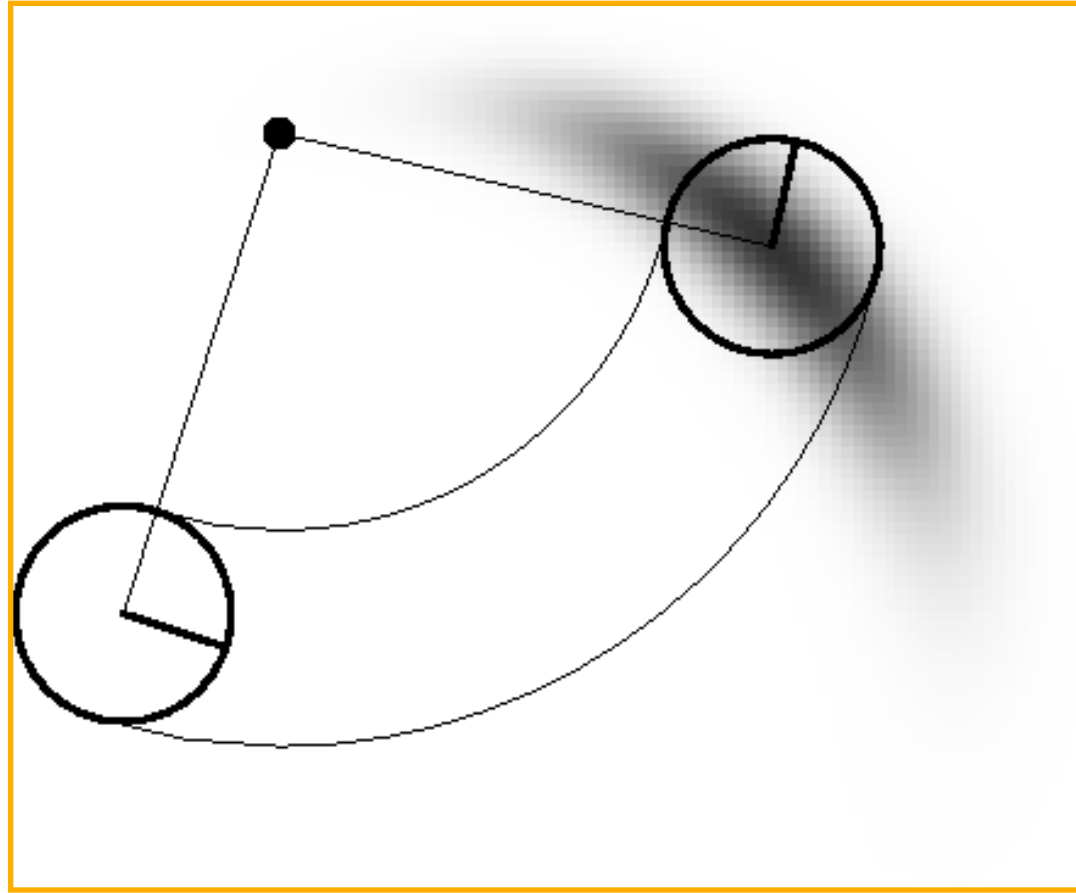
1. $\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 | \delta_{rot1} | + \alpha_2 \delta_{trans})$
2. $\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|))$
3. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1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



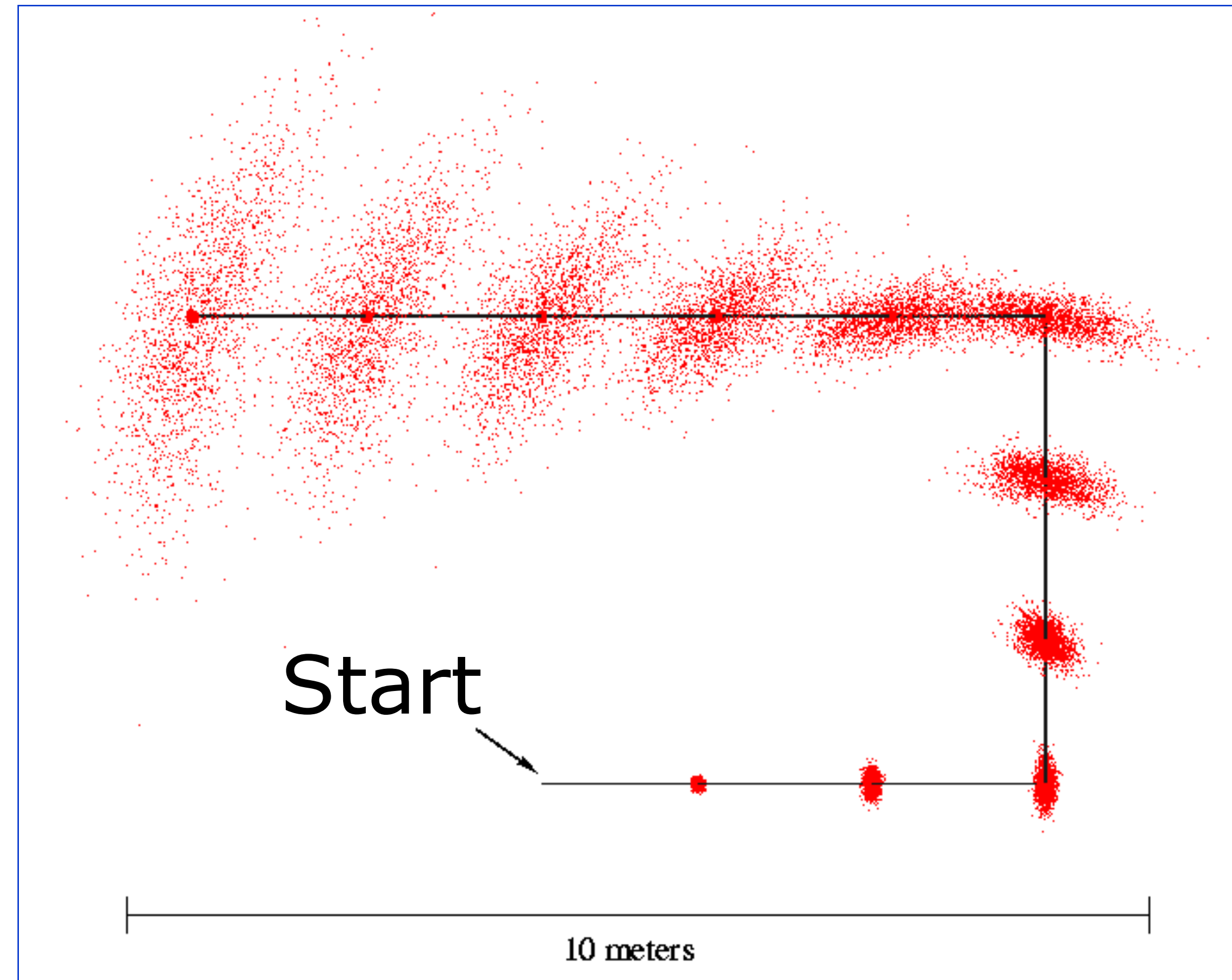
Examples (odometry based)



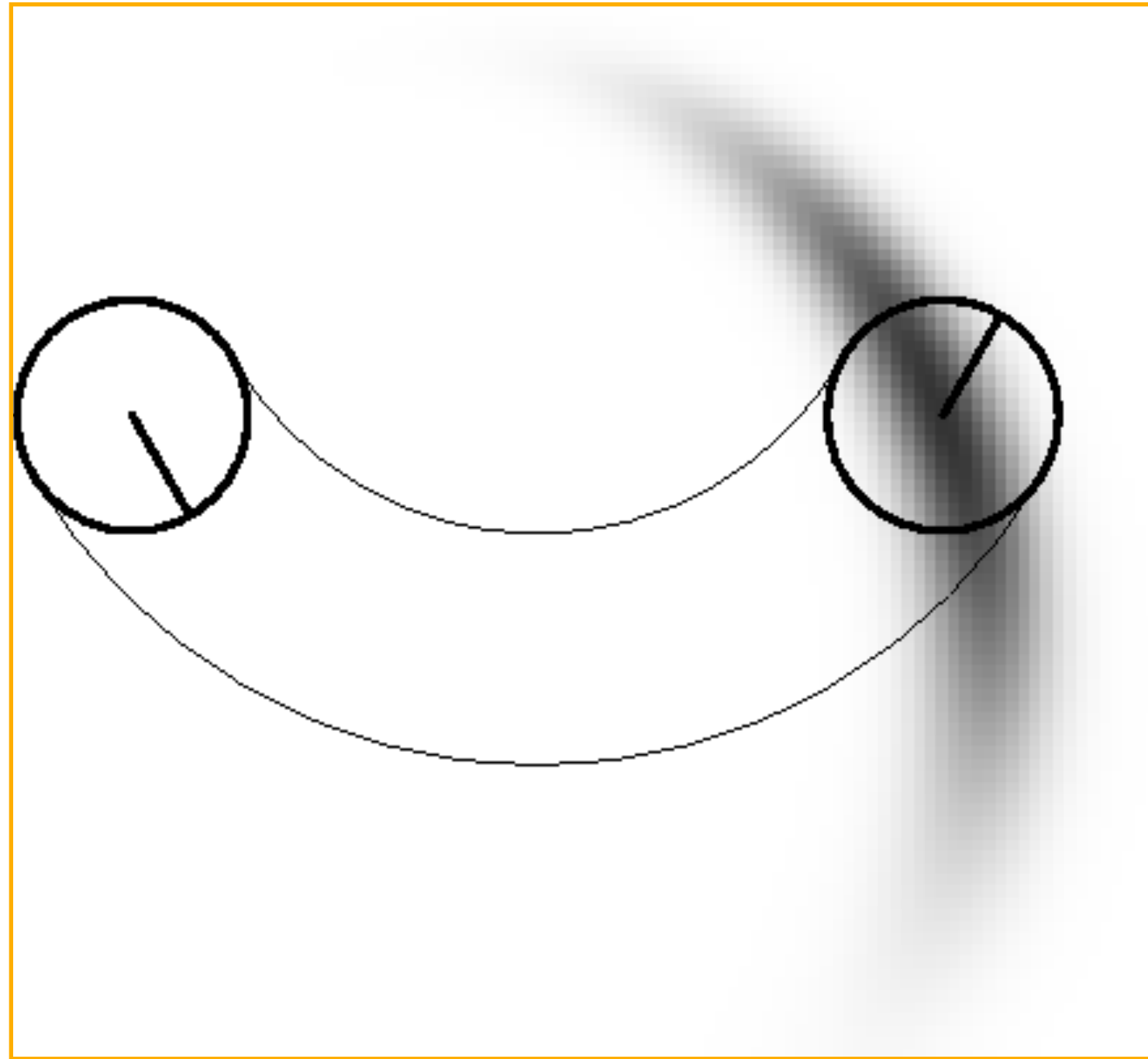
Examples (velocity based)



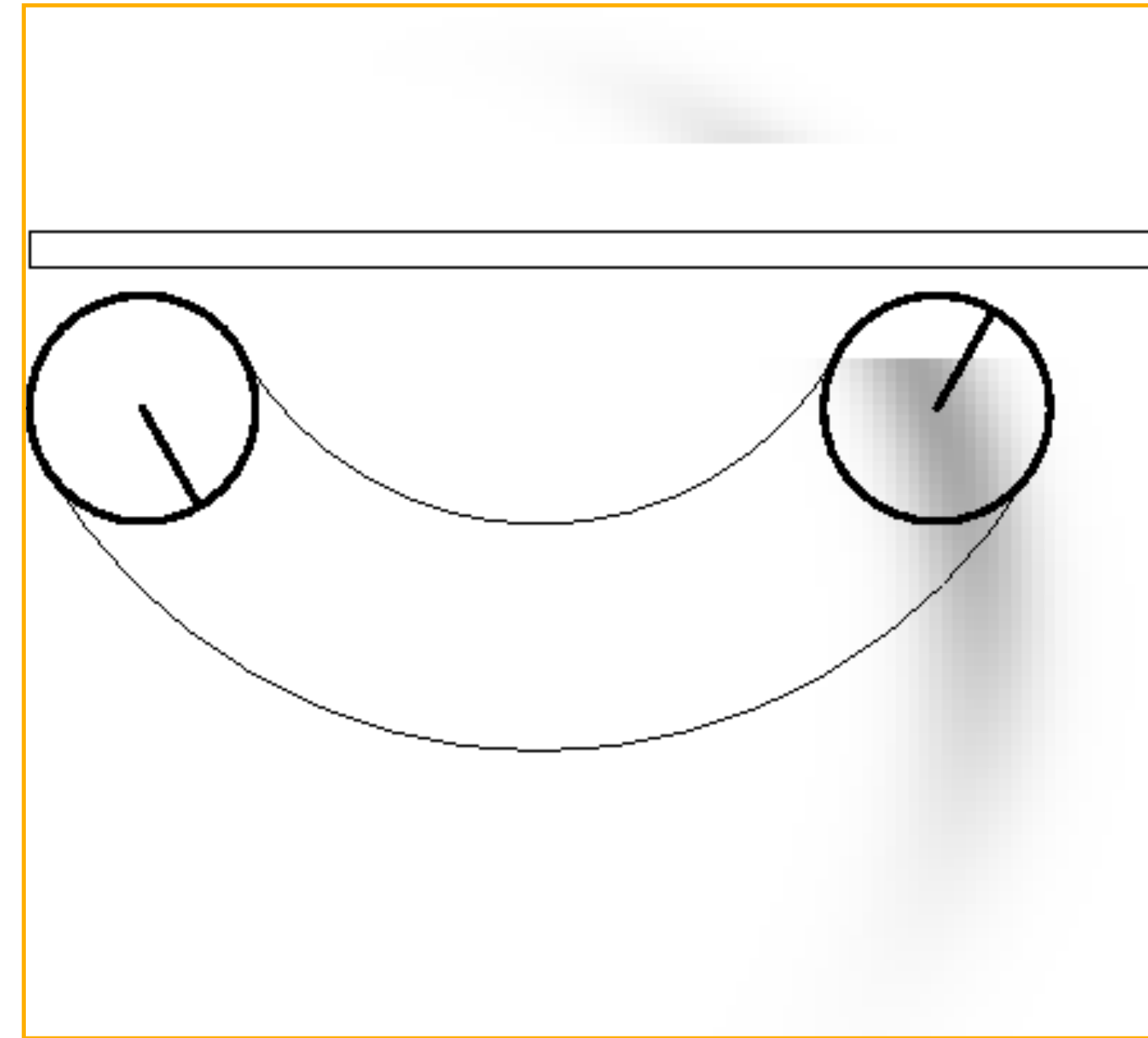
Sample-based Motion



Motion Model with Map



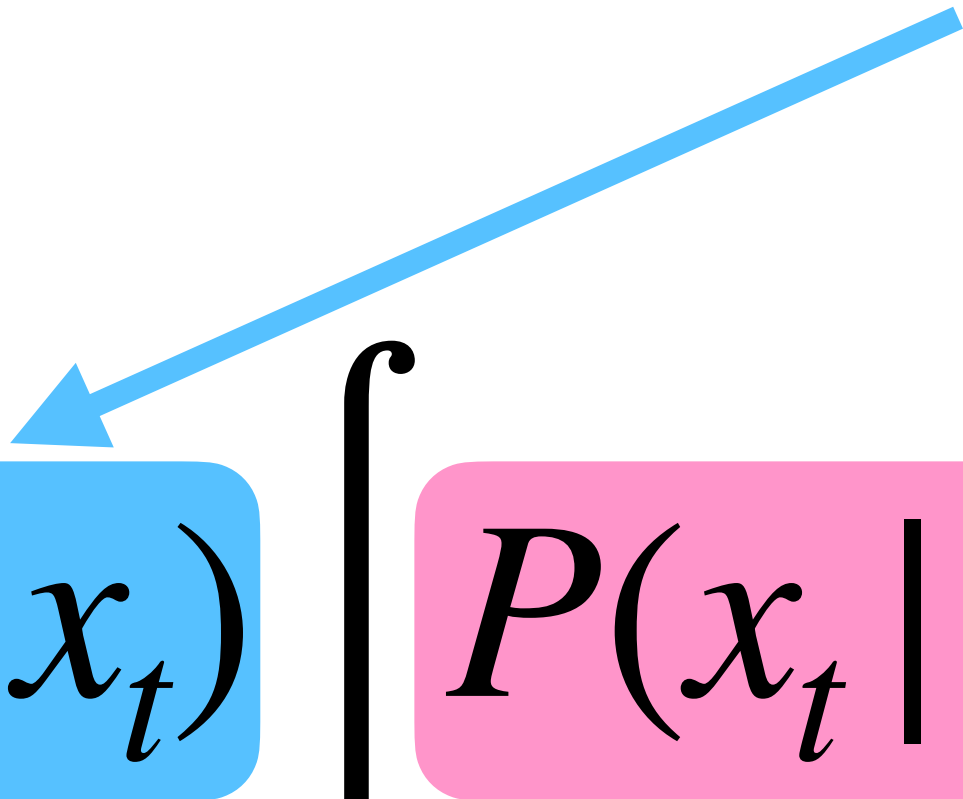
$$P(x | u, x')$$



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

- When does this approximation fail?

Probabilistic Sensor Models

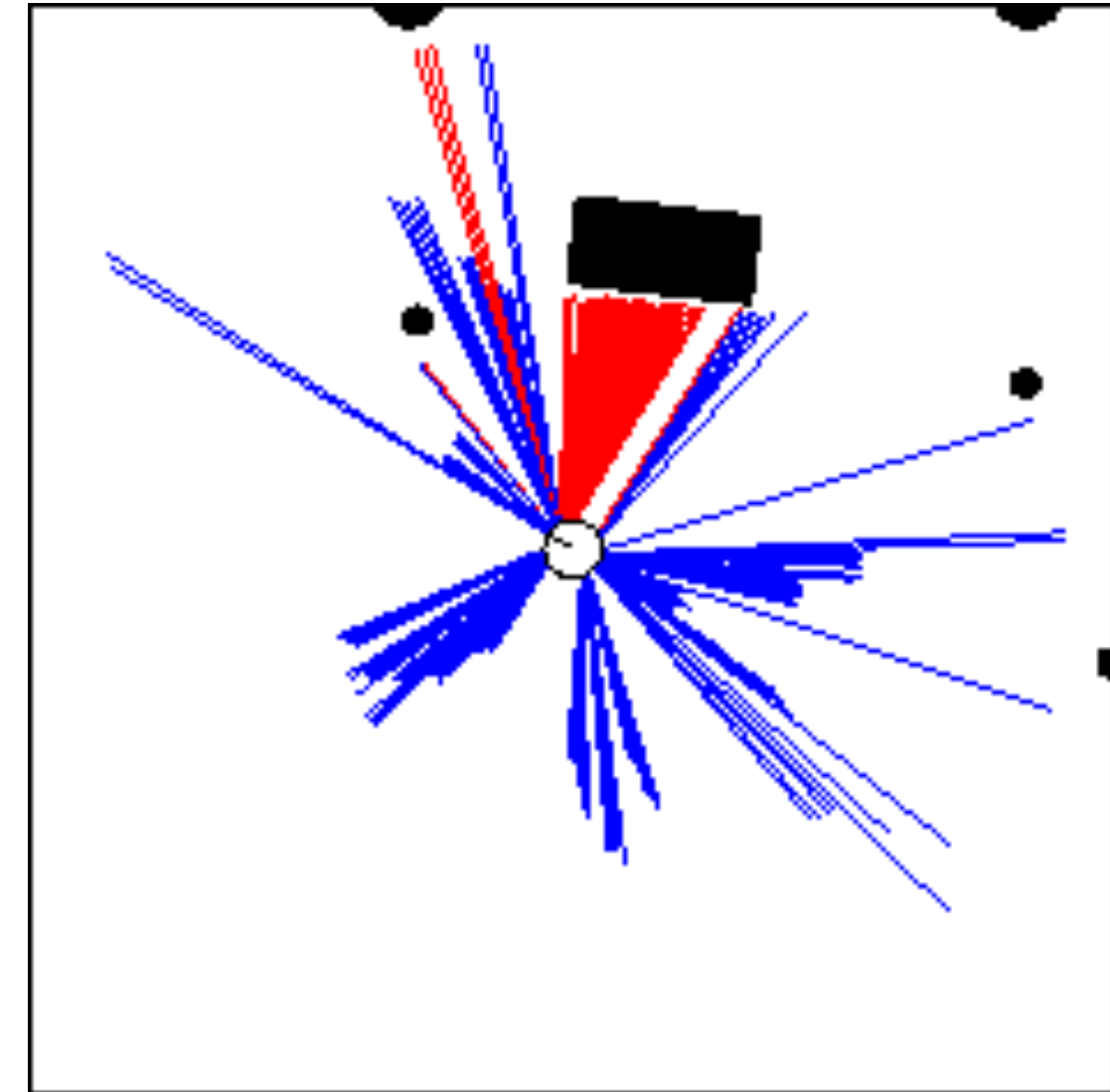
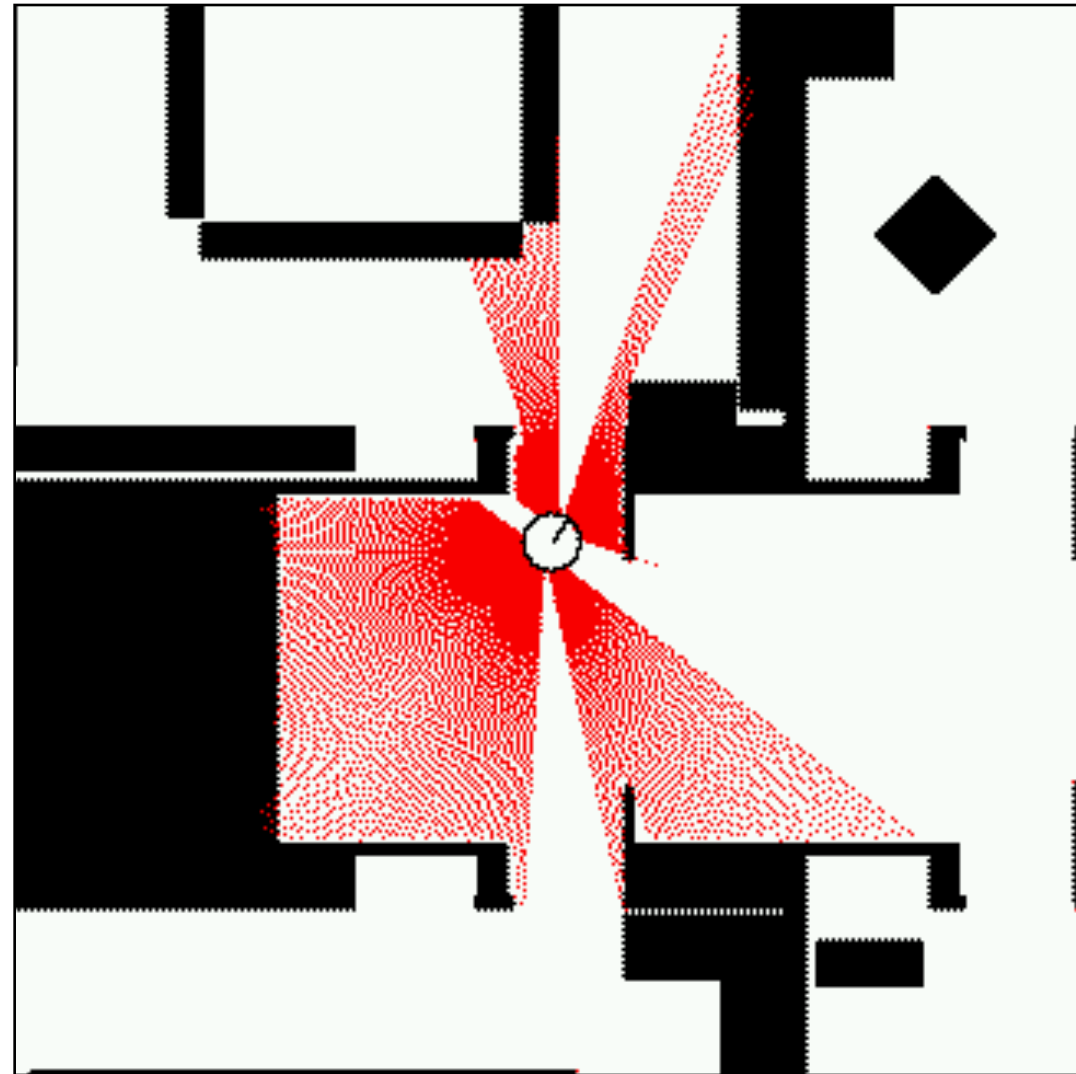
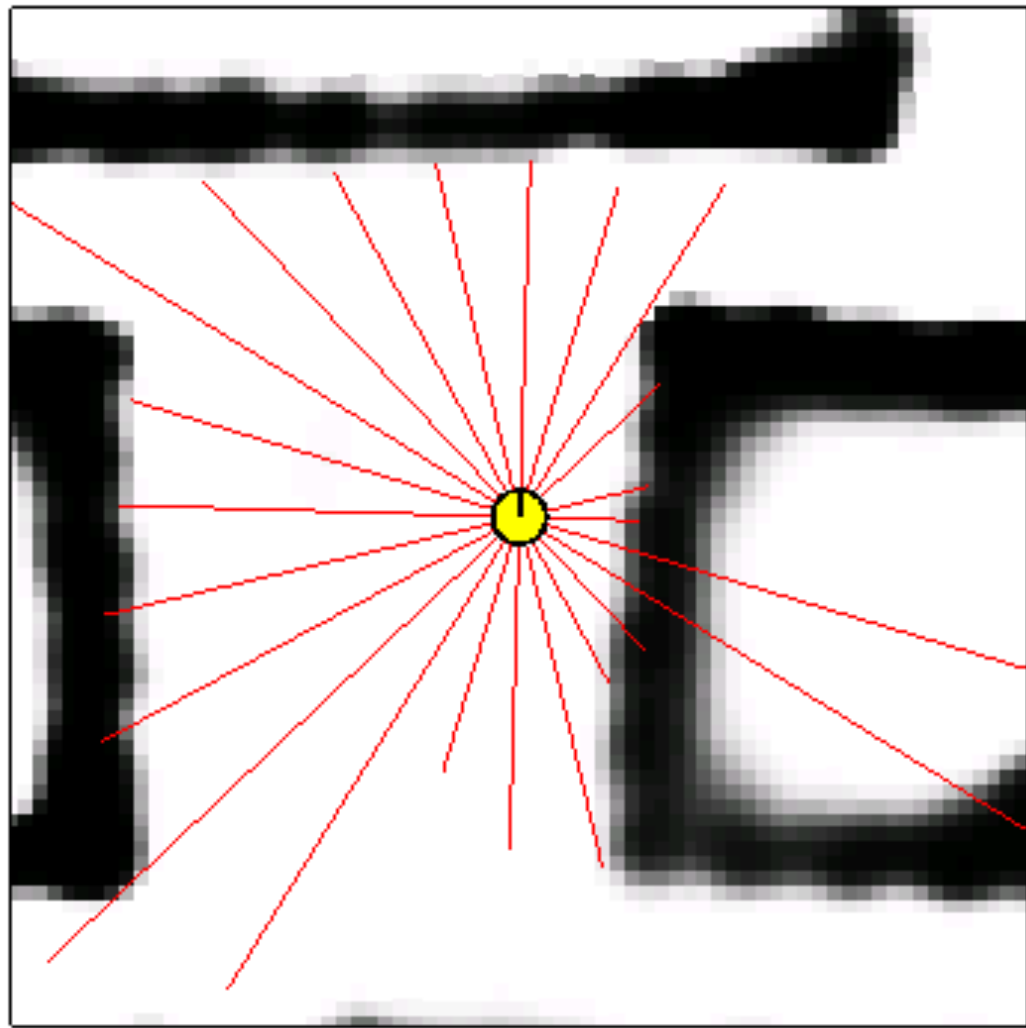
$$Bel(x_t) = \eta P(z_t | x_t) \int P(x_t | x_{t-1} u_t) Bel(x_{t-1}) dx_{t-1}$$


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.

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



Beam-based Sensor Model

- Scan z consists of K measurements.

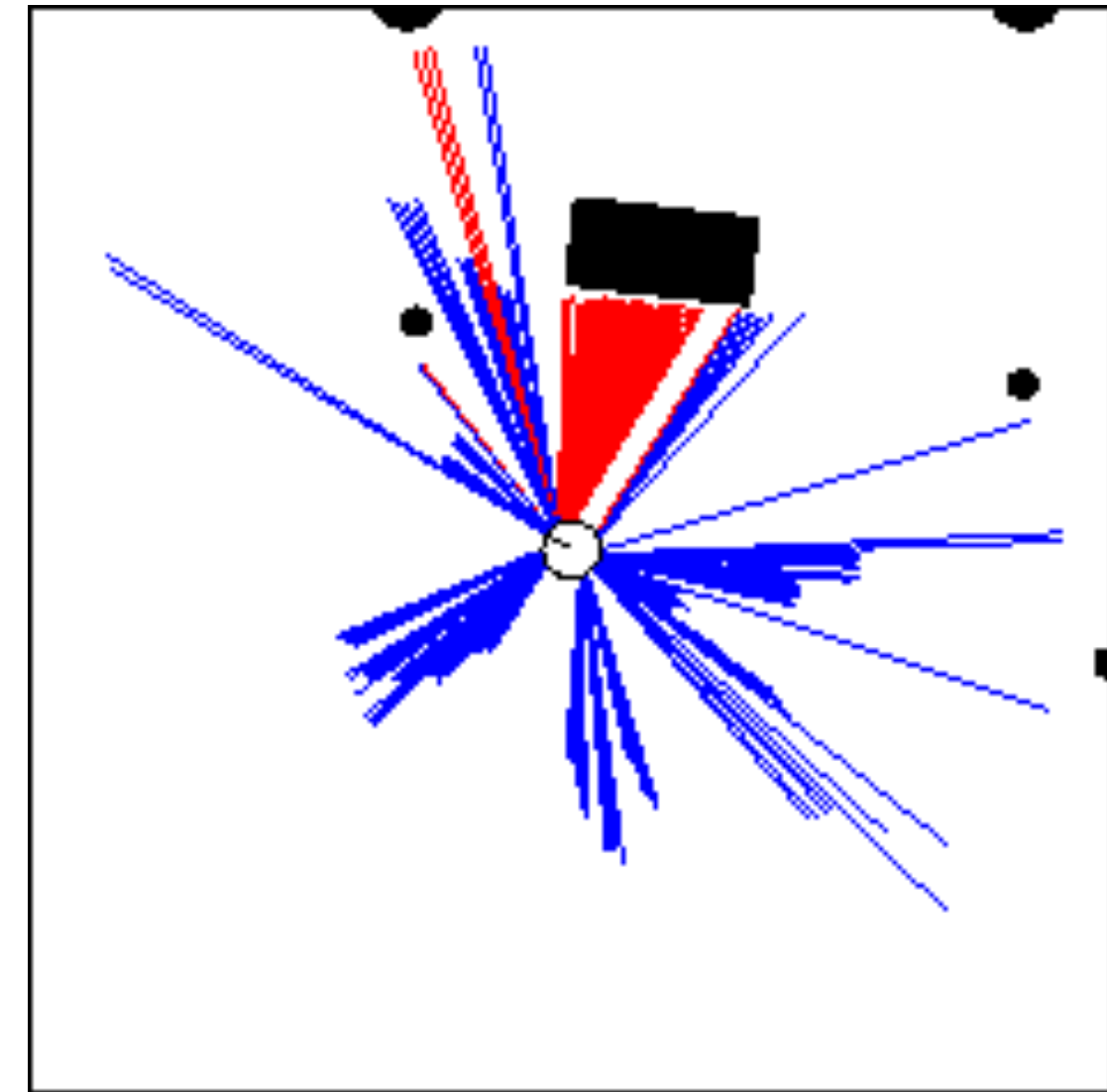
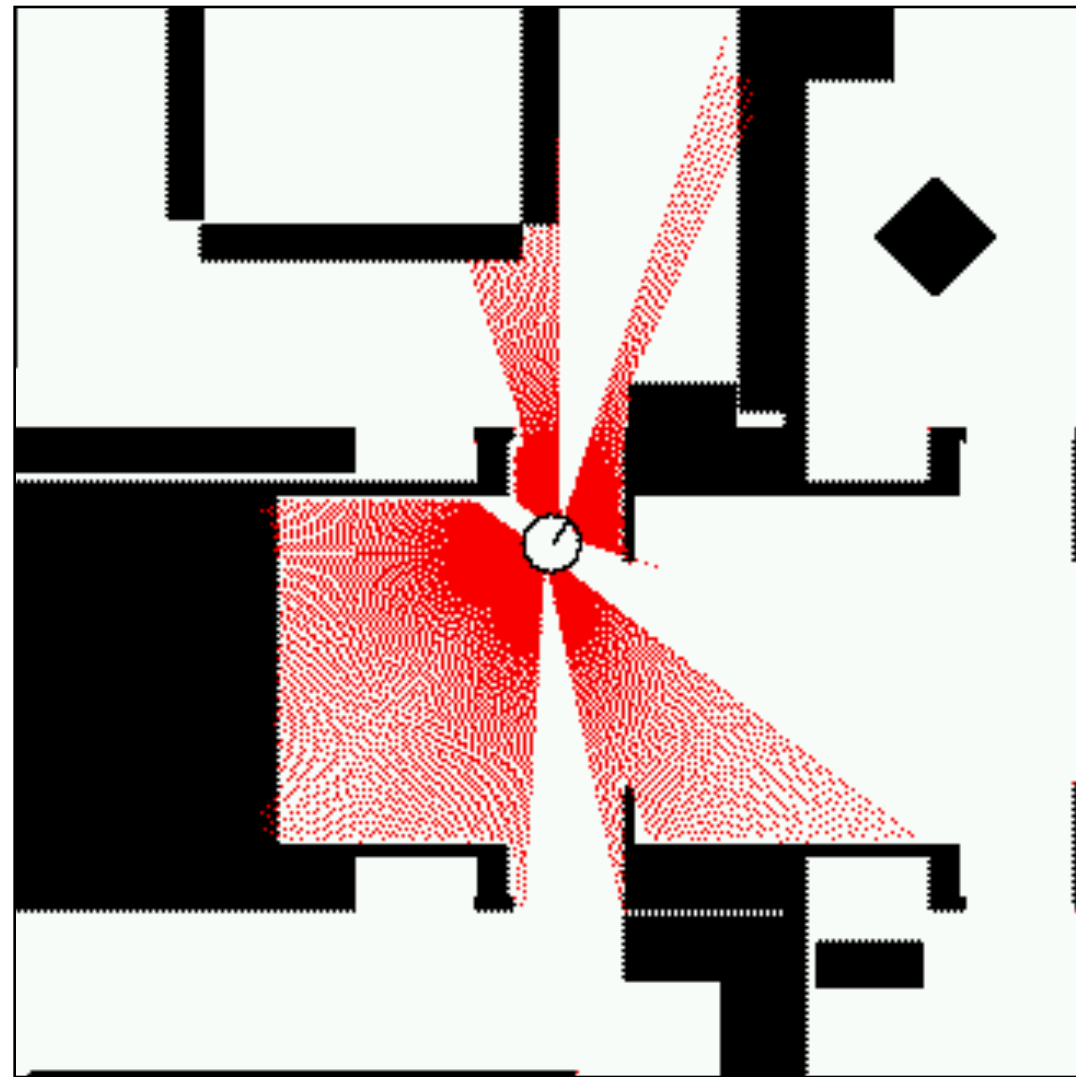
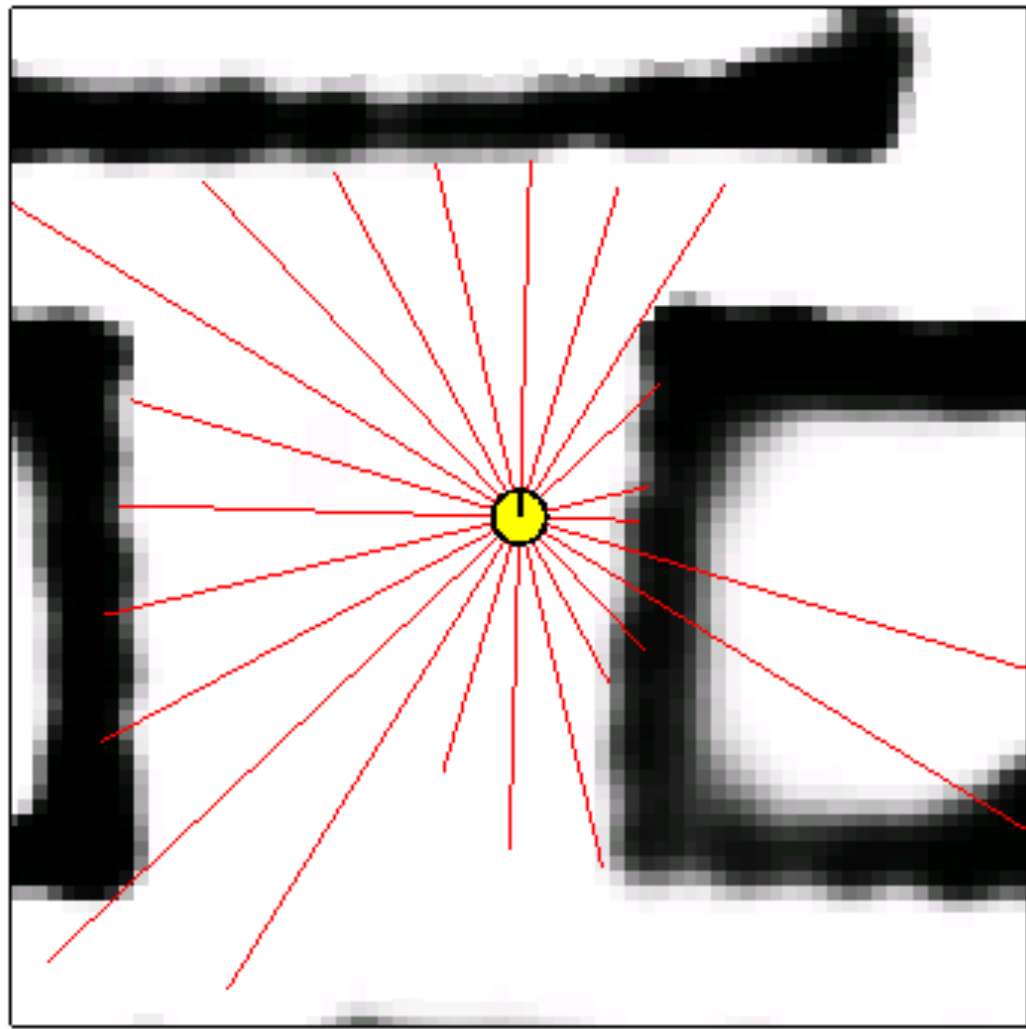
$$z = \{z_1, z_2, \dots, z_K\}$$

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

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



Beam-based Sensor Model



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

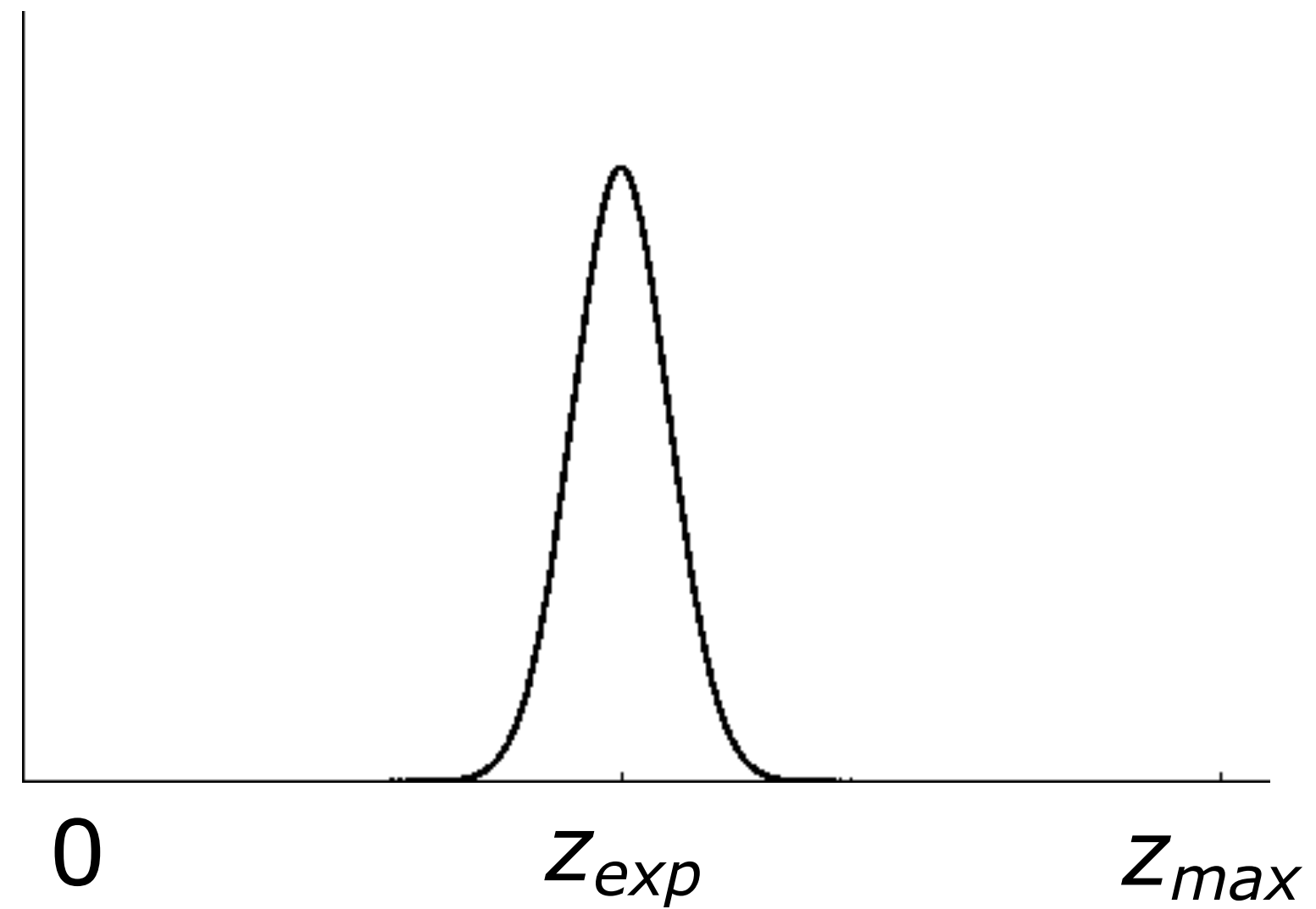
Proximity Measurement

- 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

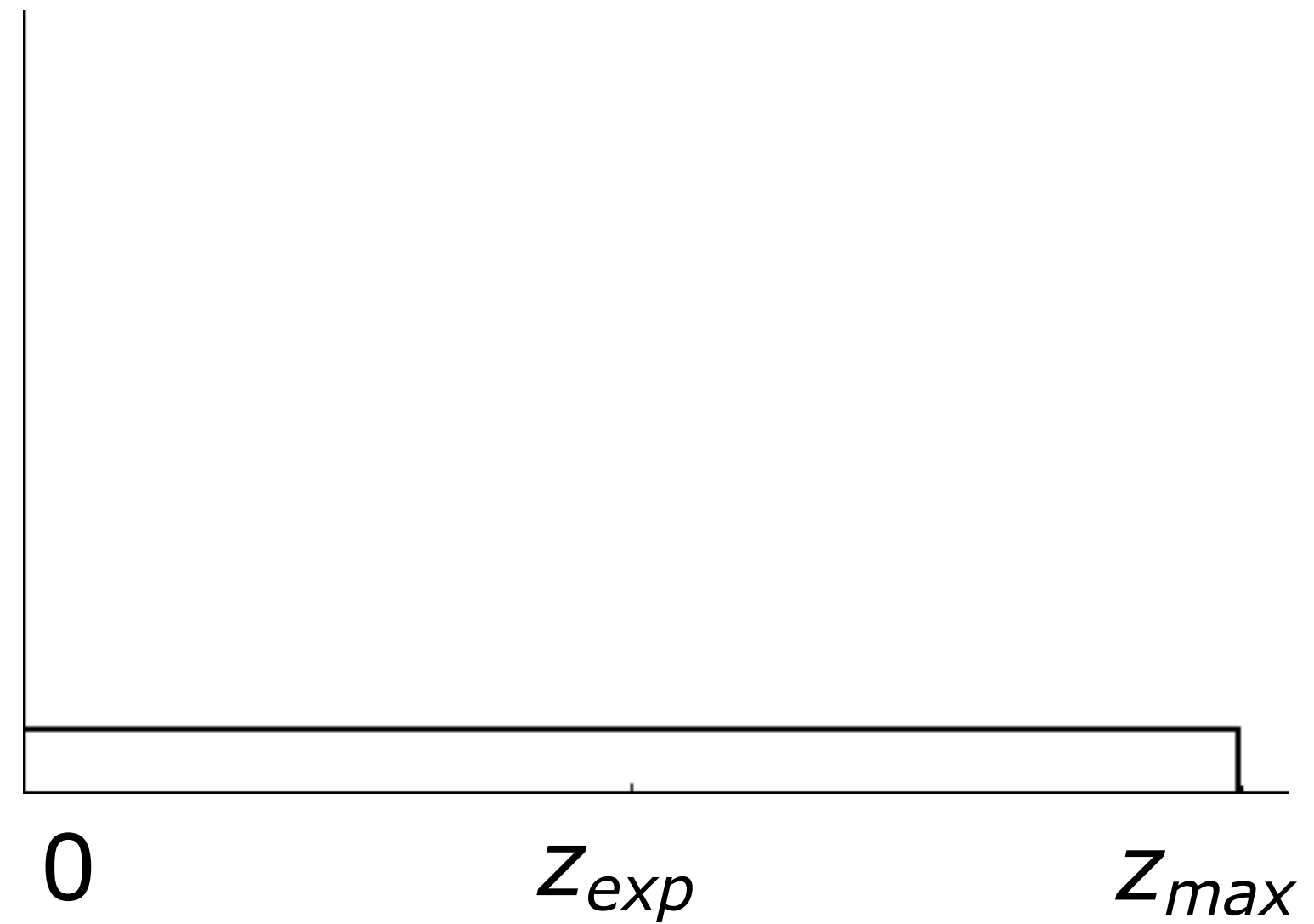
Measurement noise



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

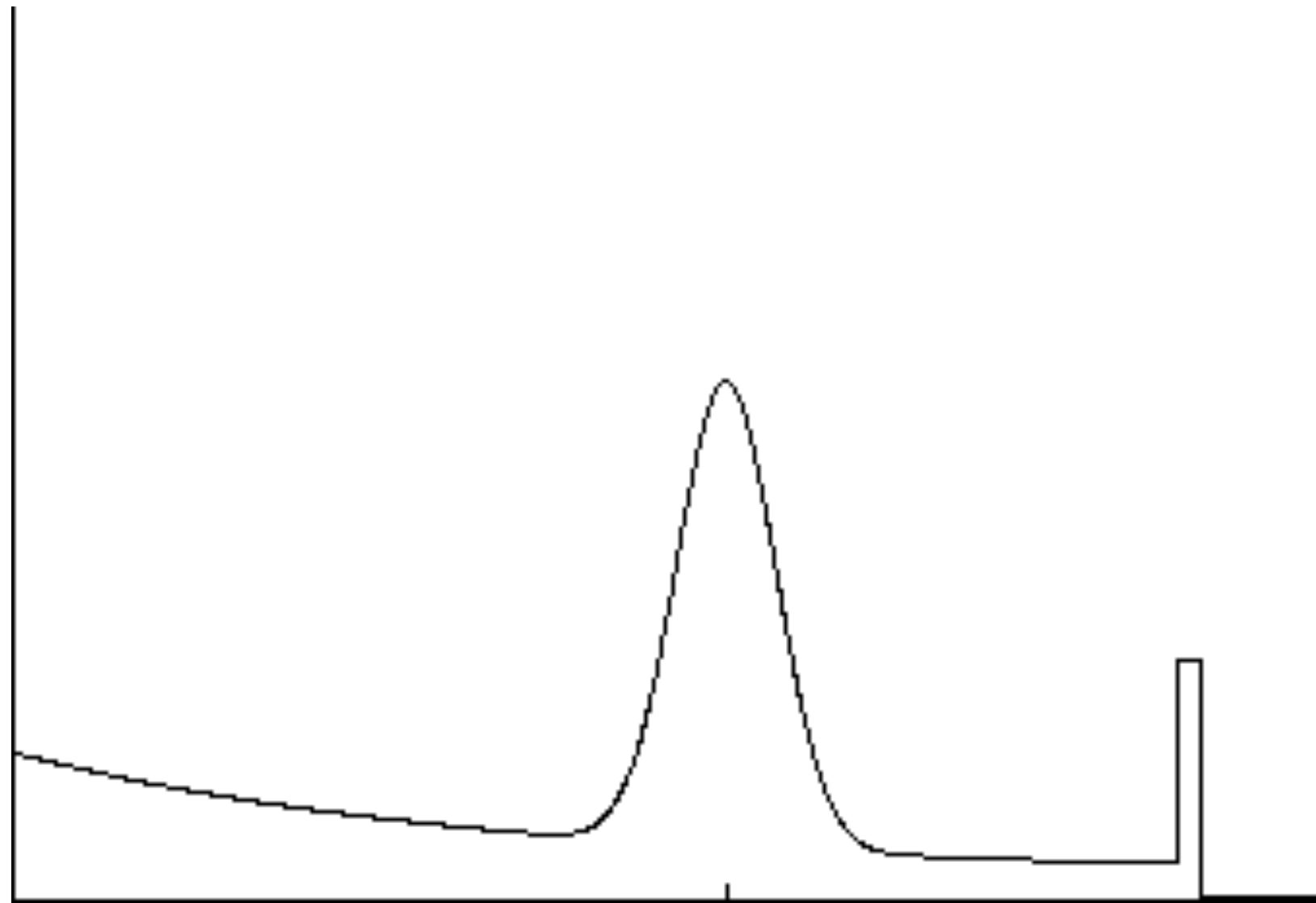
Beam-based Proximity Model

Random measurement



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

Mixture Density



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

How can we determine the model parameters?

Approximation

- Maximize log likelihood of the data z :

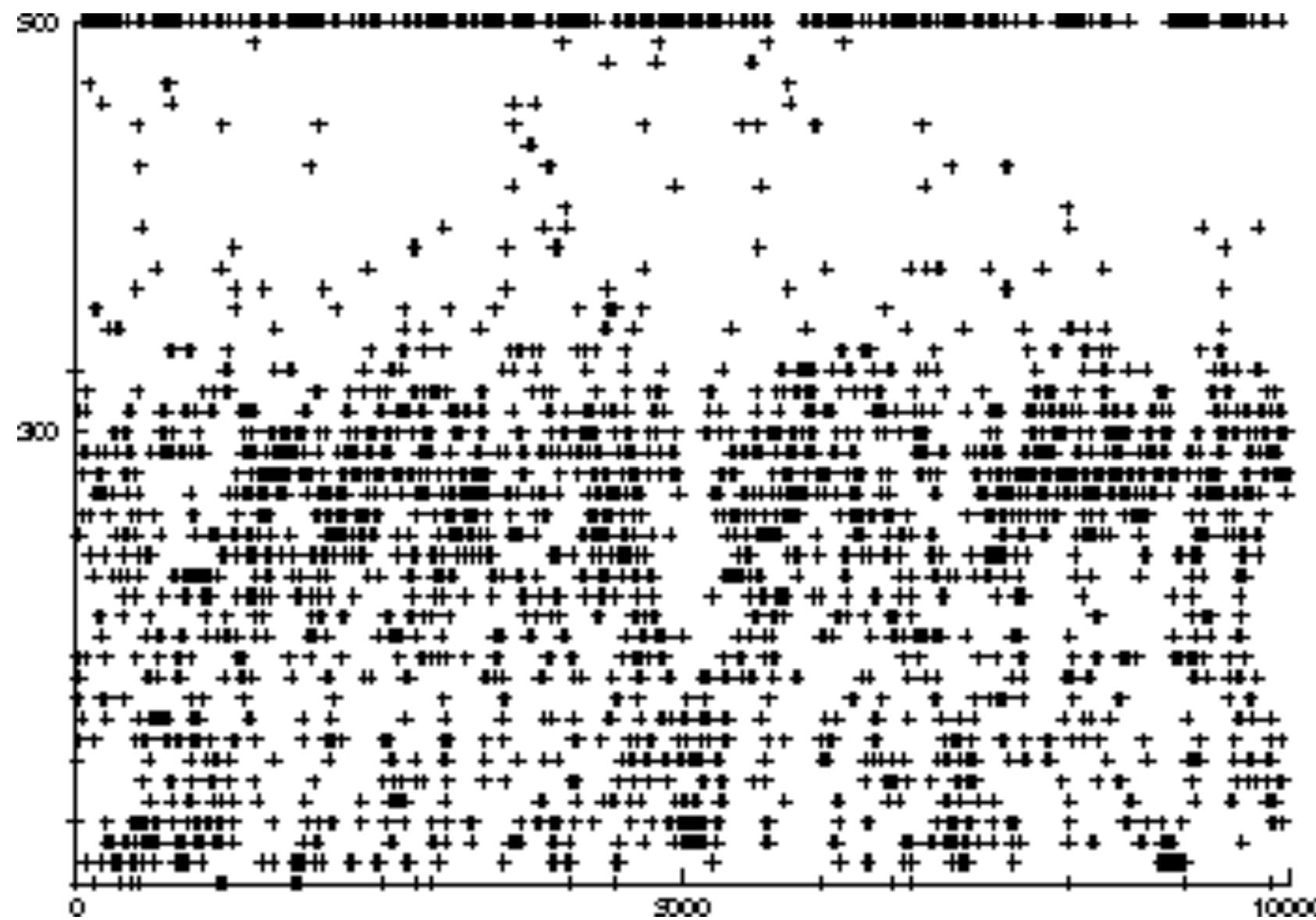
$$P(z \mid z_{\text{exp}})$$

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

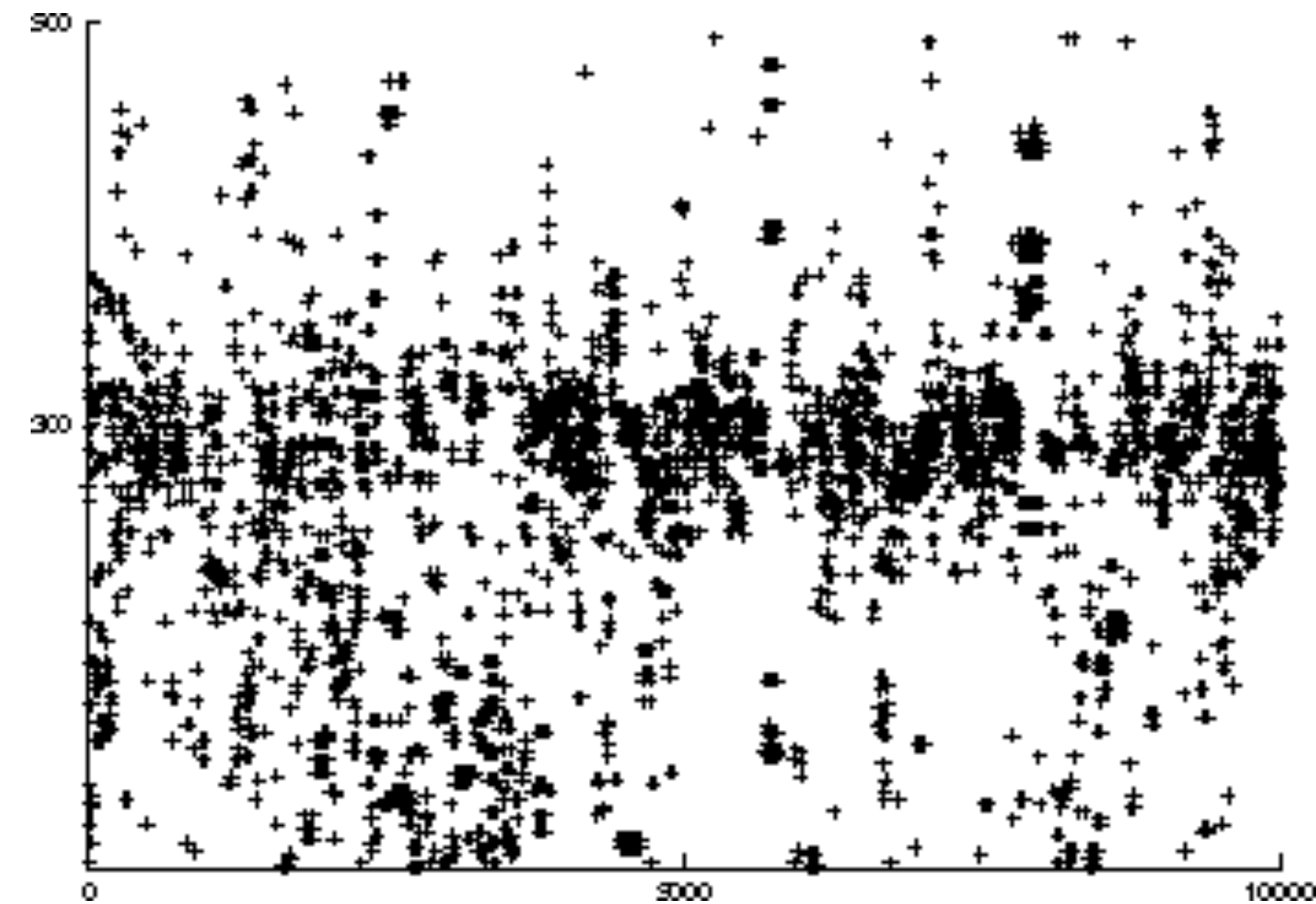


Raw Sensor Data

Measured distances for expected distance of 300 cm.

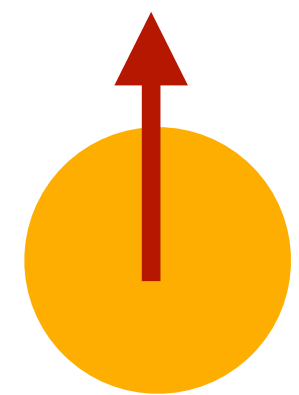


Sonar

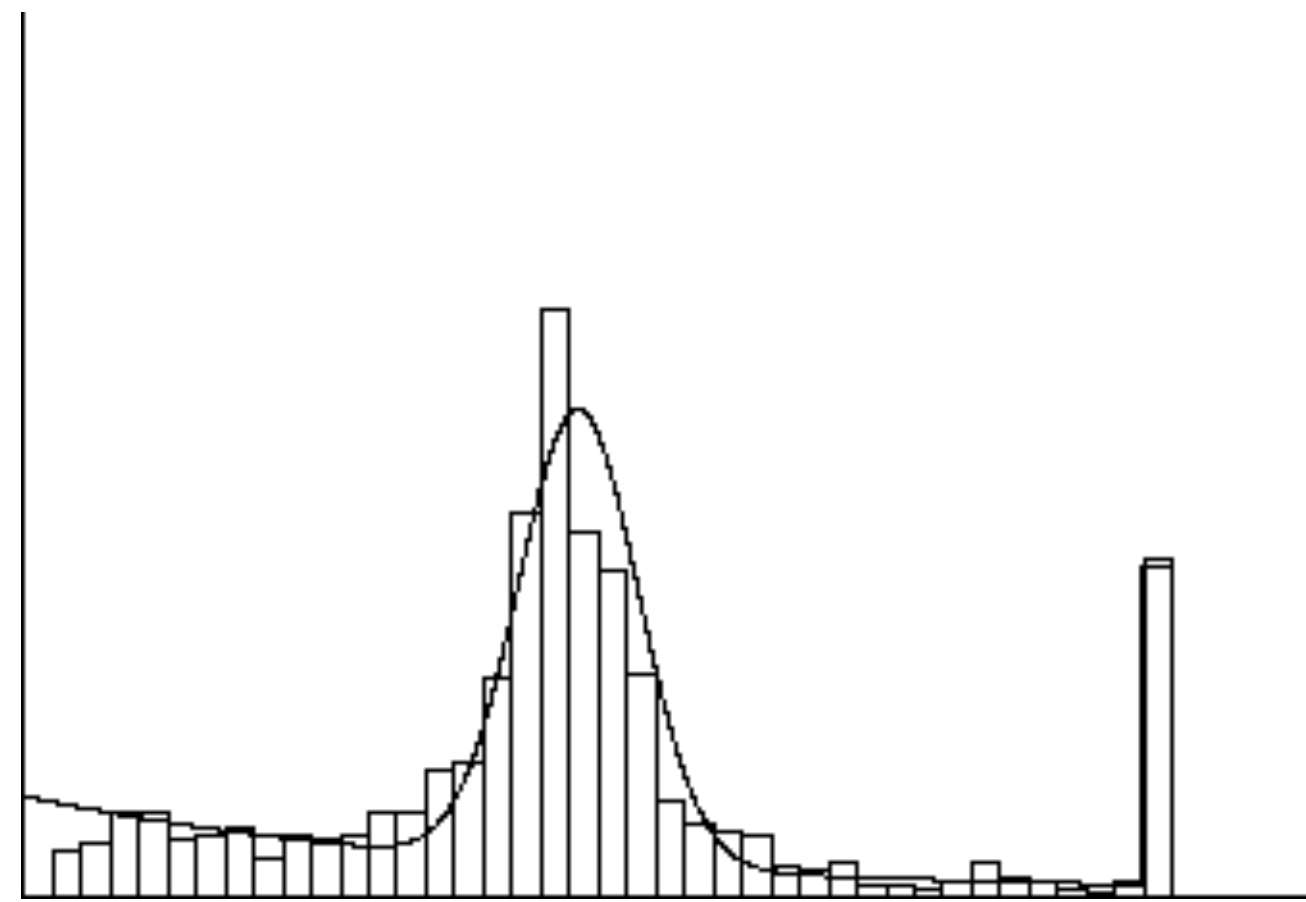


Laser

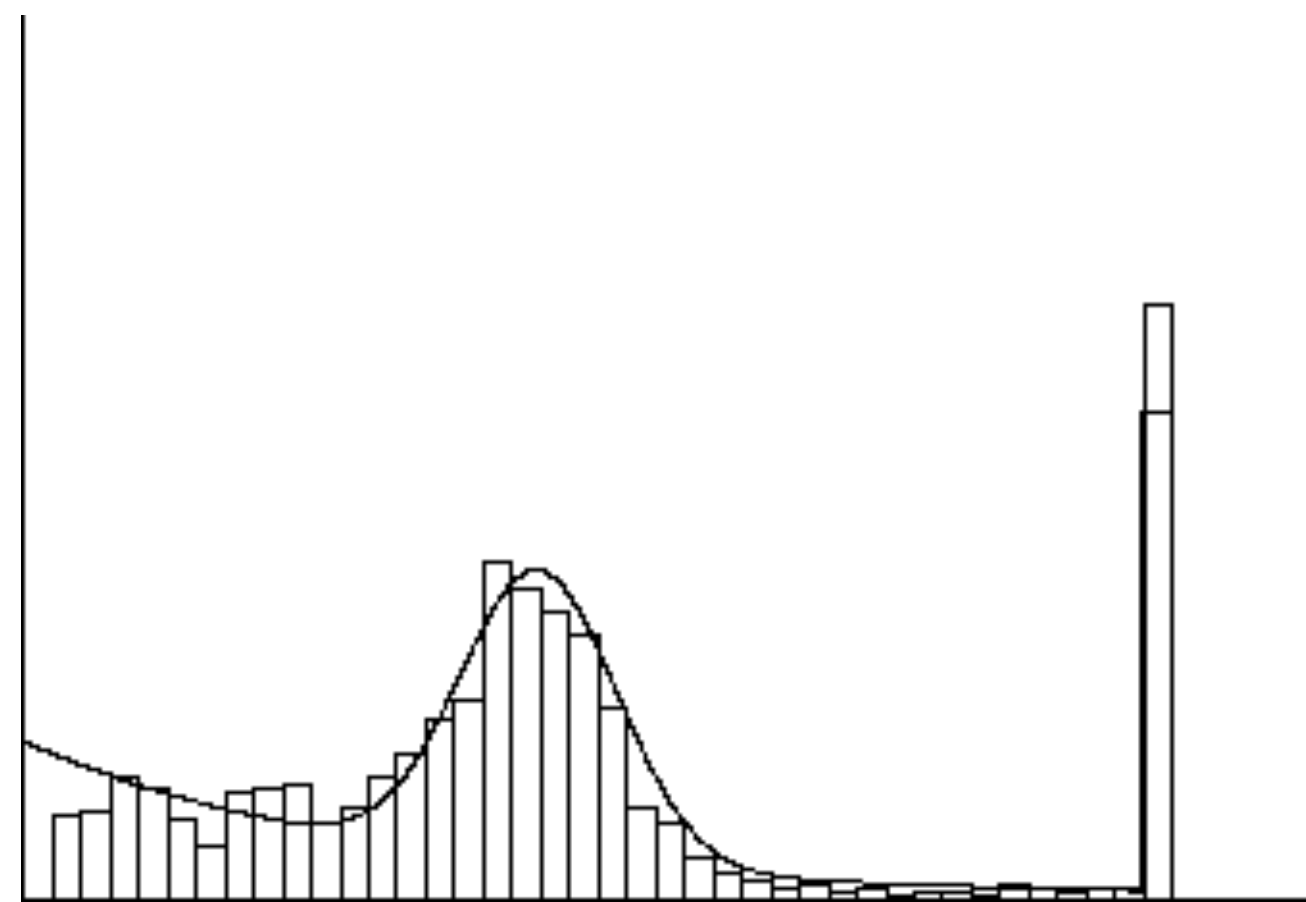
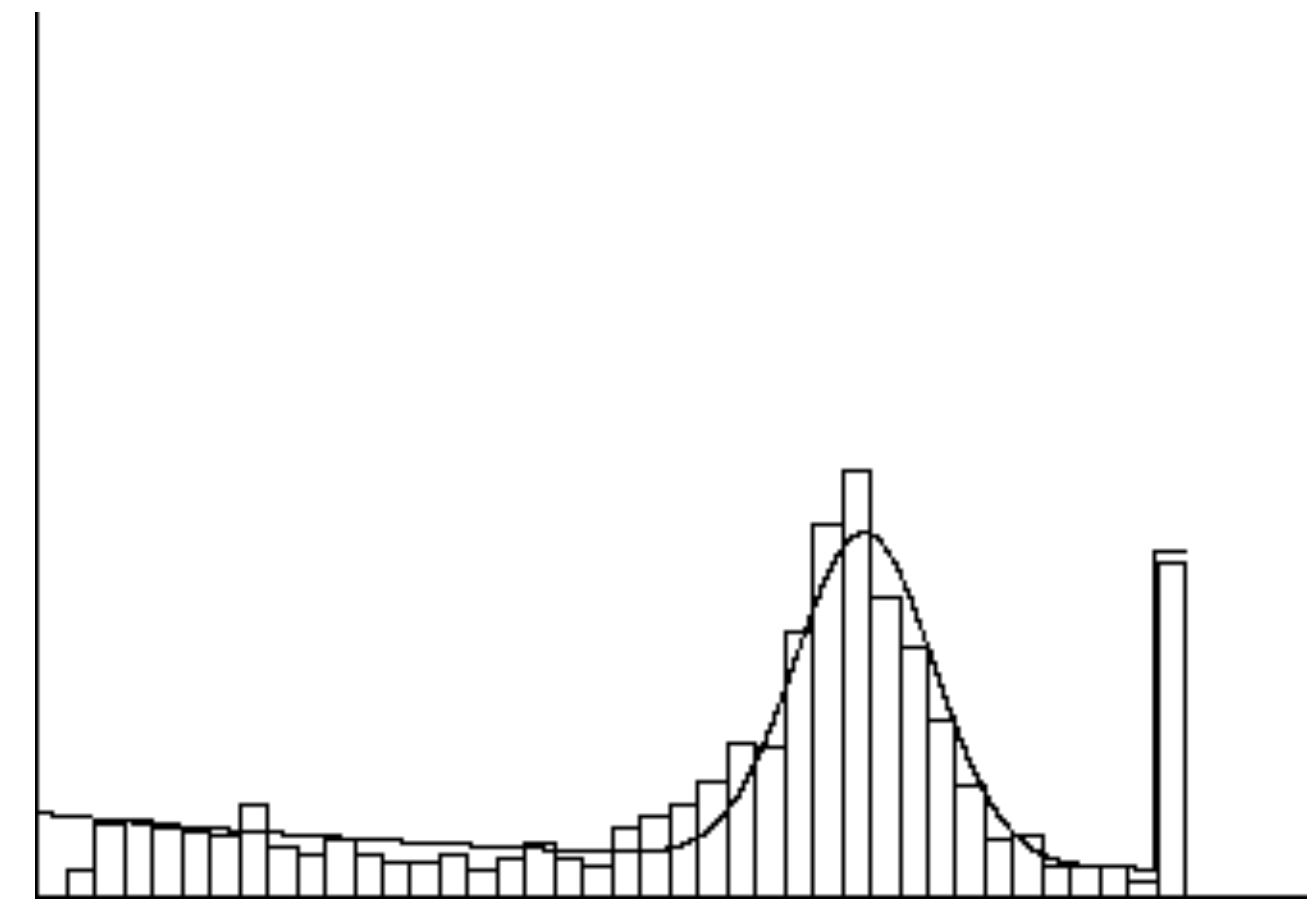
of trials



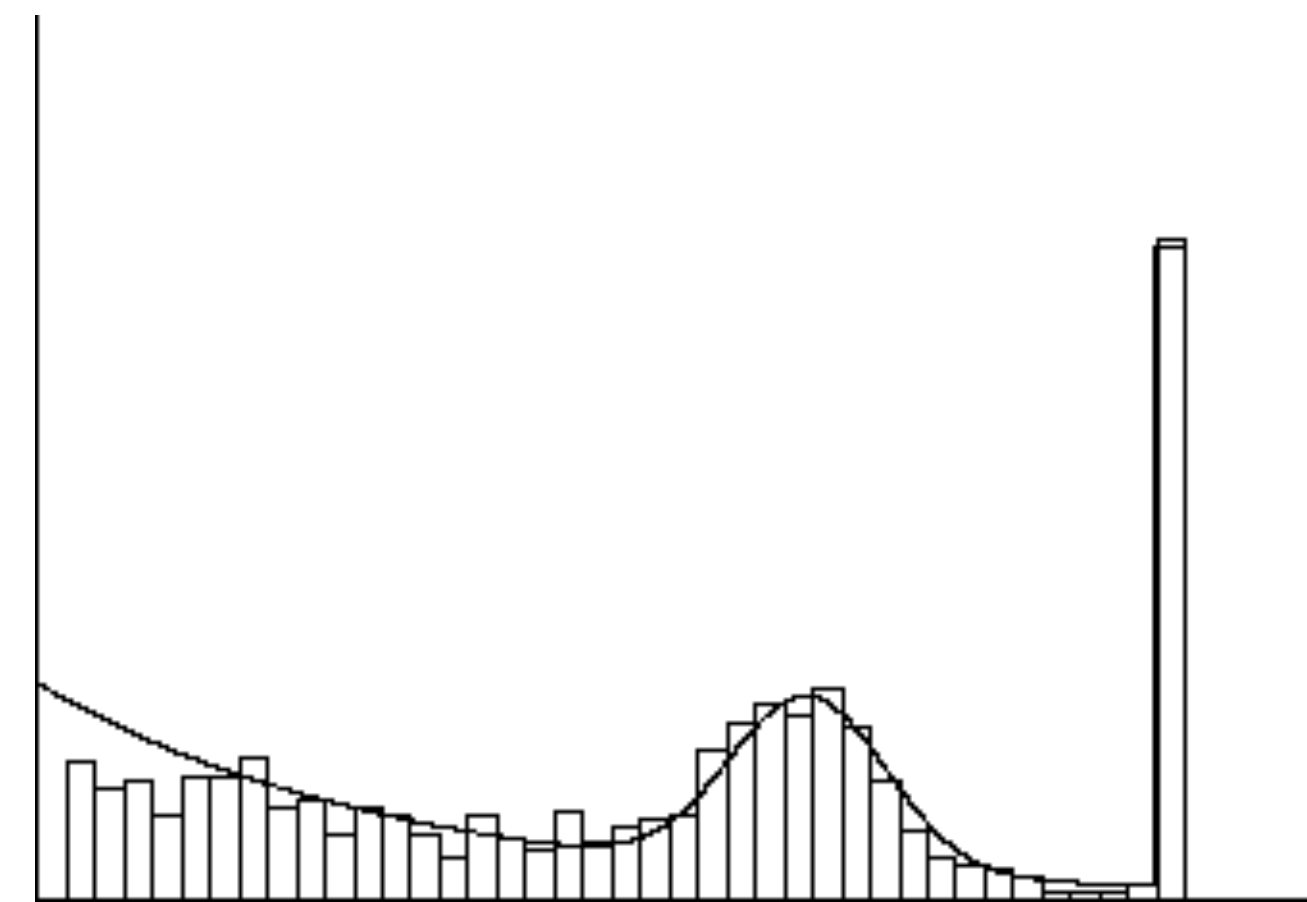
Approximation Results



Laser



Sonar



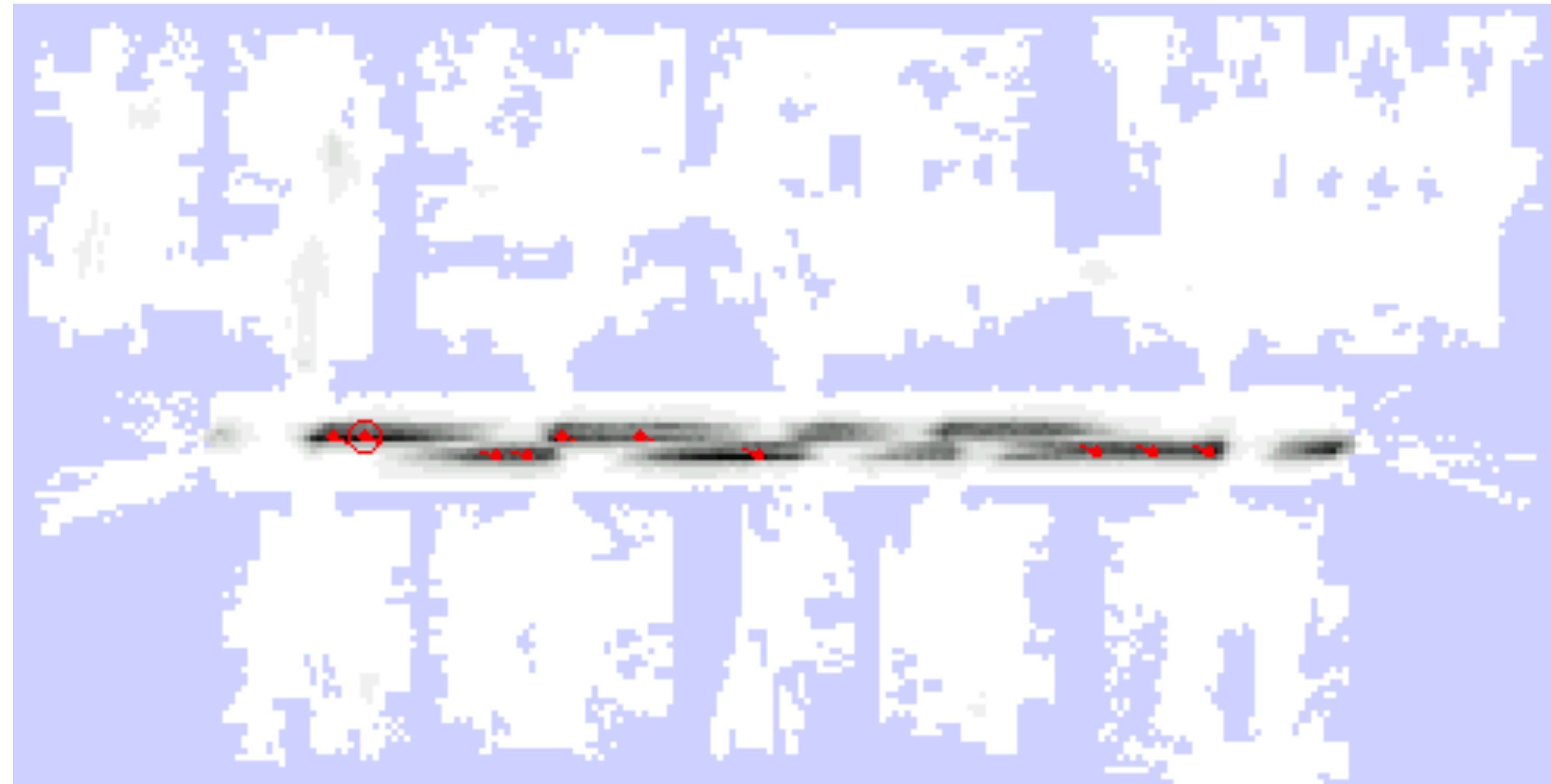
300cm

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 Project (Open ended)

Think along these axes to
decide your final project!

Evaluating your
implementation/system with
quantitative results are **VERY**
important!

Long horizon tasks

Tasks

Objects

Rearrangment of a set of objects

Multi-robot task execution

Robots



Final Project (Open ended)

Think along these axes to decide your final project!

Evaluating your implementation/system with quantitative results are **VERY** important!

Objects

Rearrangement of a set of objects

You may use:

- Kineval codebase
- Other sim environments (*pybullet, Gazebo, DRAKE, Isaac sim*)
- Turtlebot3 (*provided only upon compelling proposal, only 5 are available*)
- Other robots you may have access to.

Multi-robot task execution

Long horizon tasks

Tasks

Robots

During the P7 sessions we will show other robotic platforms and sensors that are accessible for the Final Projects