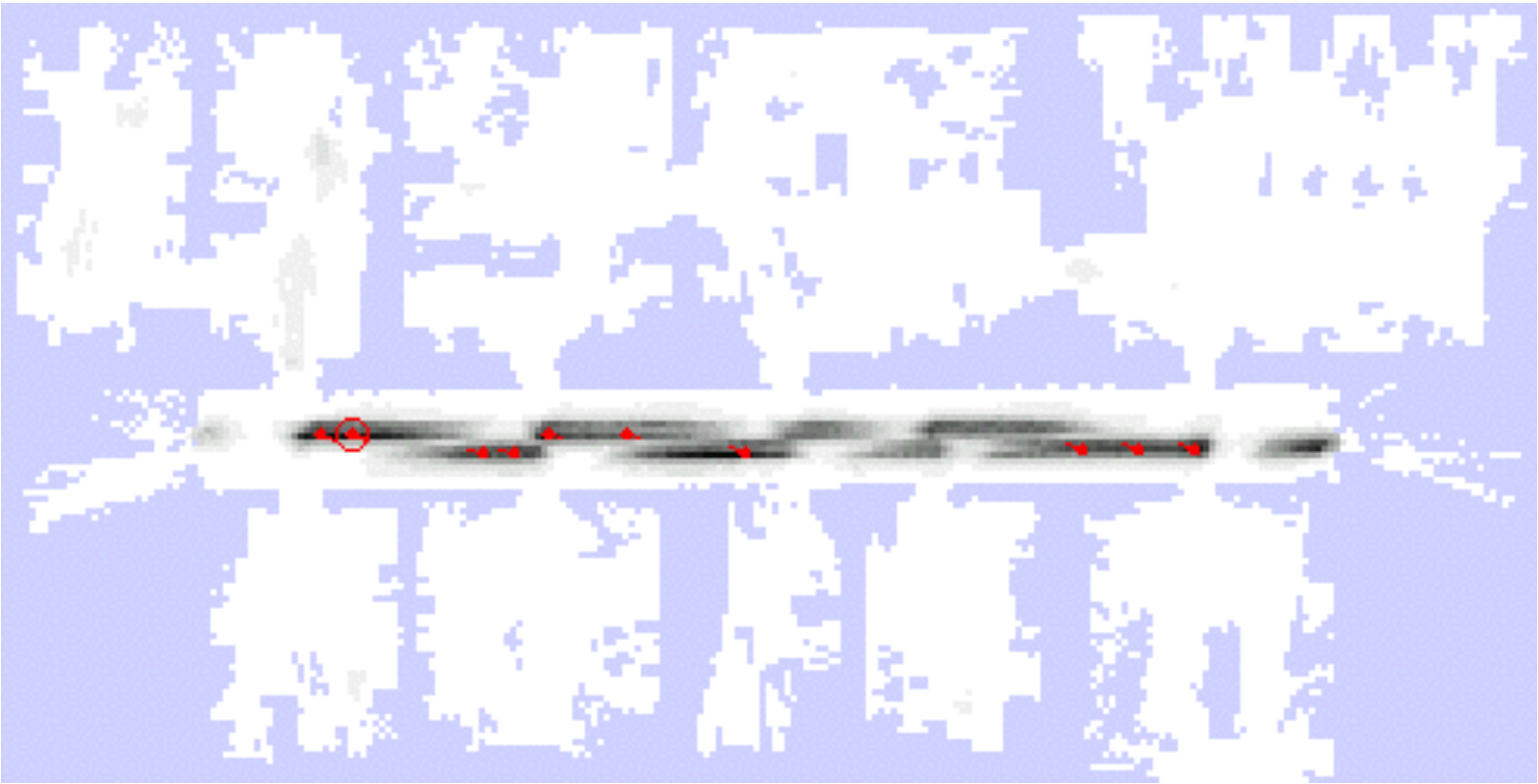
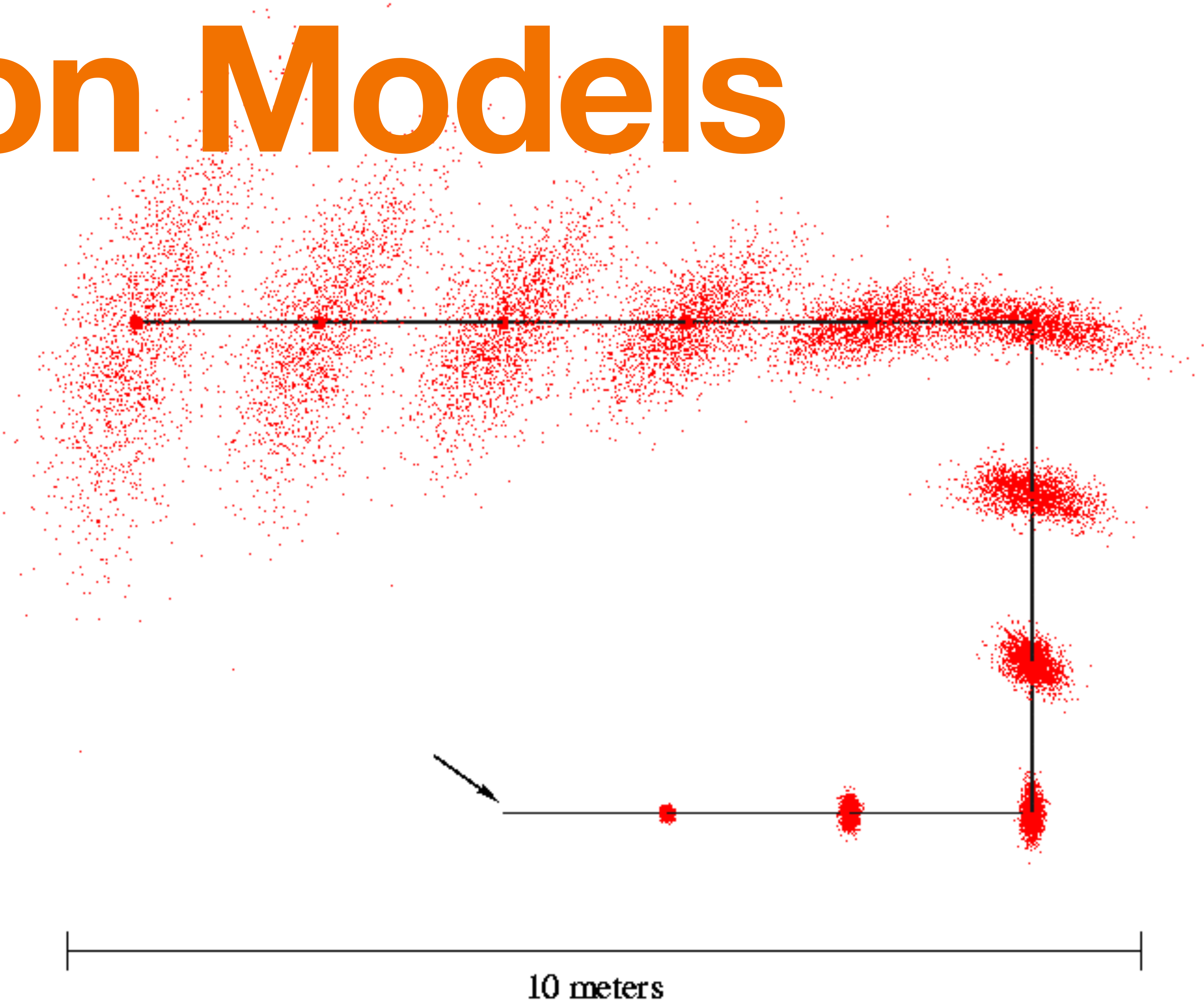
Lecture 18
Mobile Robotics - II - Sensor and Motion Models

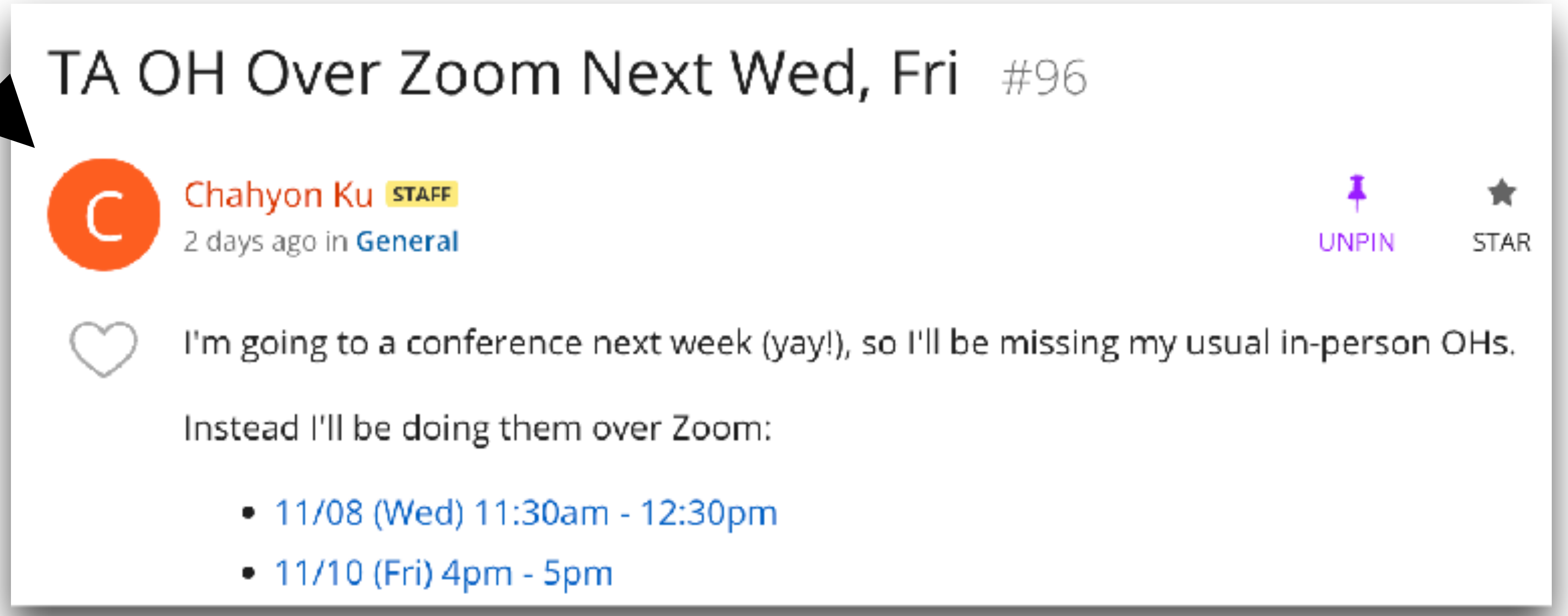




from Probabilistic Robotics

Course logistics

- **Quiz 16-17 was posted yesterday and was due before the lecture.**
- Project 4 is posted on 10/30 and will be due **11/15 (extended)**
- Details on the Final Project will be announced later this week.
- Chahyon's OH for this week will be on Zoom.



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

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

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

Conditioning

- Bayes rule and background knowledge:

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

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

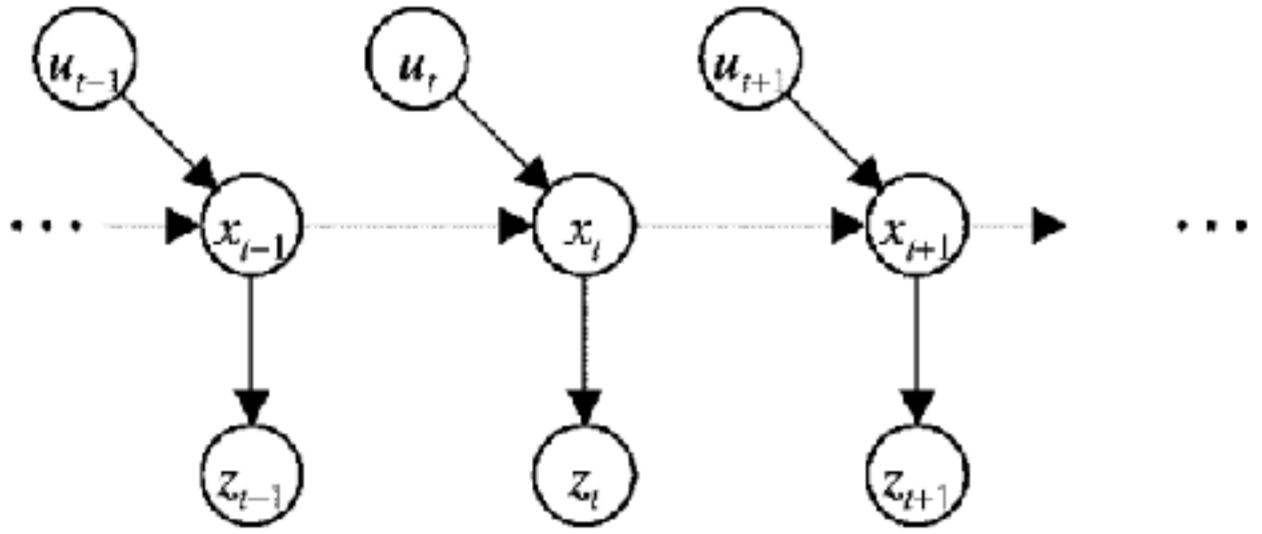
Recursive Bayesian Updating

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

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

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

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

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

Probabilistic Motion Models

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

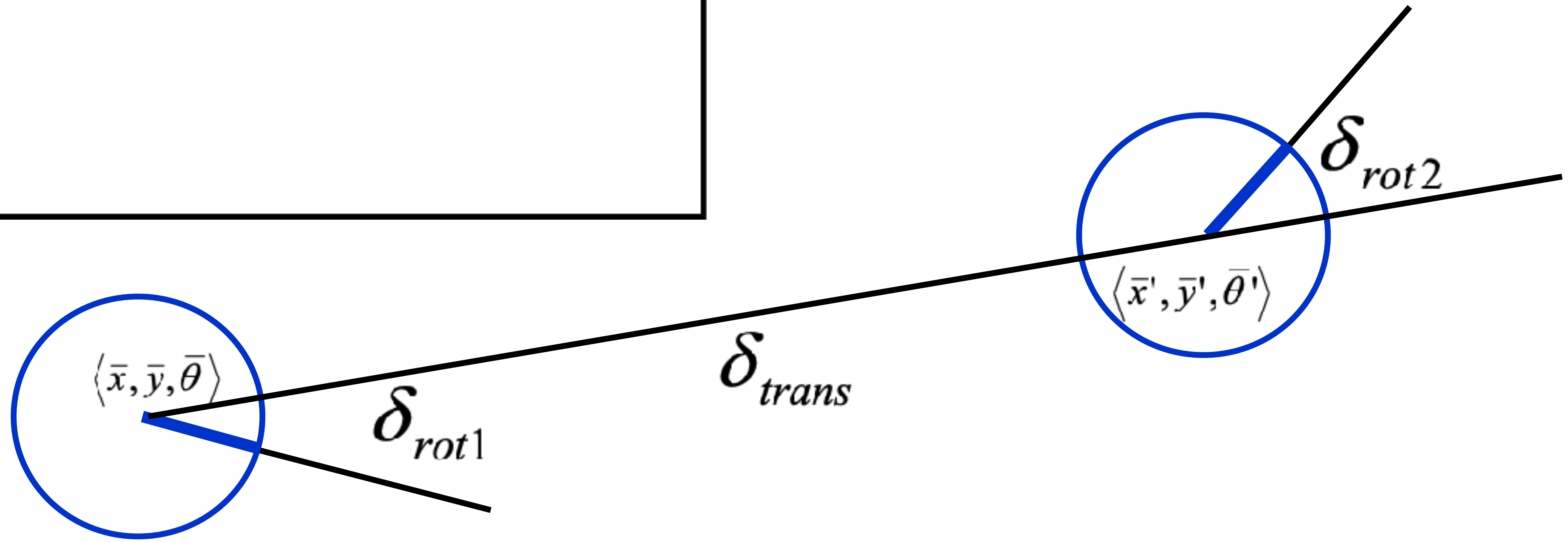
Probabilistic Kinematics

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

$$\delta_{trans} =$$

$$\delta_{rot1} =$$

$$\delta_{rot2} =$$



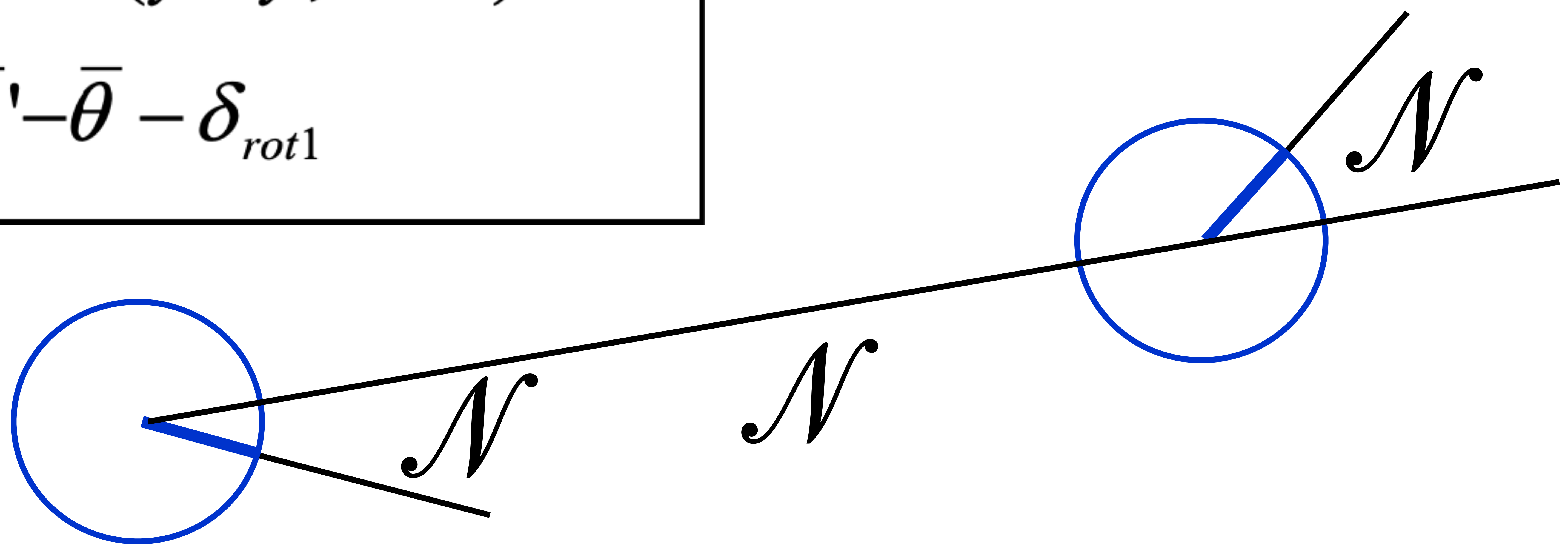
Slide borrowed from Dieter Fox

Noise Model for Motion

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

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

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



Noise Model for Motion

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

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

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

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

Slide borrowed from Dieter Fox

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

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

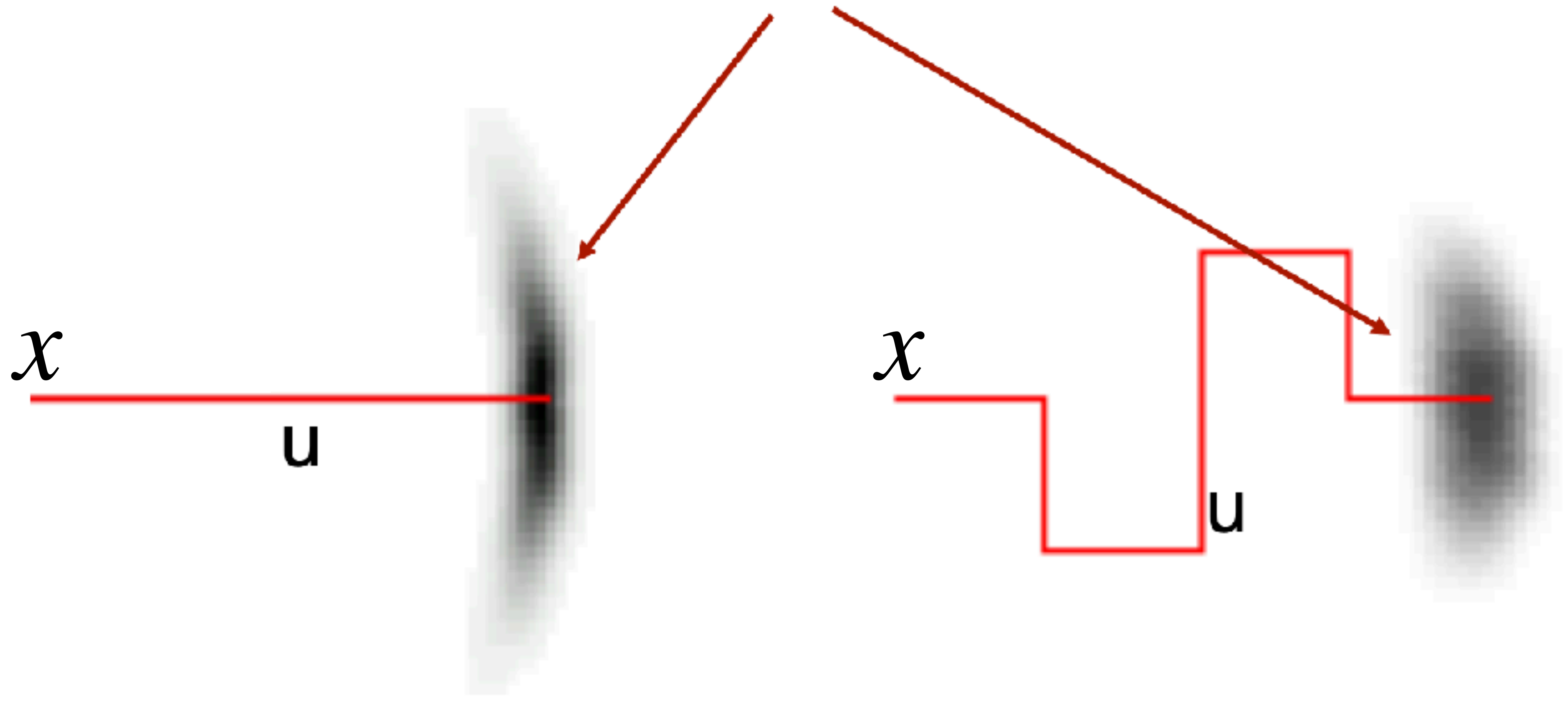
Finding the posterior

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):

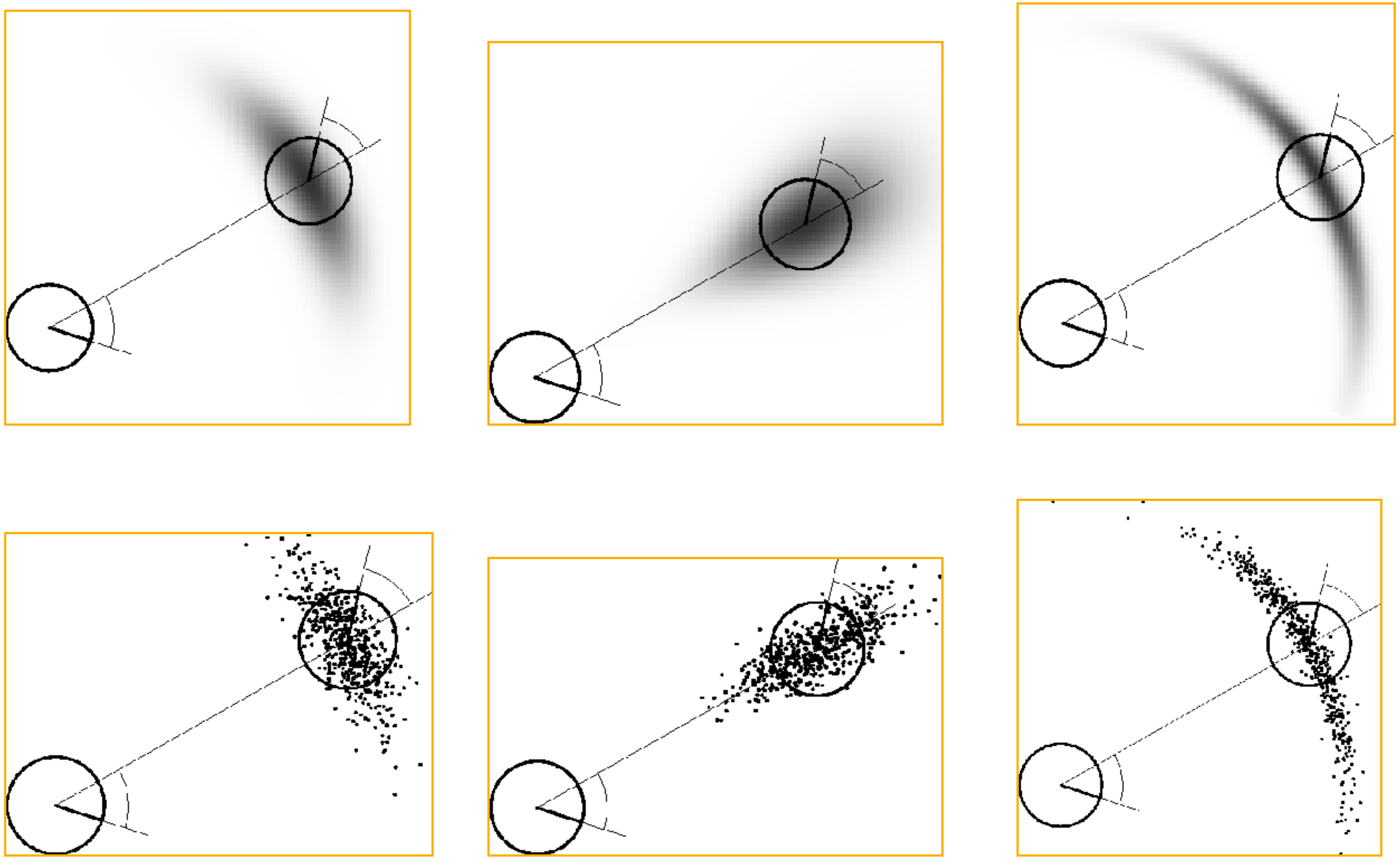
   $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$

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. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2\, \delta_{trans})$

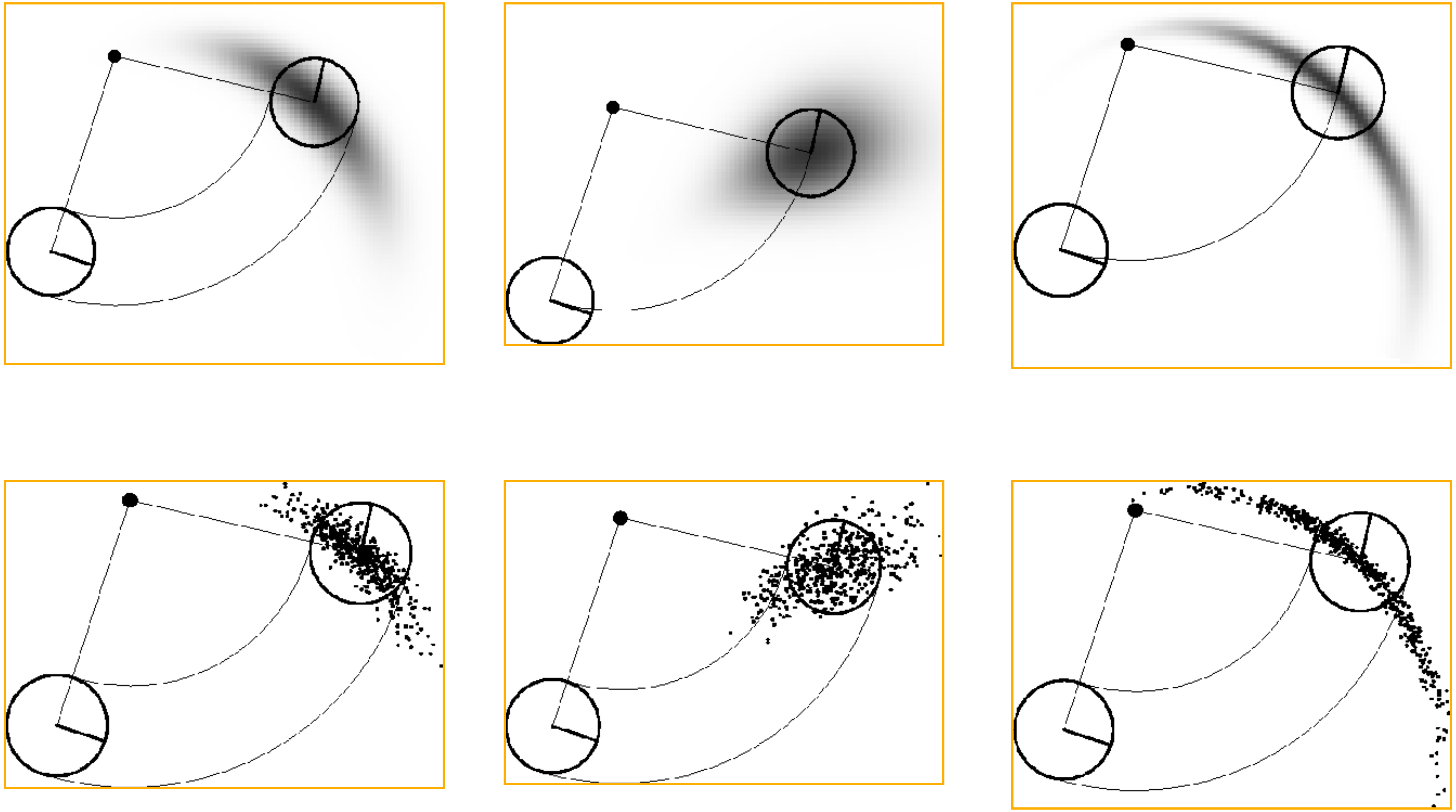
4. $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
5. $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$
6. $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

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

Examples (odometry based)

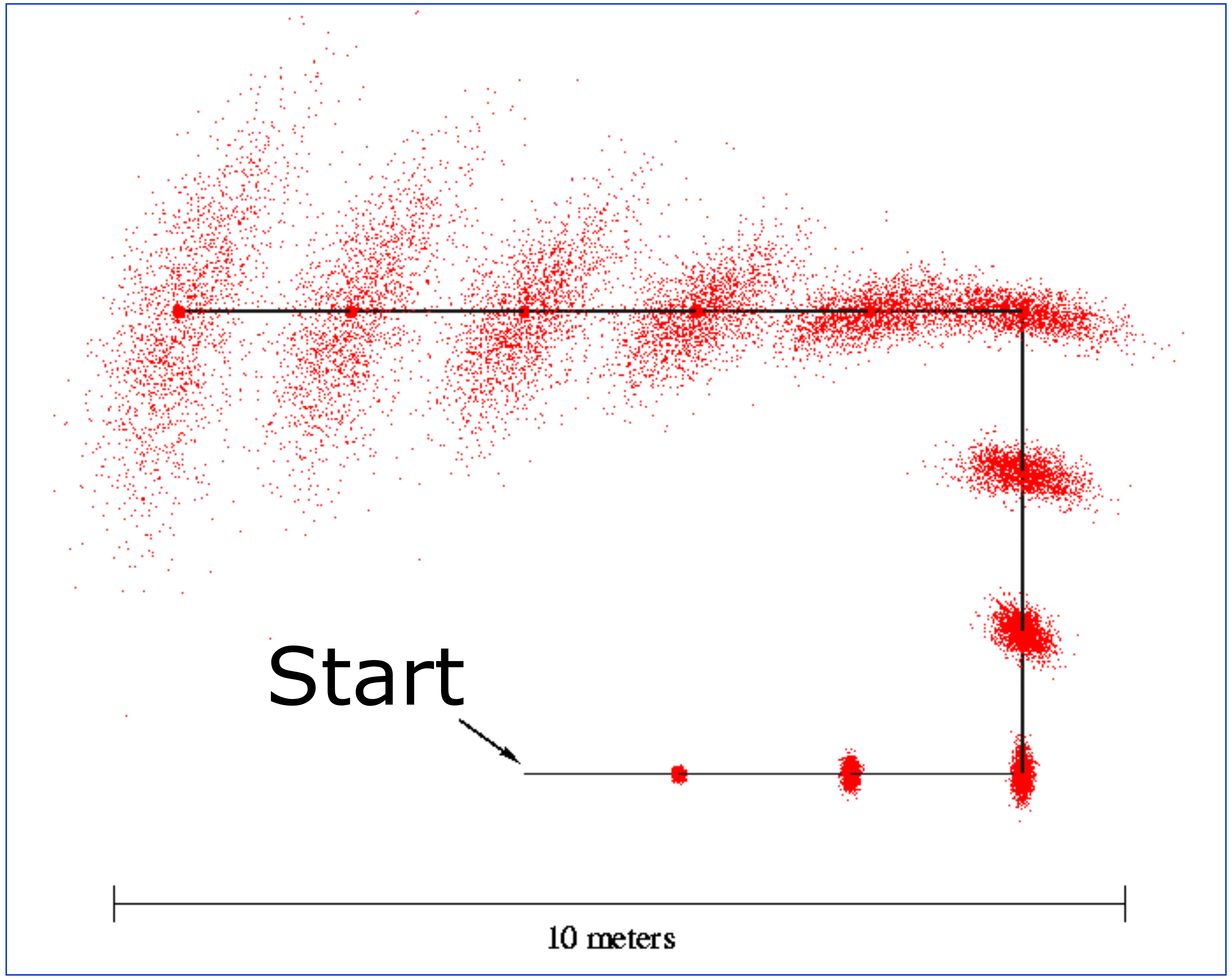


Examples (velocity based)

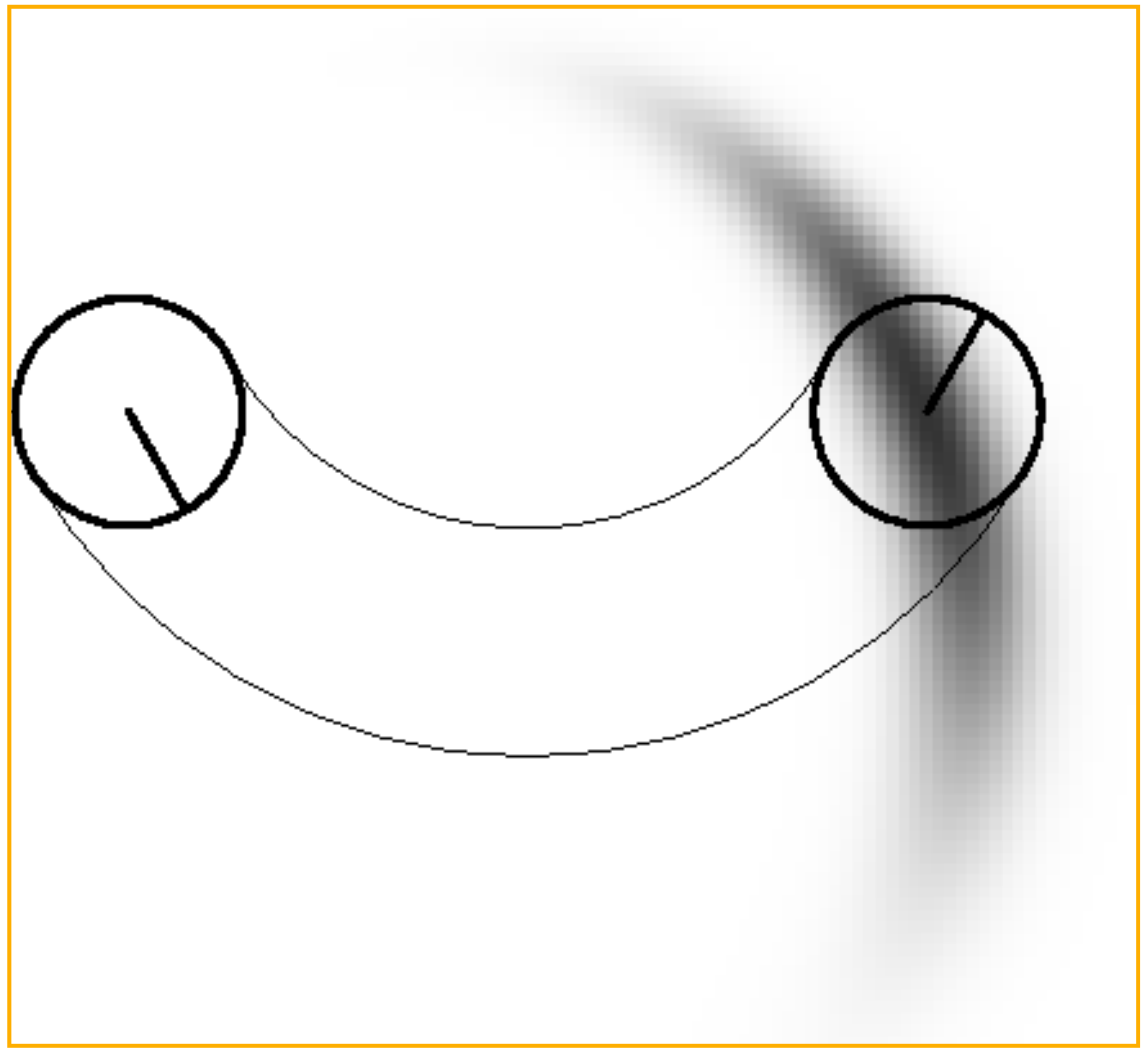


Slide borrowed from Dieter Fox

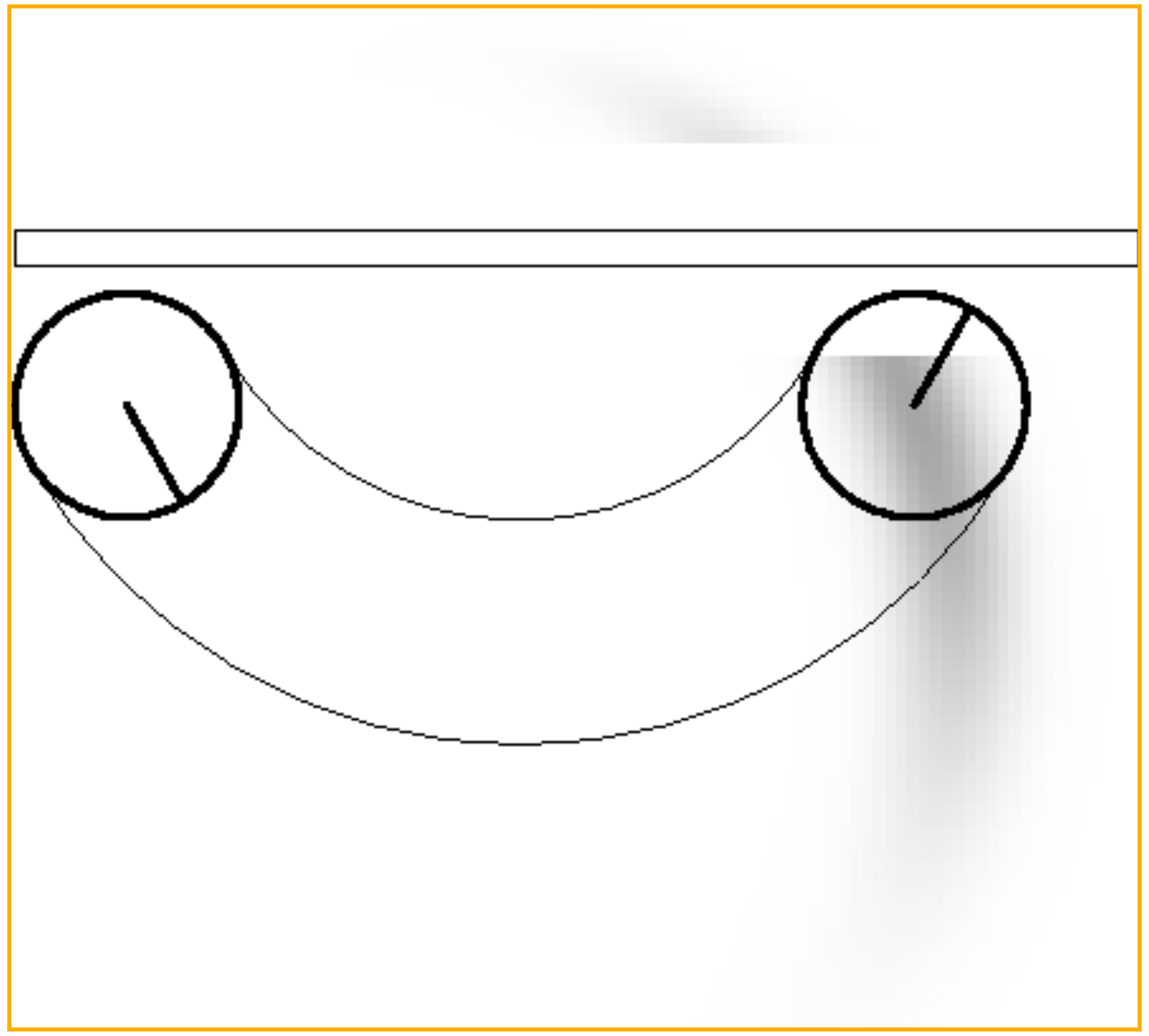
Sample-based Motion



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?

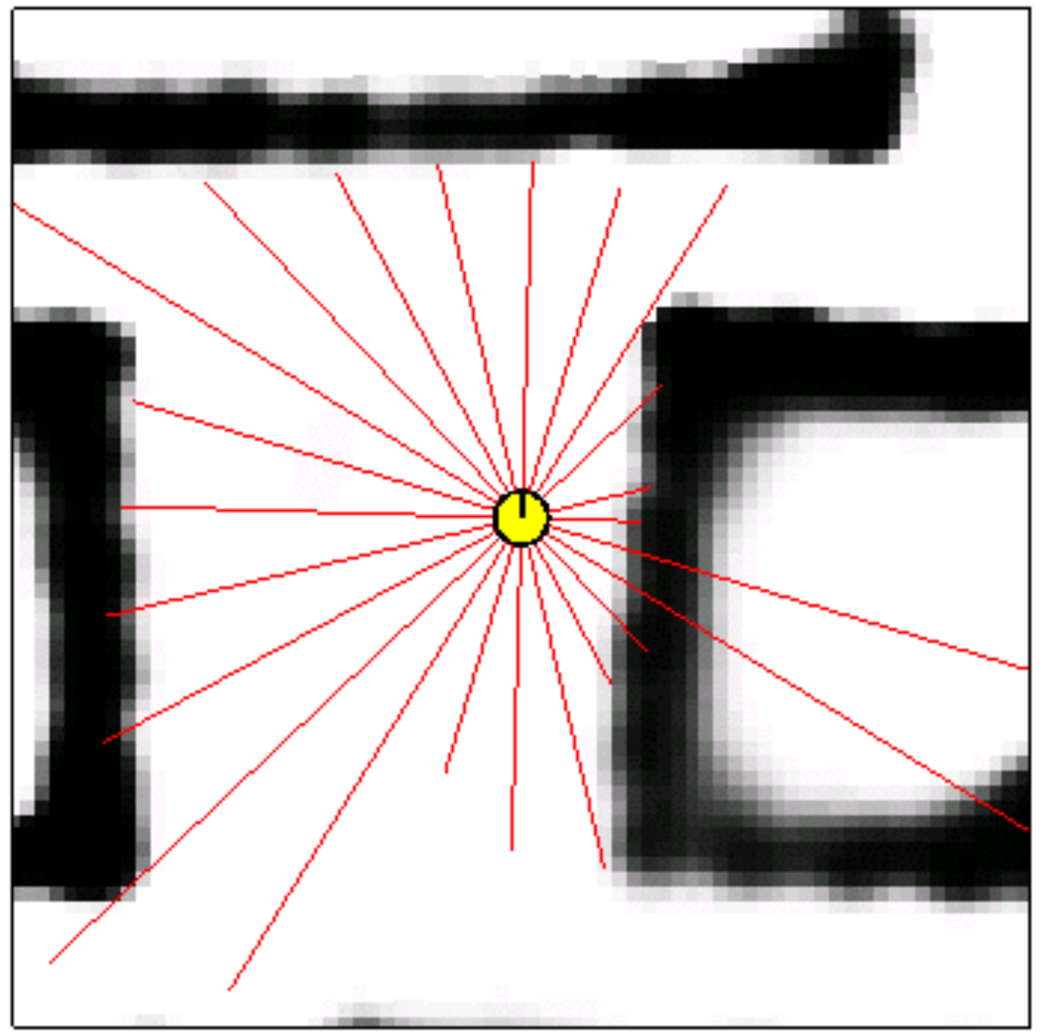
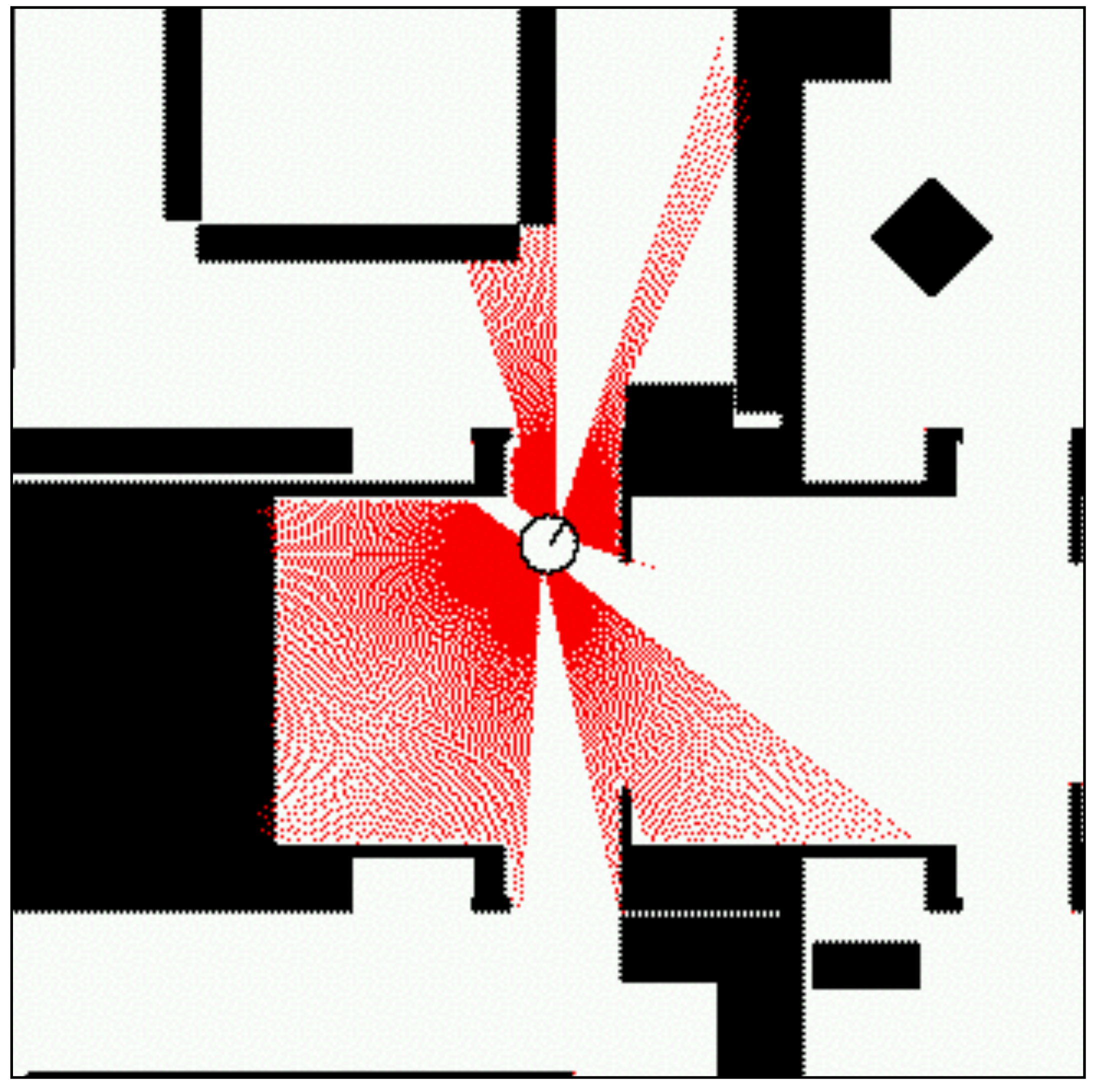
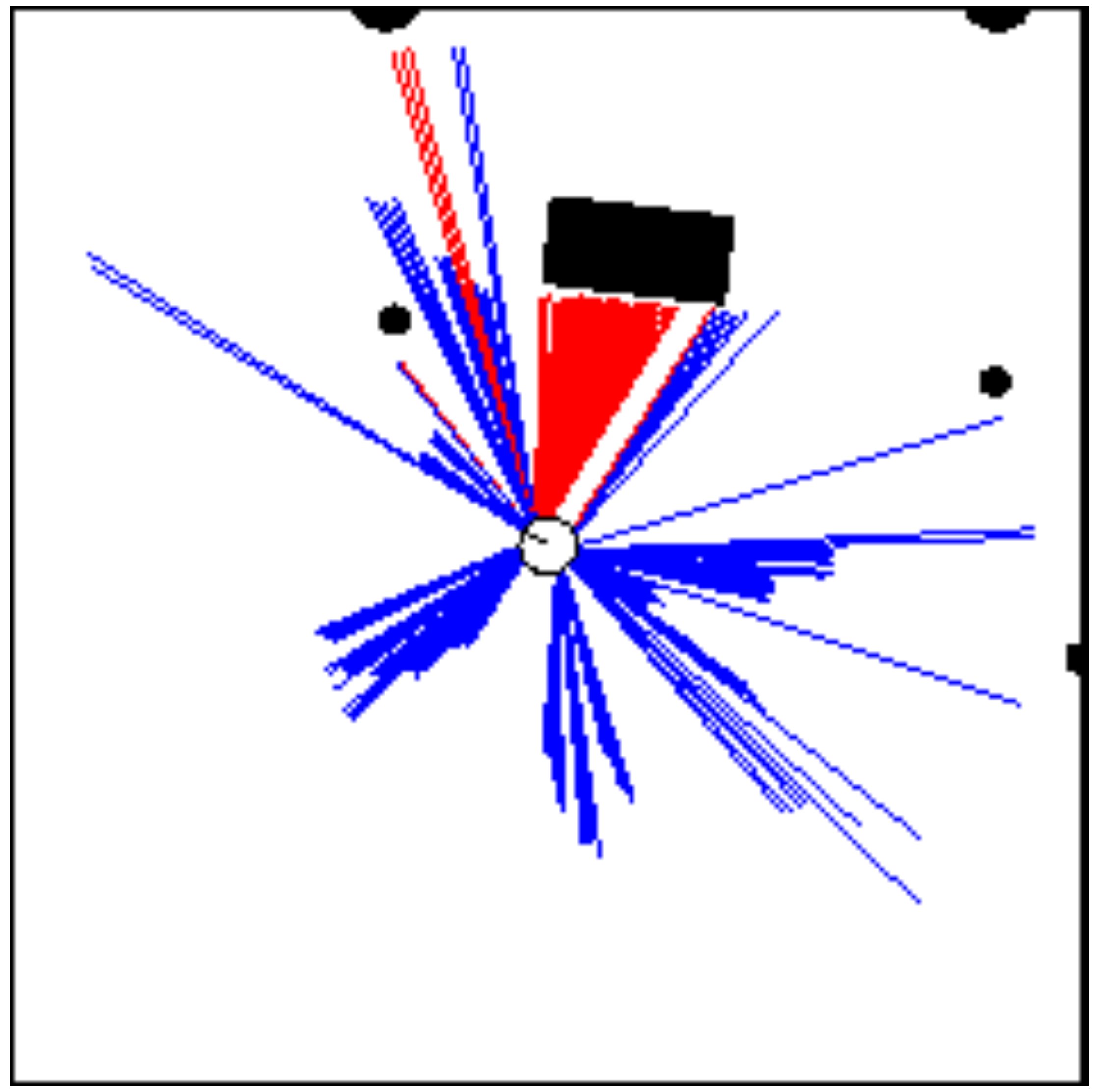
Probabilistic Sensor Models

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

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

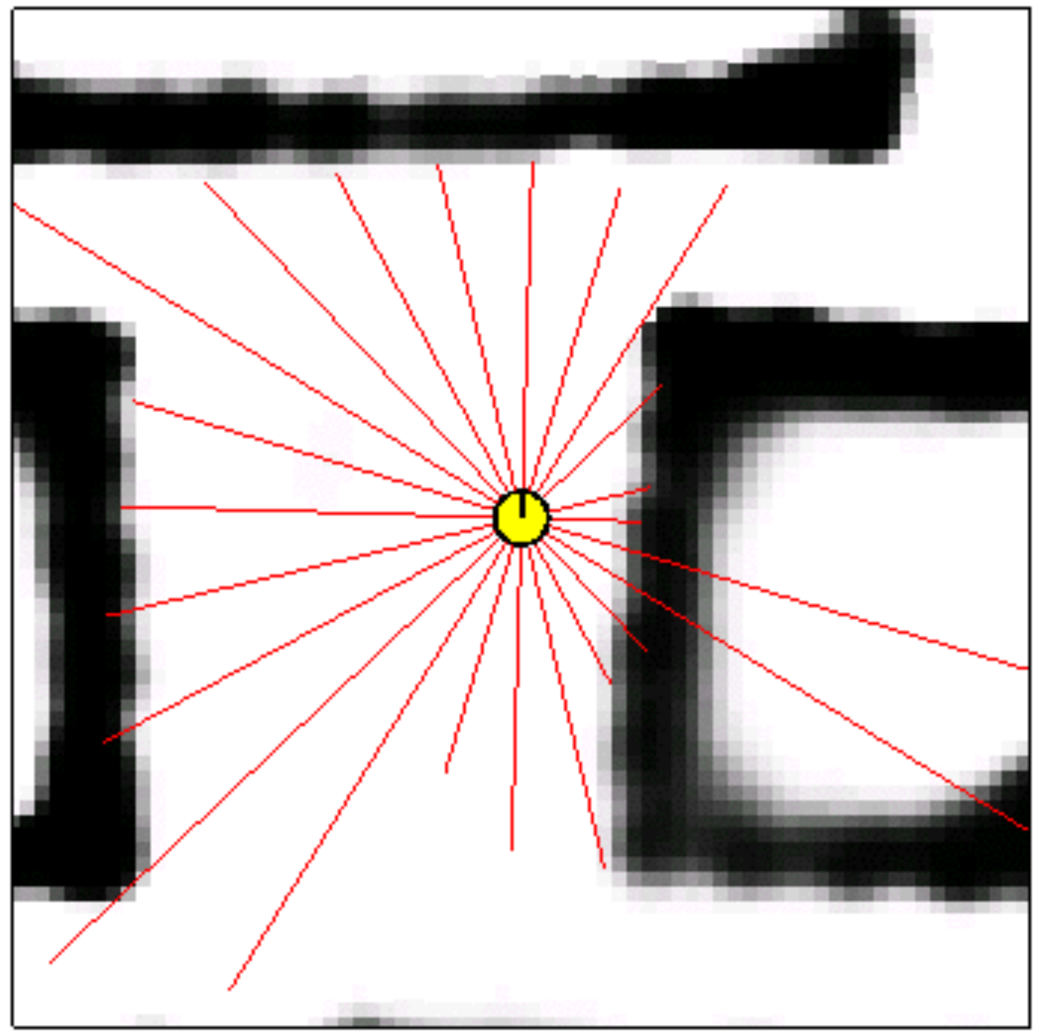
- Scan z consists of K measurements.

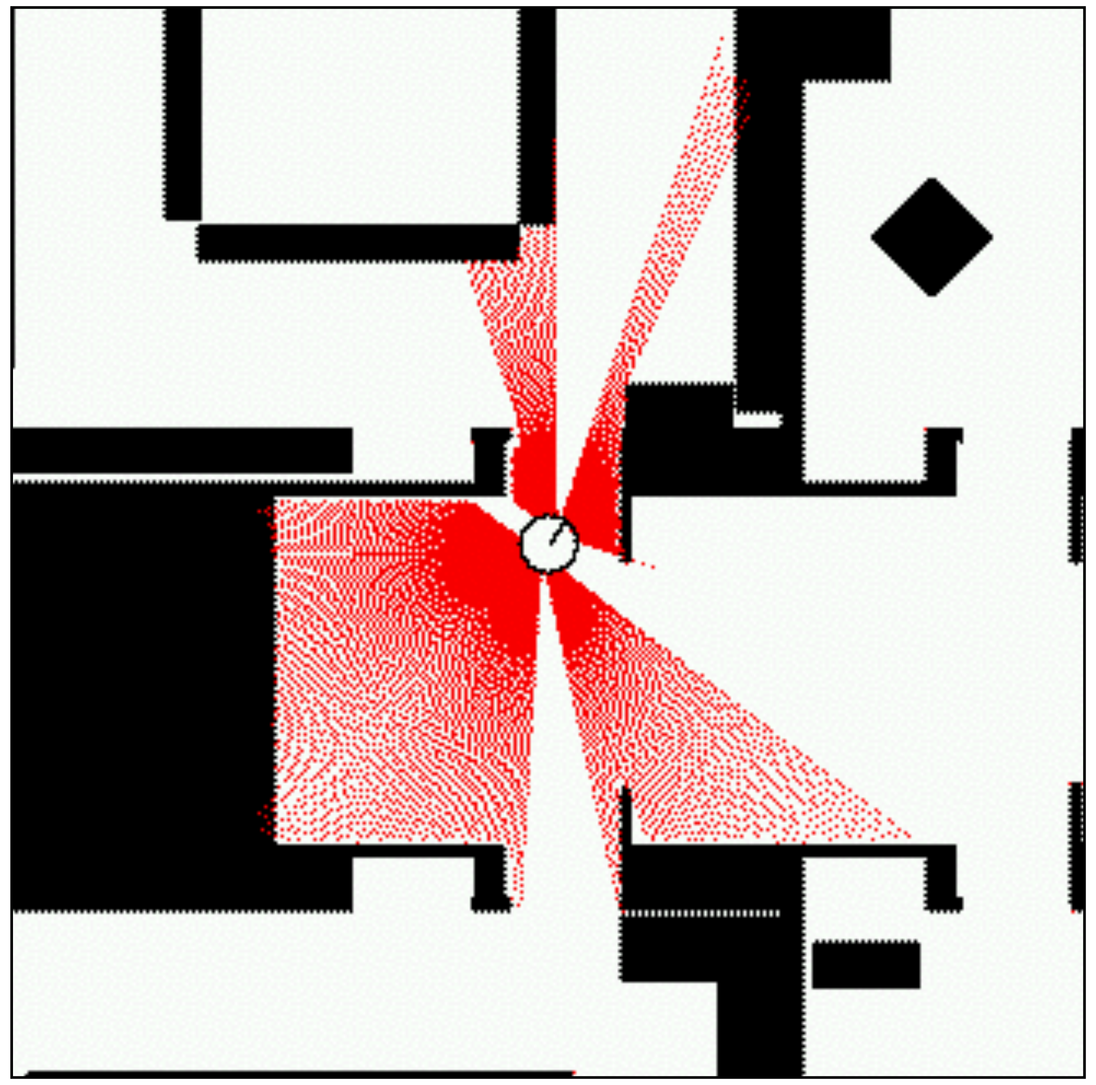
$$z = \{z_1, z_2, \ldots, z_K\}$$

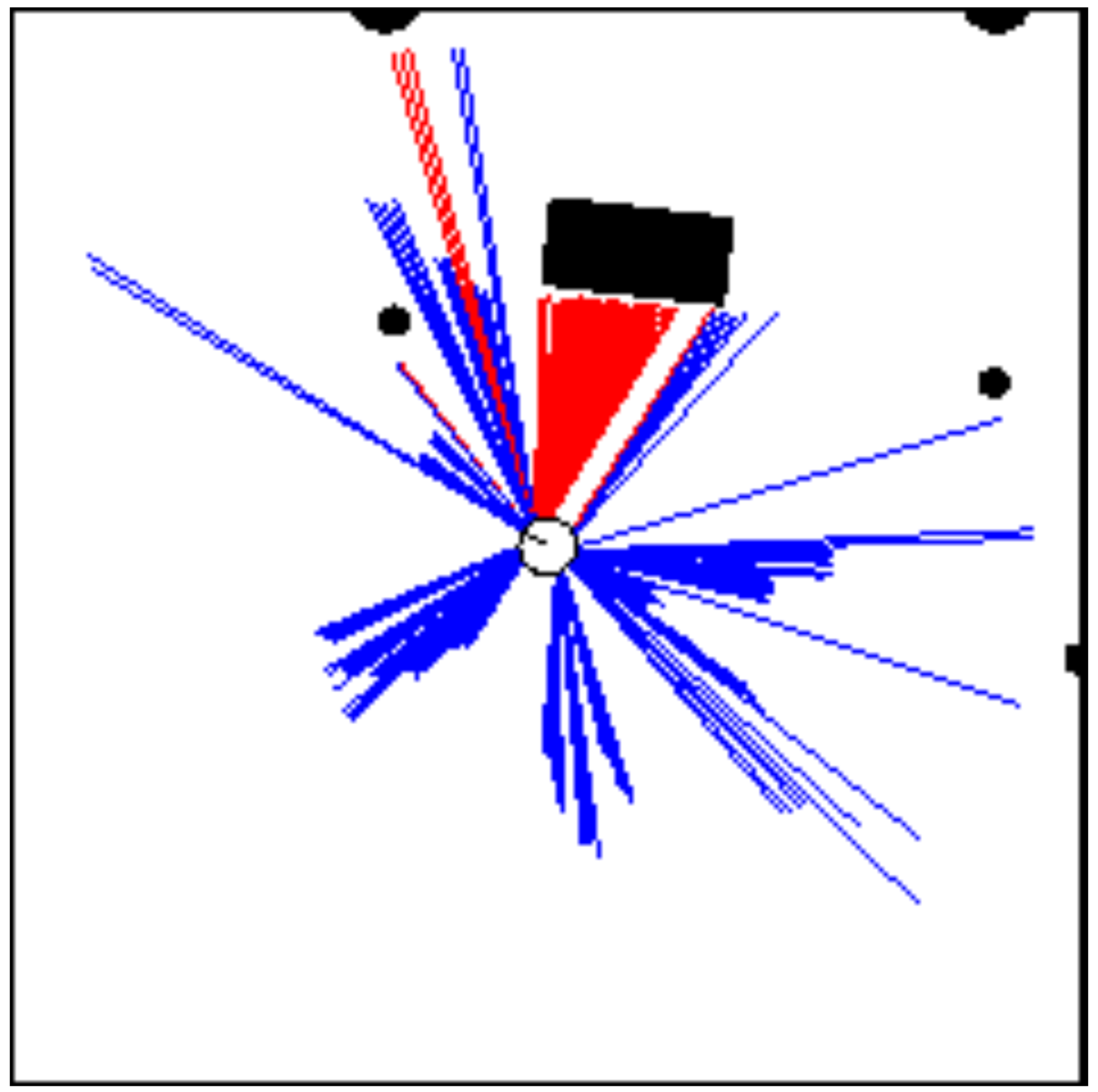
- Individual measurements are independent given the robot position and a map.

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

Beam-based Sensor Model







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

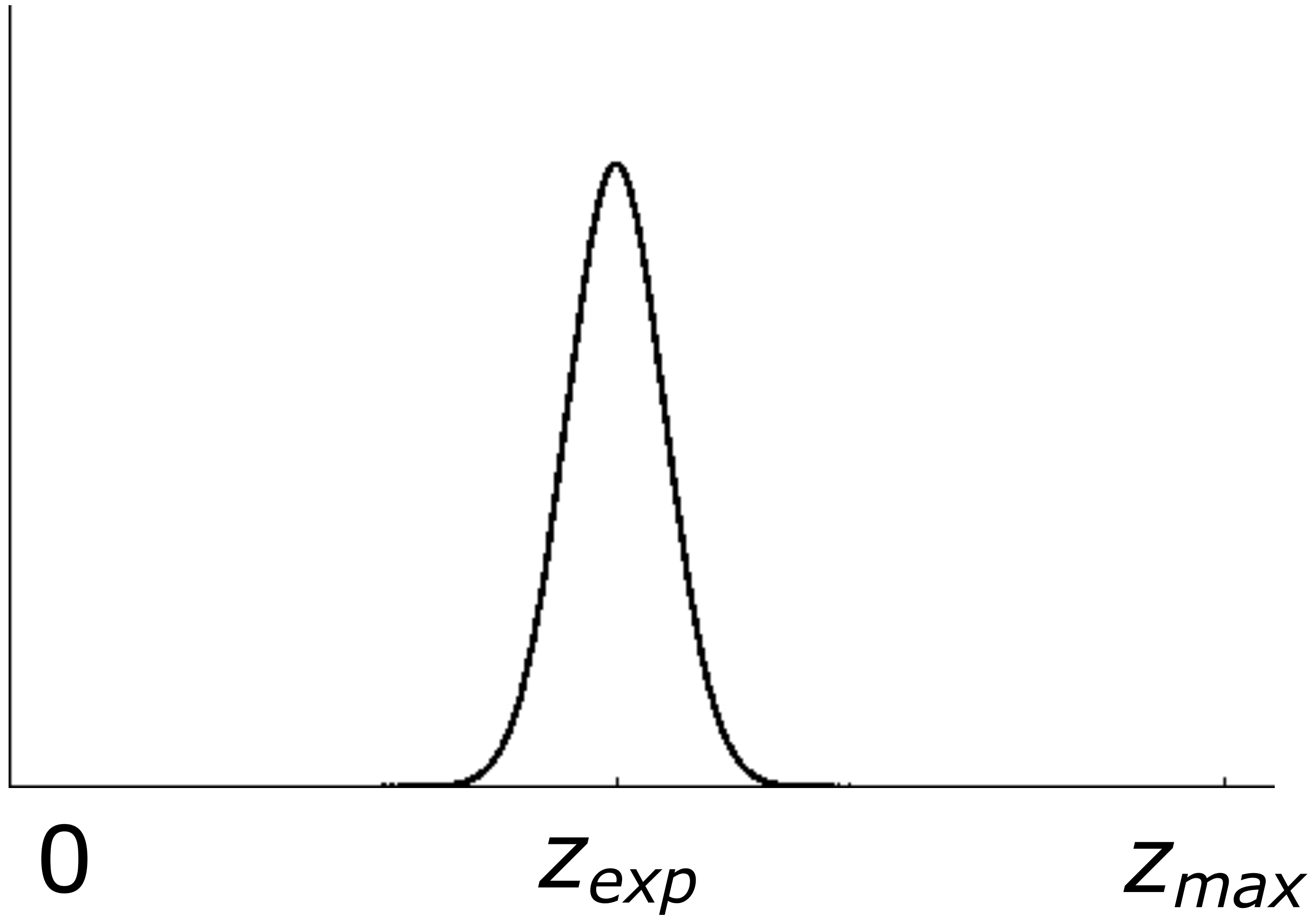
Slide borrowed from Dieter Fox

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.

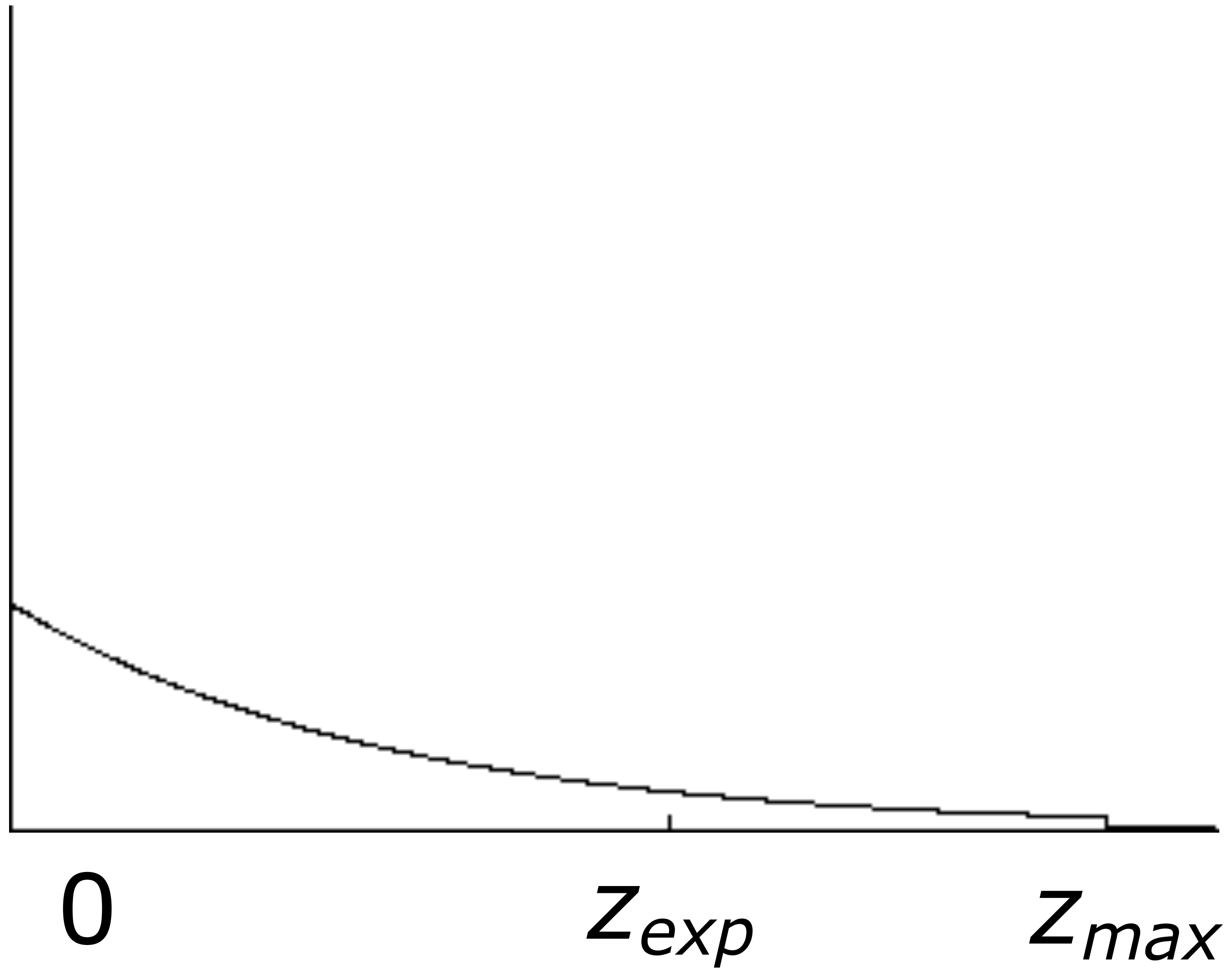
Beam-based Proximity Model

Measurement noise



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

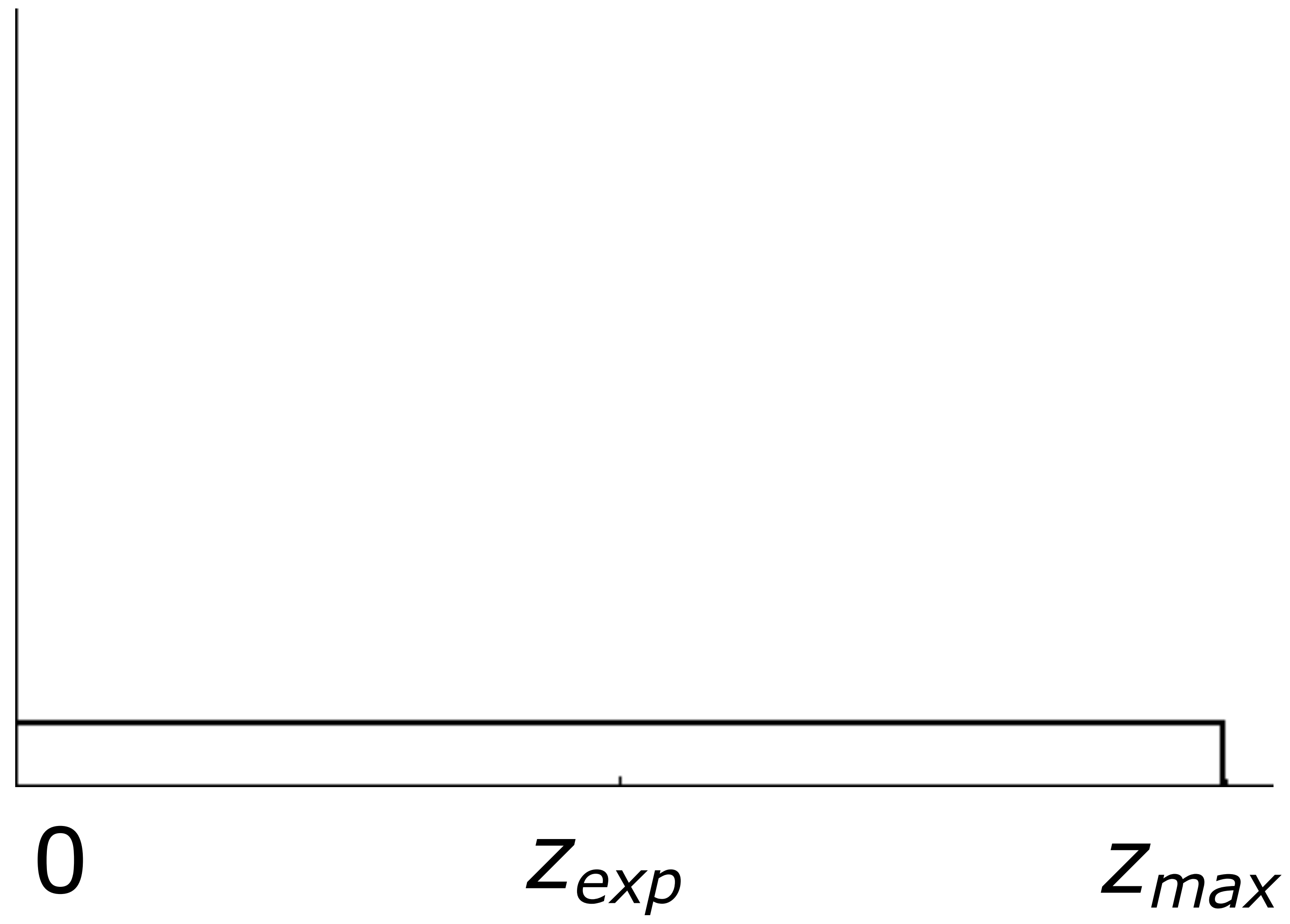
Unexpected obstacles



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

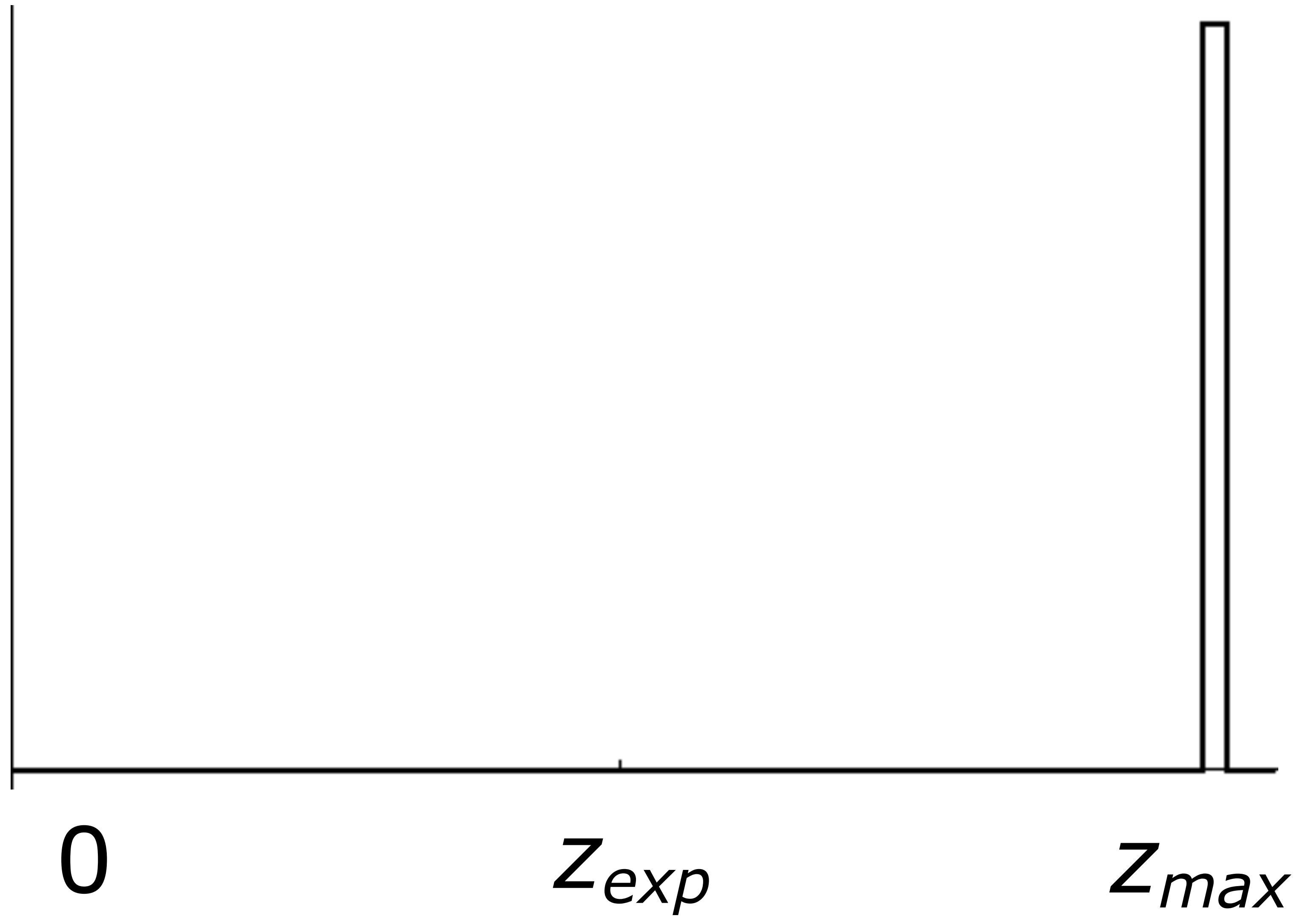
Beam-based Proximity Model

Random measurement



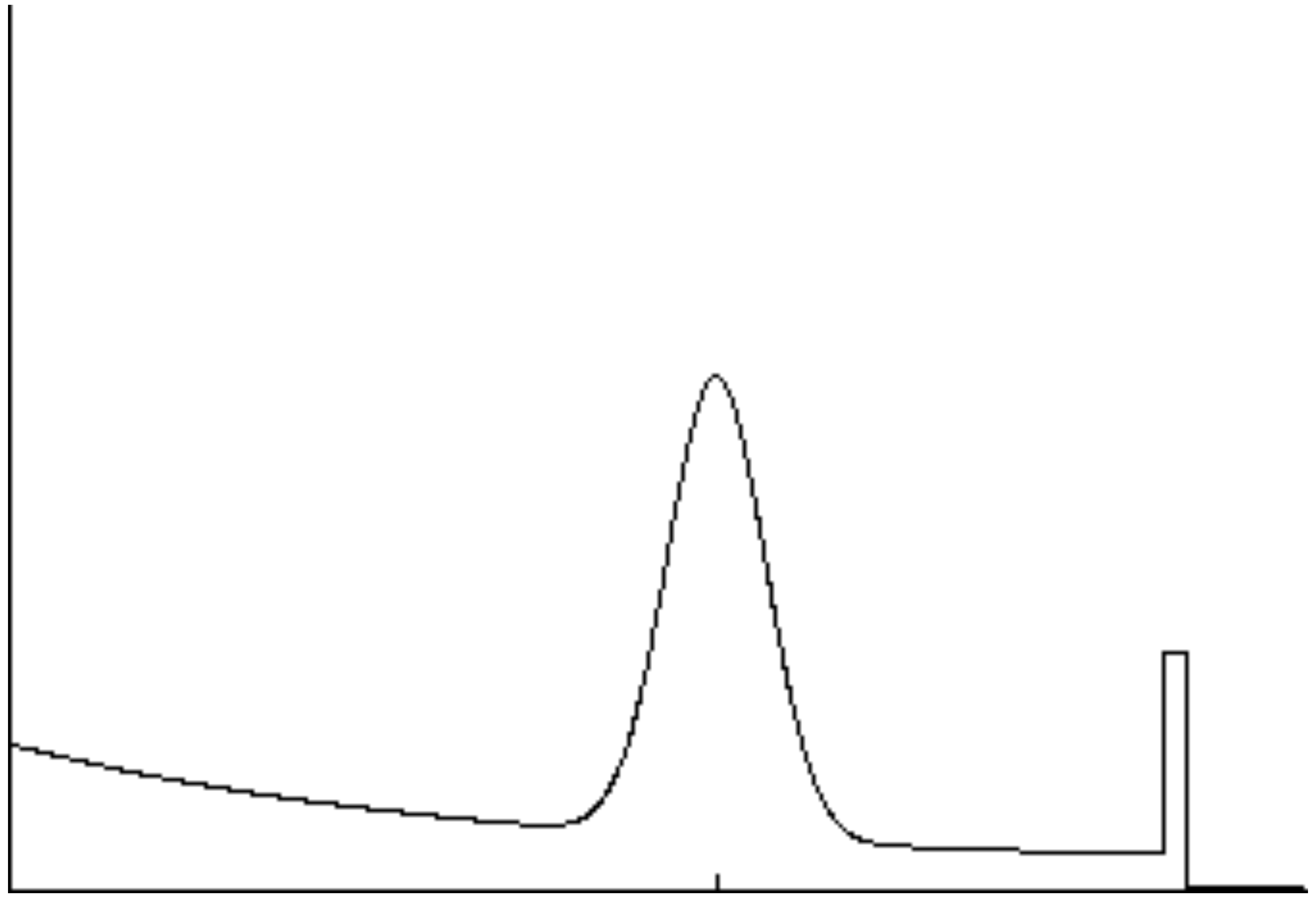
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Max range



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

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:

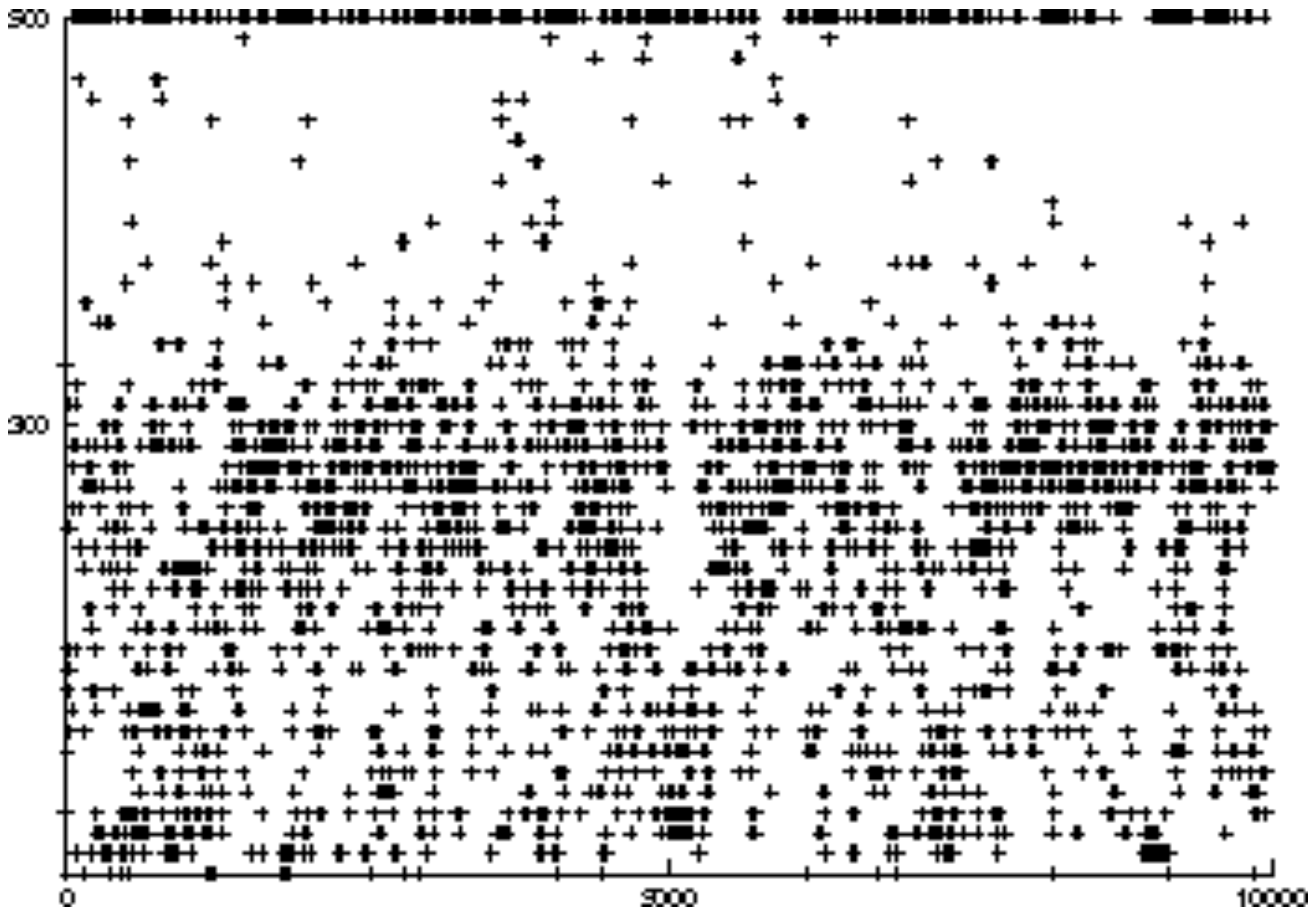
$$P(z \mid z_{\exp})$$

- Search parameter space.

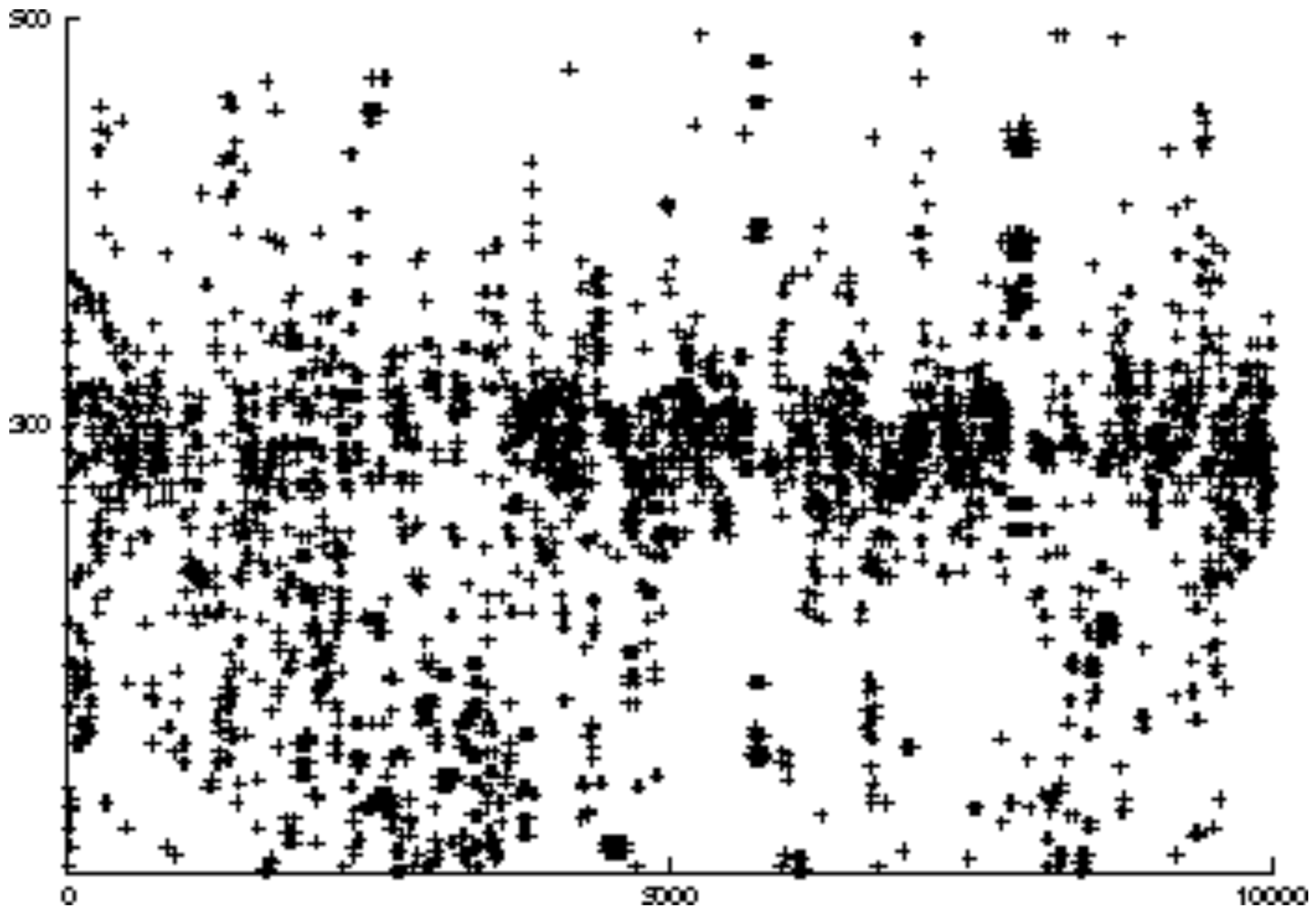
- EM to find mixture parameters
  - Assign measurements to densities.
  - Estimate densities using assignments.
  - Reassign measurements.

Raw Sensor Data

Measured distances for expected distance of 300 cm.

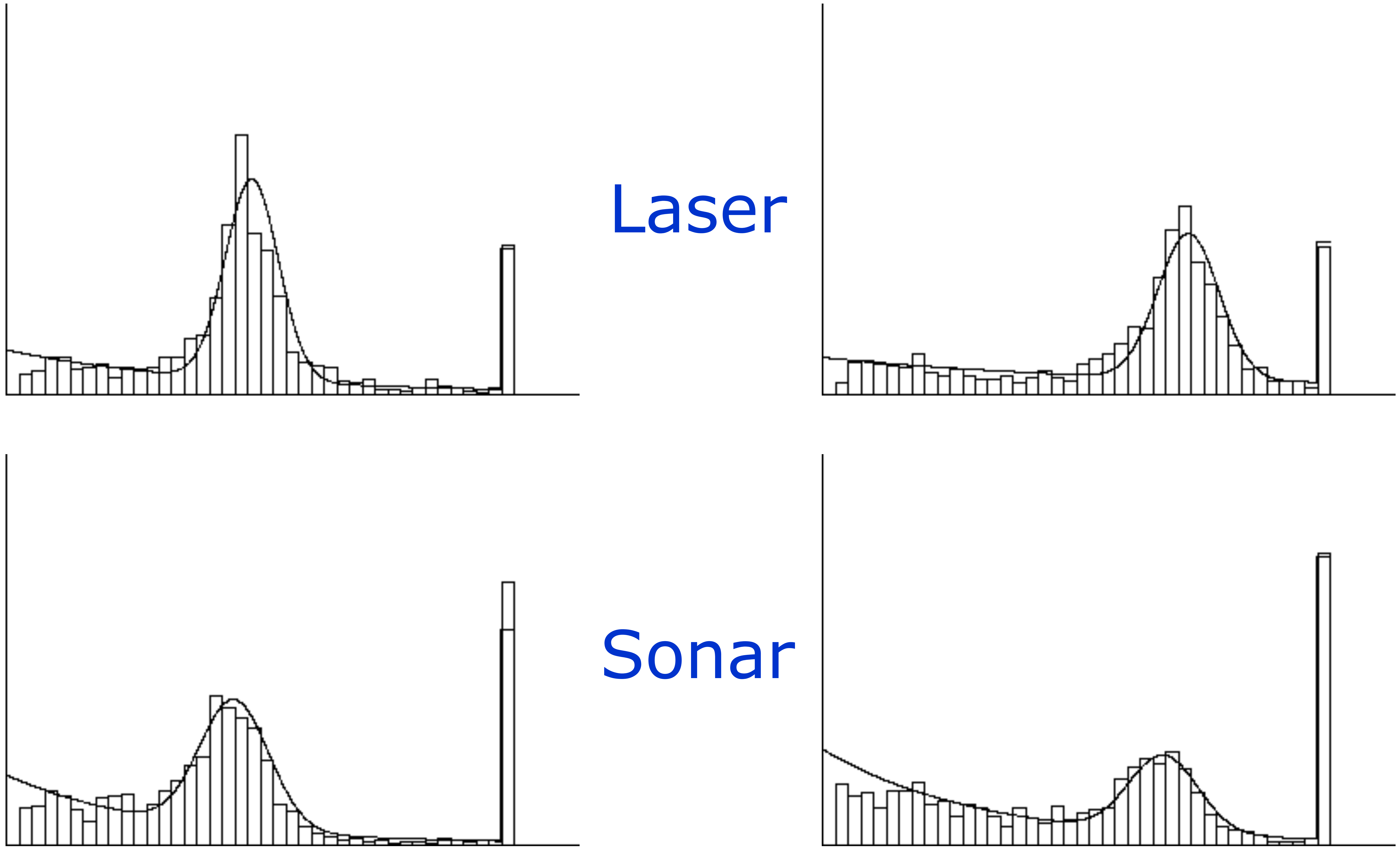


Sonar



Laser

Approximation Results



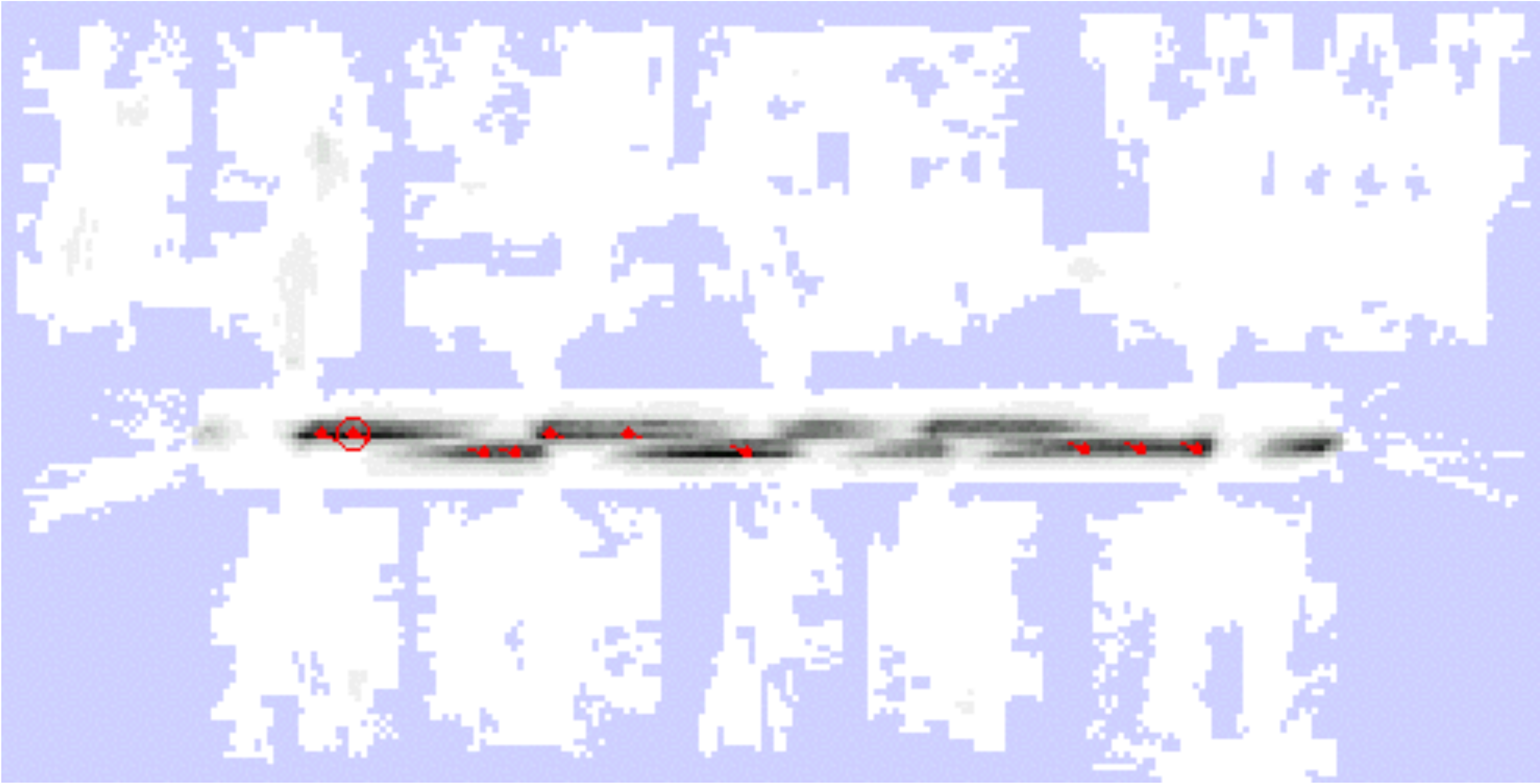
300cm

400cm

Slide borrowed from Dieter Fox

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.

Scan-based Model

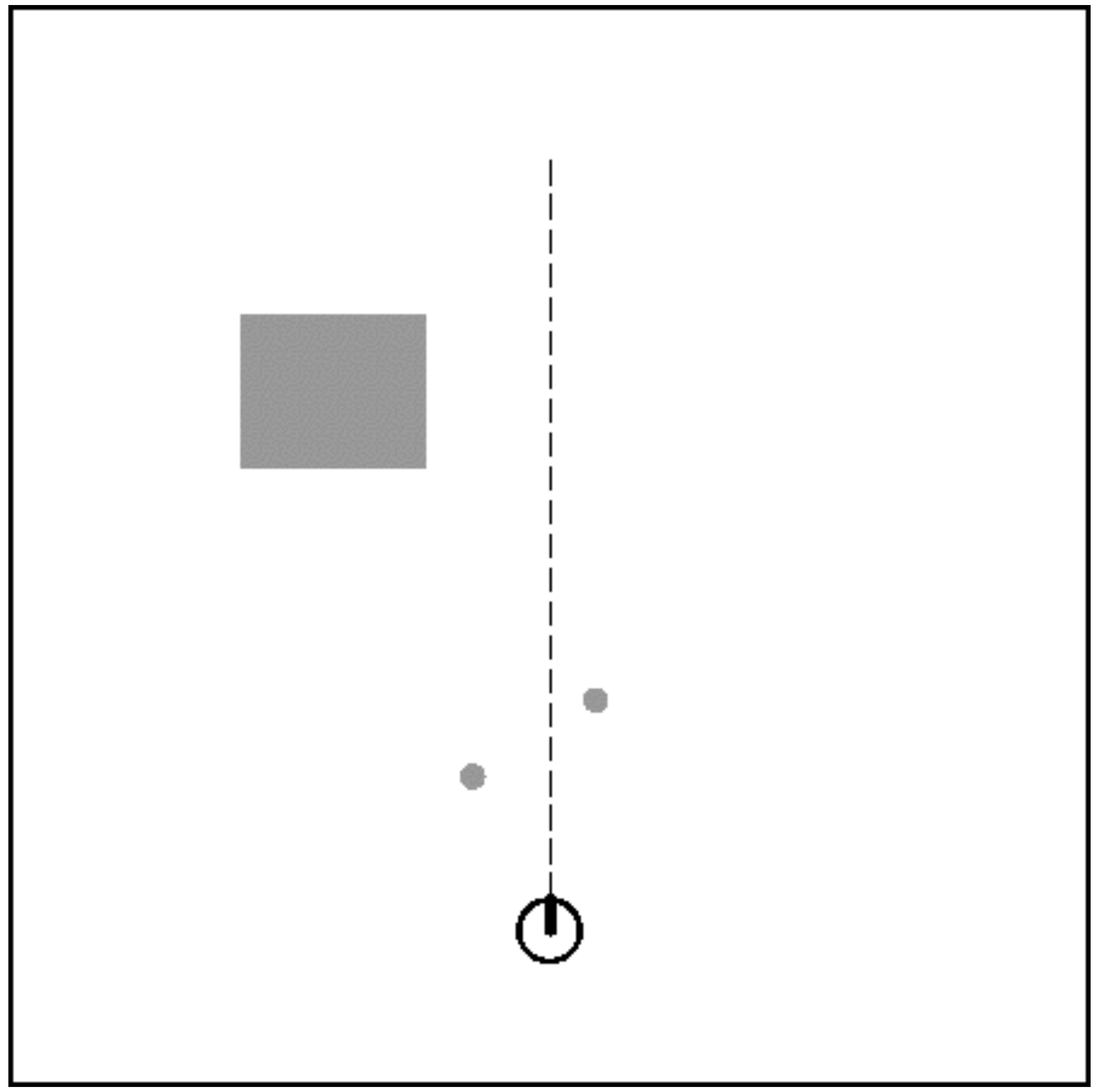
- Beam-based model is …
  - not smooth for small obstacles and at edges.
  - not very efficient.

- Idea: Instead of following along the beam, just check the end point.

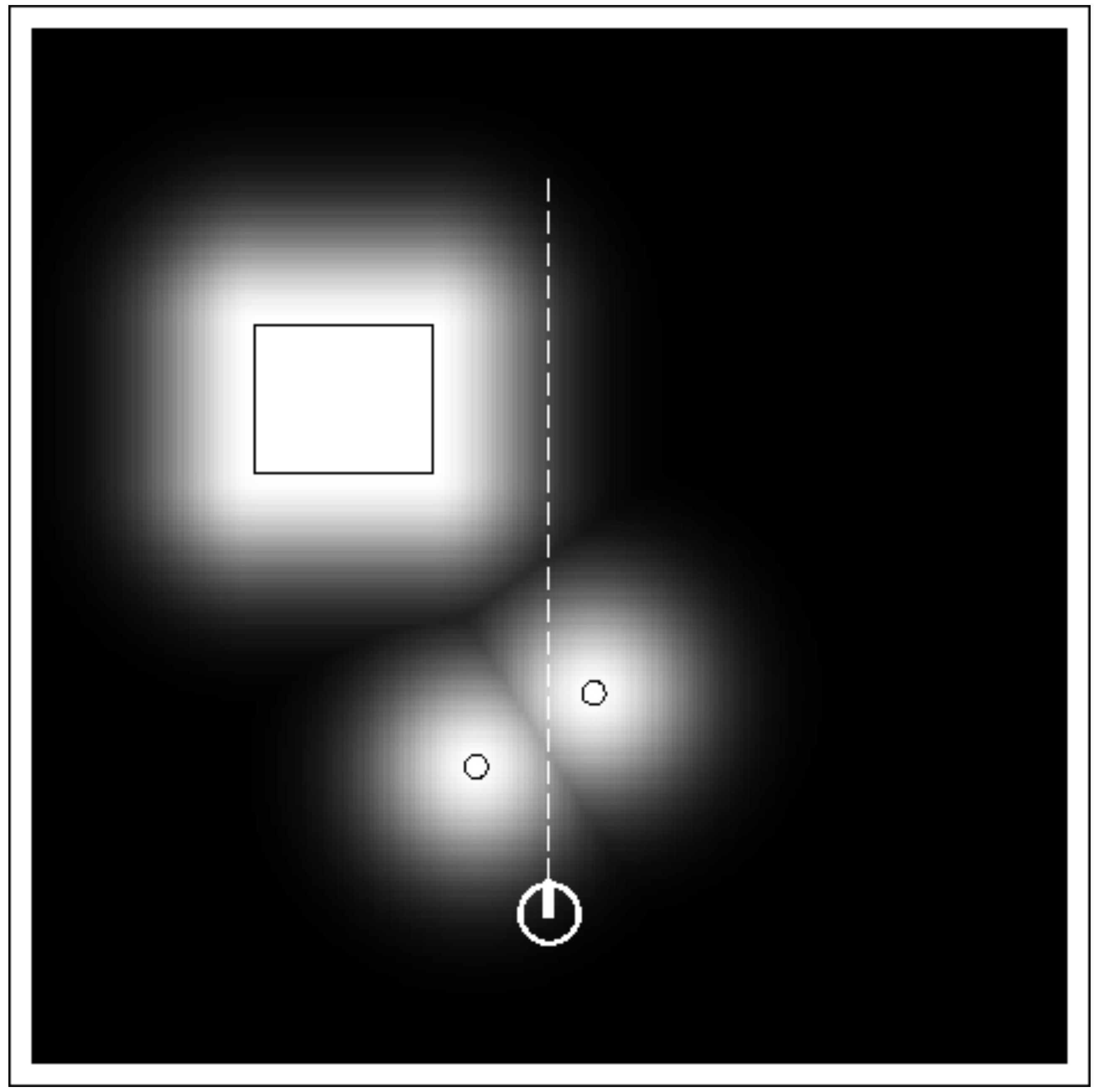
Scan-based Model

- Probability is a mixture of …
  - a Gaussian distribution with mean at distance to closest obstacle,
  - a uniform distribution for random measurements, and
  - a small uniform distribution for max range measurements.
- Again, independence between different components is assumed.

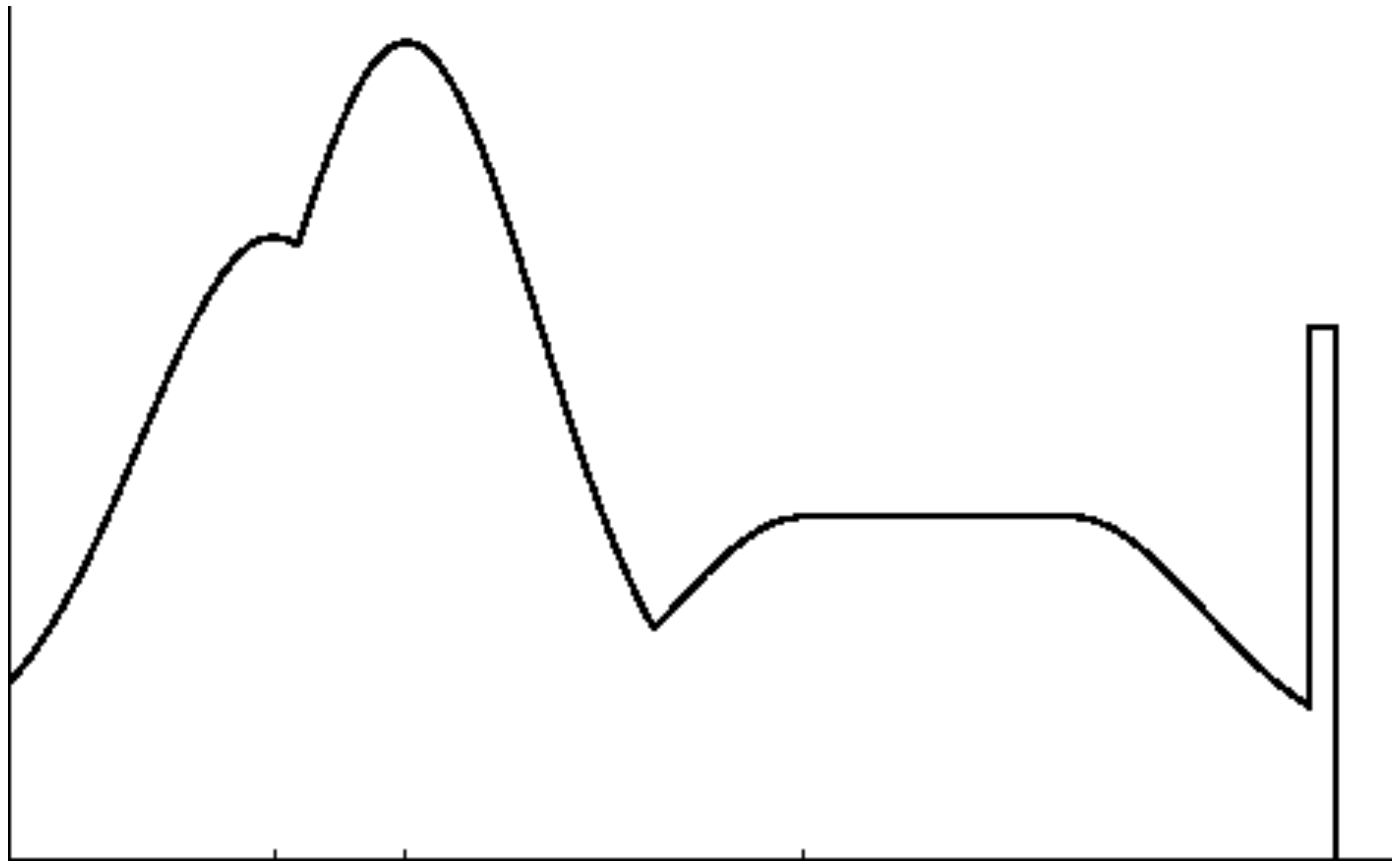
Example



Map m

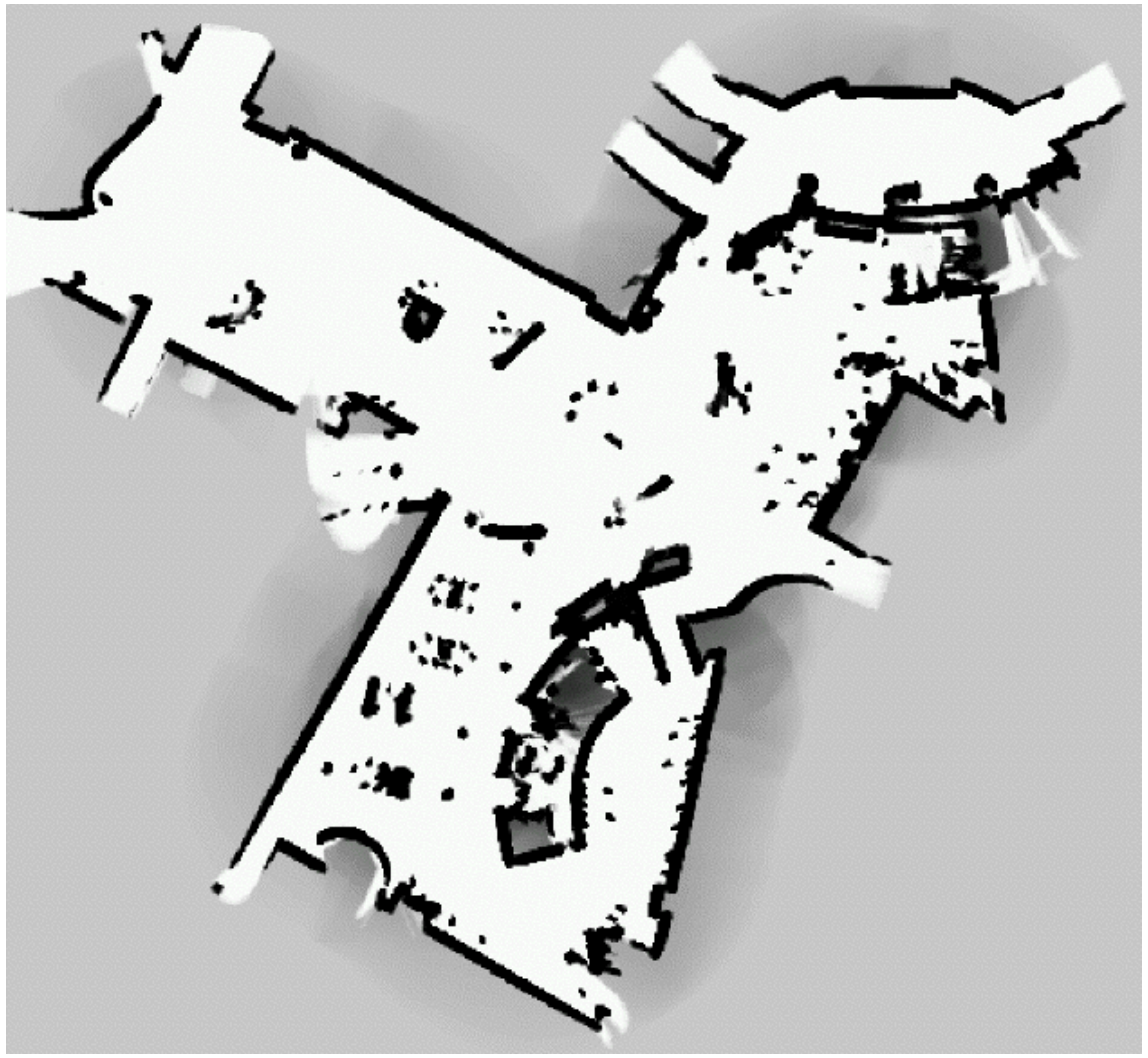


Likelihood field

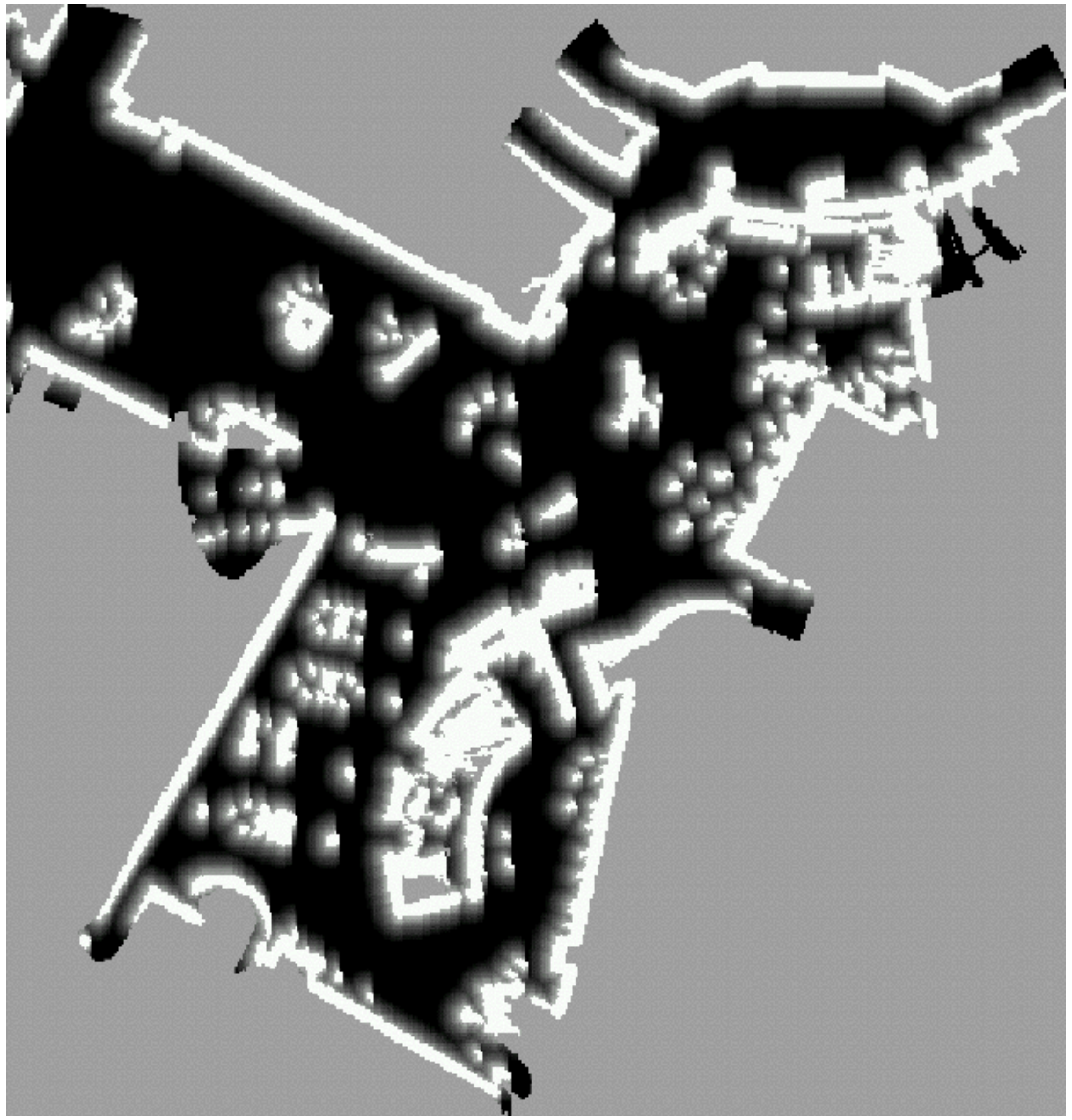


$P(z|x,m)$

San Jose Tech Museum



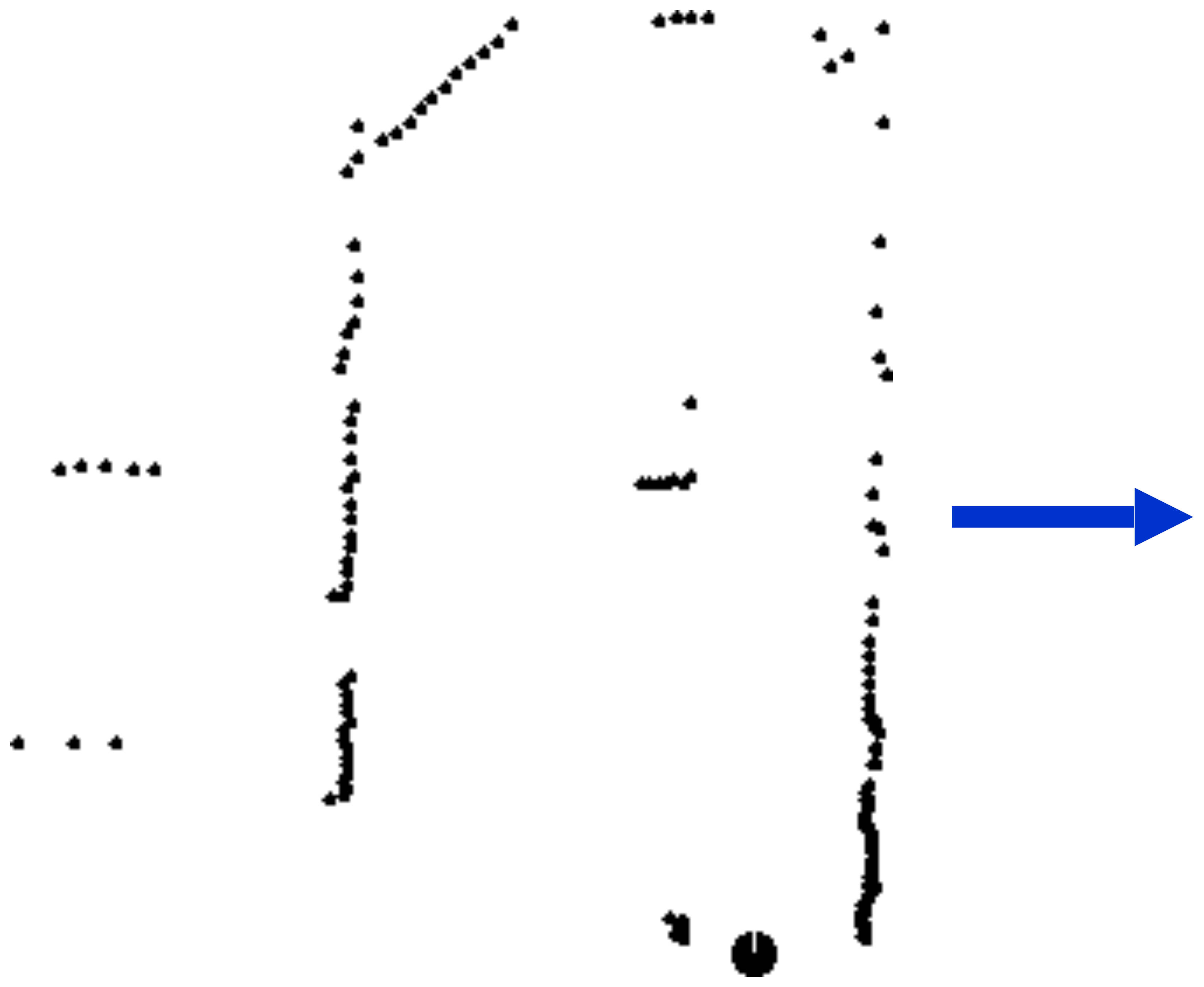
Occupancy grid map

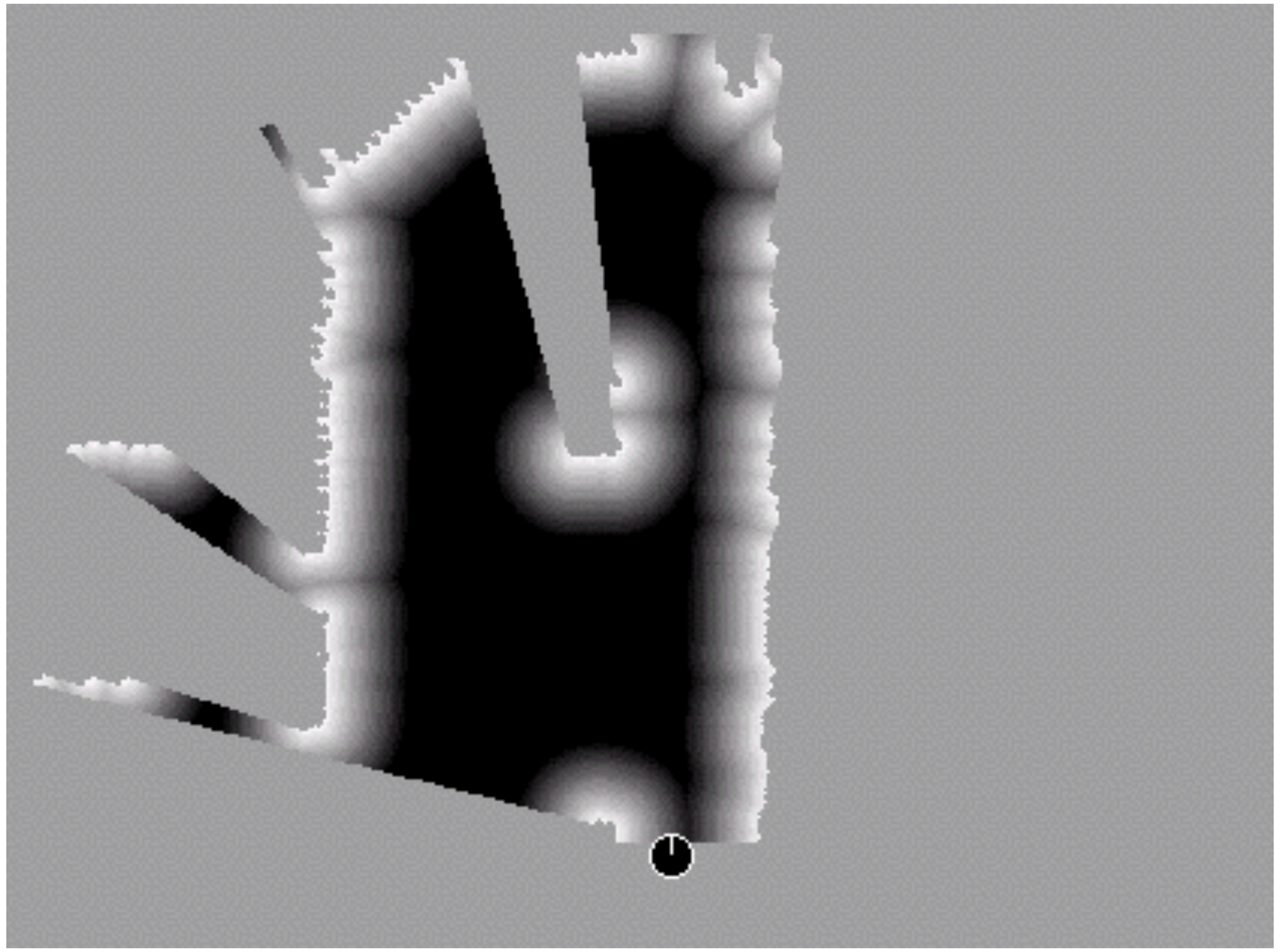


Likelihood field

Scan Matching

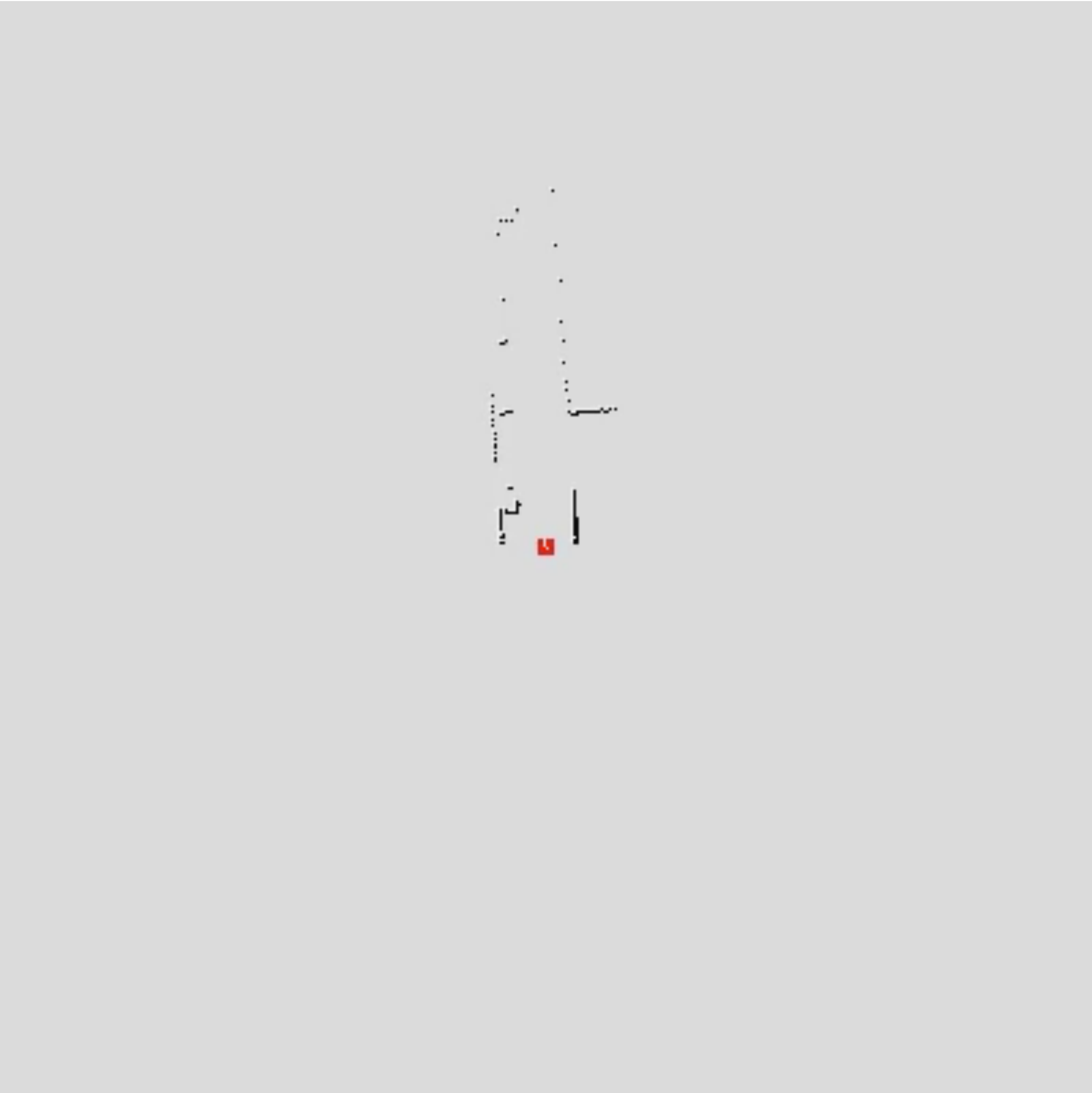
- Extract likelihood field from scan and use it to match different scan.





Scan Matching

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



~0.01 sec

Properties of Scan-based Model

- Highly efficient, uses 2D tables only.
- Smooth w.r.t. to small changes in robot position.
- Allows gradient descent, scan matching.
- Ignores physical properties of beams.
- Works for sonars?

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 using ICP, correlation.

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

Landmarks

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

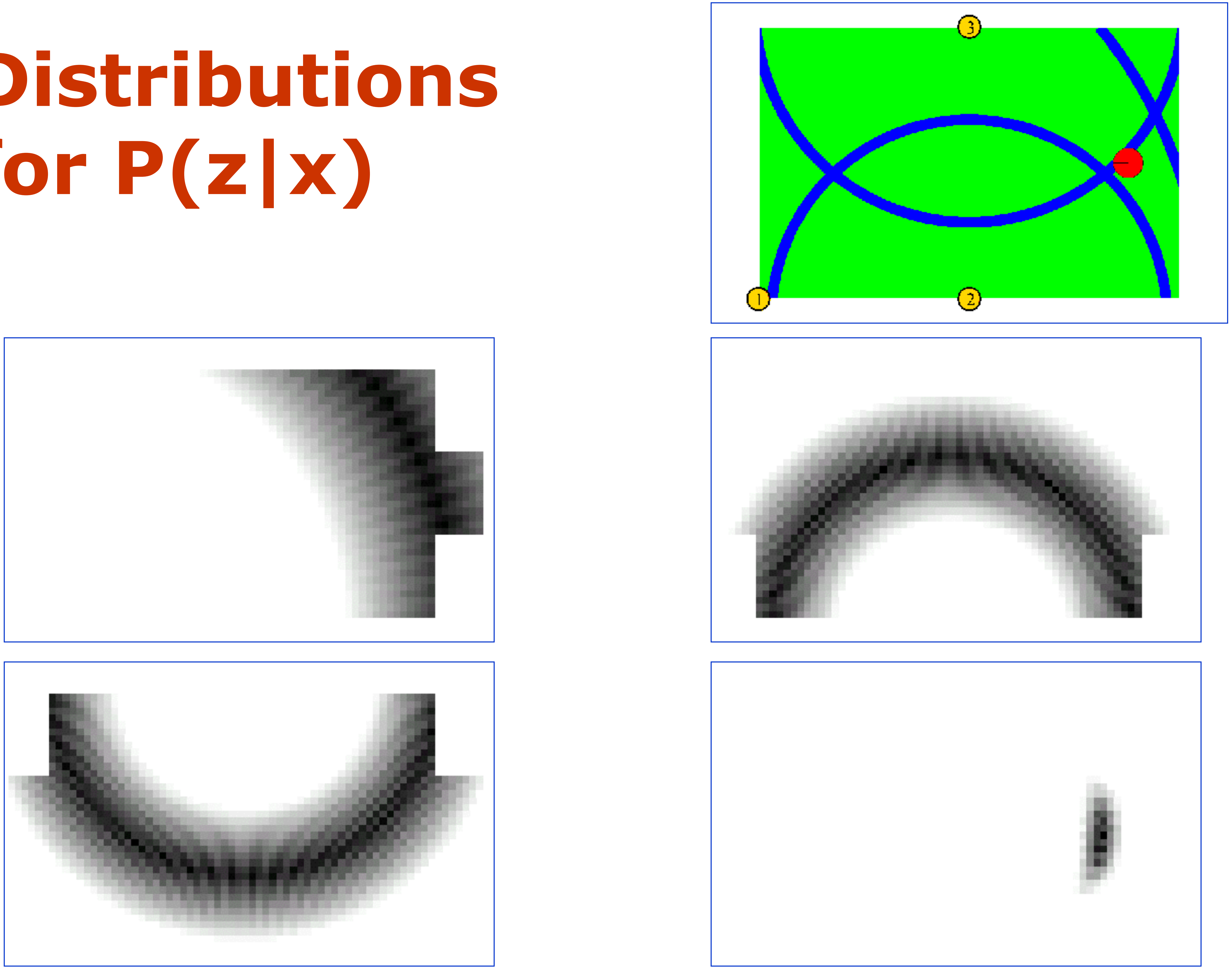
- Sensor provides
  - distance, or
  - bearing, or
  - distance and bearing.

Distance and Bearing

Probabilistic Model

1. Algorithm **landmark_detection_model**(z,x,m):
   $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$
2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$
3. $\hat{\alpha} = \text{atan2}(m_y(i) - y, m_x(i) - x) - \theta$
4. $p_{\text{det}} = \text{prob}(\hat{d} - d, \varepsilon_d) \cdot \text{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$
5. Return $z_{\text{det}} p_{\text{det}} + z_{\text{fp}} P_{\text{uniform}}(z \mid x, m)$

Distributions for P(z|x)



Summary of Parametric Motion and Sensor Models

- Explicitly modeling uncertainty in motion and sensing is key to robustness.
- In many cases, good models can be found by the following approach:
  1. Determine parametric model of noise free motion or measurement.
  2. Analyze sources of noise.
  3. Add adequate noise to parameters (eventually mix 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.
- It is important to be aware of the underlying assumptions!

Next Lecture
Mobile Robotics - III - Kalman