

# Visual Navigation for Flying Robots

## **Probabilistic Models and State Estimation**

Dr. Jürgen Sturm

#### Scientific Research

Be Creative & Do Research

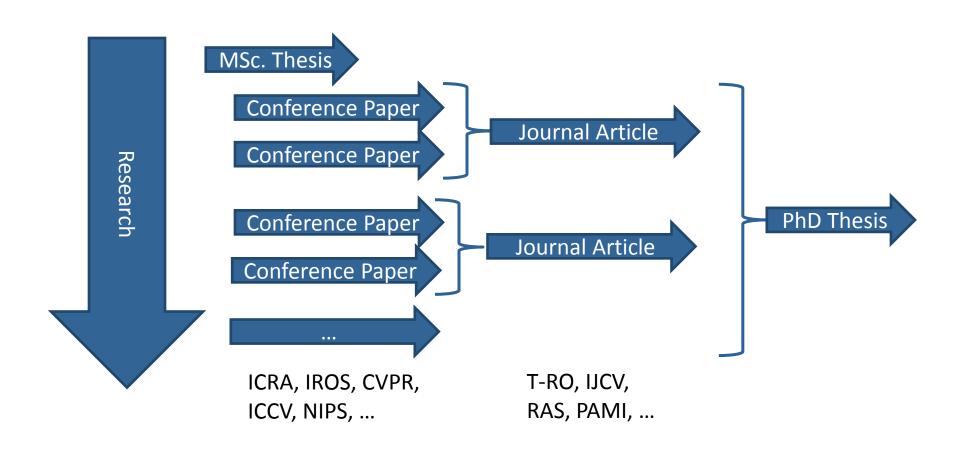
Talk with People & Get Inspired

Write and Submit Paper

Present Work at Conference

Prepare Talk/Poster

#### Flow of Publications



## **Important Conferences**

#### Robotics

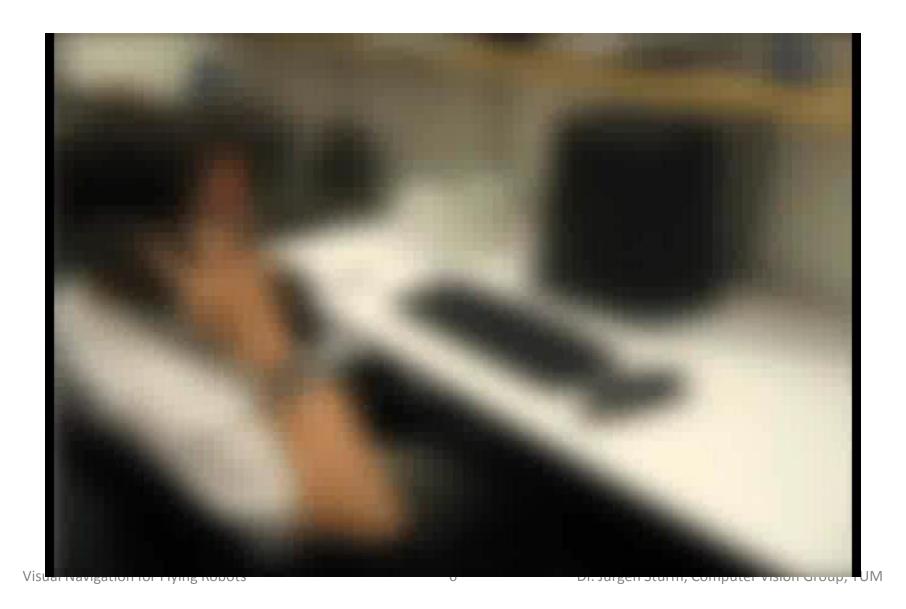
- International Conference on Robotics and Automation (ICRA) 
   next week
- International Conference on Intelligent Robots and Systems (IROS)
- Robotics: Science and Systems (RSS)

#### Computer Vision

- International Conference on Computer Vision (ICCV)
- Computer Vision and Pattern Recognition (CVPR)
- German Conference on Computer Vision (GCPR)

#### **ICRA: Preview**

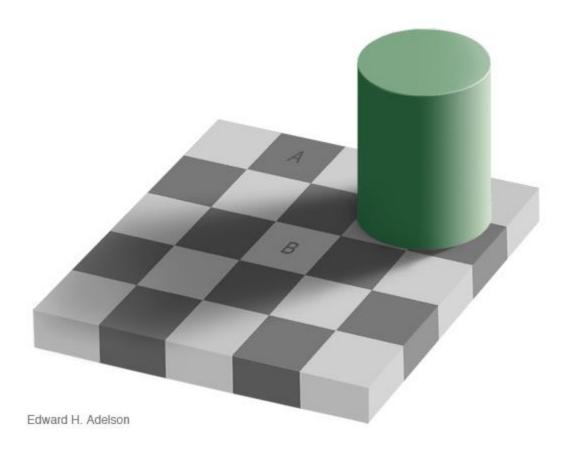
- Christian's work got nominated for the "Best Vision Paper"
- Four sessions (with 6 papers each) on flying robots
- Interesting sessions:
  - Localization
  - Theory and Methods for SLAM
  - Visual Servoing
  - Sensor Fusion
  - Trajectory Planning for Aerial Robots
  - Novel Aerial Robots
  - Aerial Robots and Manipulation
  - Modeling & Control of Aerial Robots
  - Sensing for Navigation
  - •
- Will report on interesting papers!



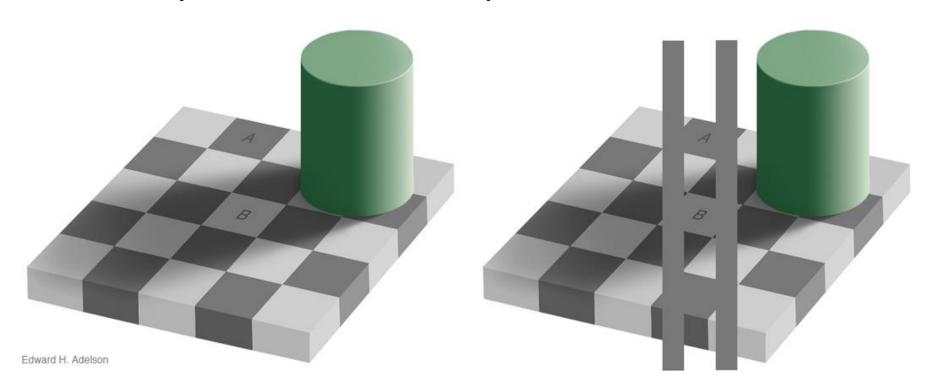


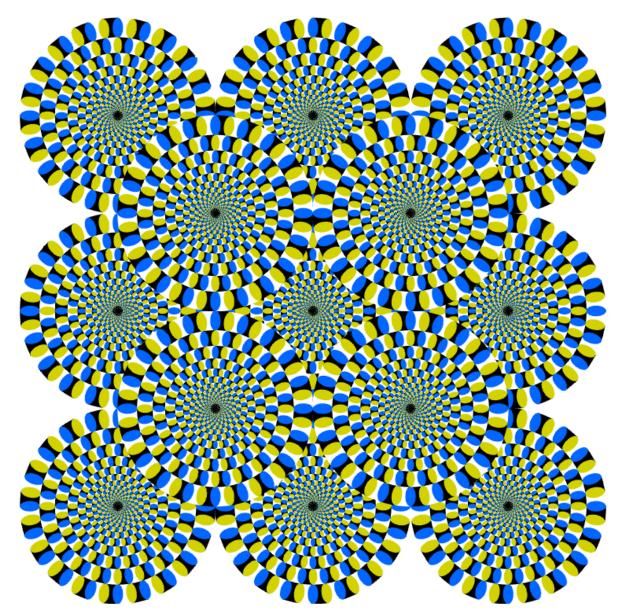
Visual Navigation for Flying Robots

Perception and models are strongly linked



- Perception and models are strongly linked
- Example: Human Perception

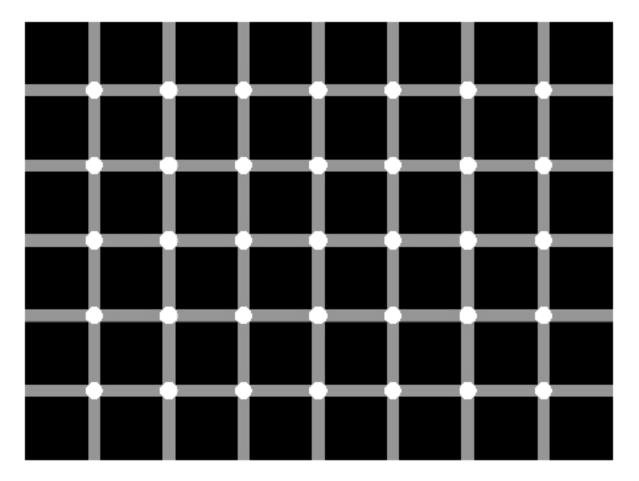




more on http://michaelbach.de/ot/index.html

## **Models in Human Perception**

Count the black dots



#### **State Estimation**

- Cannot observe world state directly
- Need to estimate the world state
- Robot maintains belief about world state
- Update belief according to observations and actions using models
- Sensor observations > sensor model
- Executed actions action/motion model

#### **State Estimation**

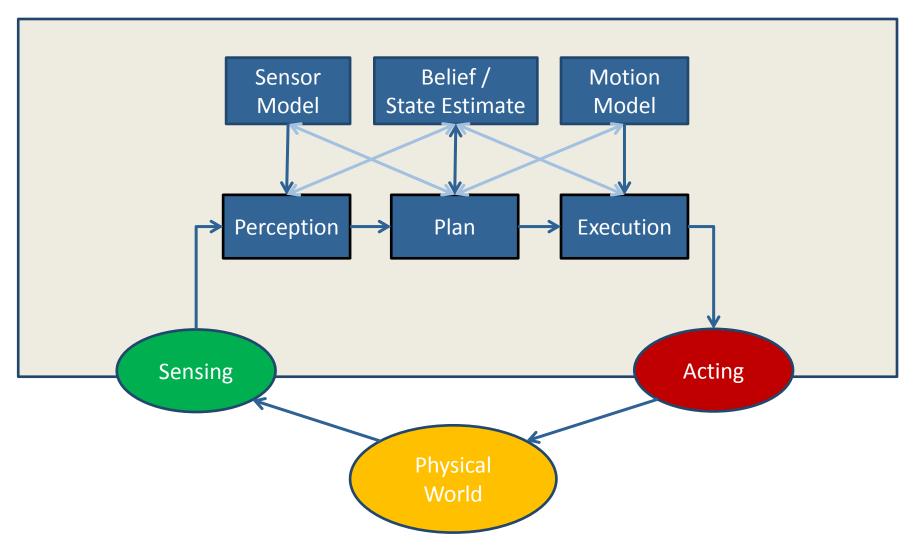
What parts of the world state are (most) relevant for a flying robot?

#### **State Estimation**

What parts of the world state are (most) relevant for a flying robot?

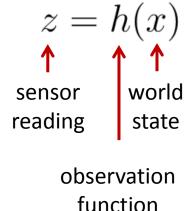
- Position
- Velocity
- Obstacles
- Map
- Positions and intentions of other robots/humans
- • •

#### **Models and State Estimation**



## (Deterministic) Sensor Model

Robot perceives the environment through its sensors



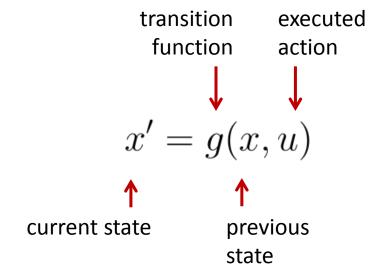
Goal: Infer the state of the world from sensor readings

$$x = h^{-1}(z)$$

## (Deterministic) Motion Model

• Robot executes an action u (e.g., move forward at 1m/s)

Update belief state according to motion model



#### **Probabilistic Robotics**

- Sensor observations are noisy, partial, potentially missing (why?)
- All models are partially wrong and incomplete (why?)
- Usually we have prior knowledge (from where?)

#### **Probabilistic Robotics**

Probabilistic sensor and motion models

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

 Integrate information from multiple sensors (multi-modal)

$$p(x \mid z_{\text{vision}}, z_{\text{ultrasound}}, z_{\text{IMU}})$$

Integrate information over time (filtering)

$$p(x \mid z_1, z_2, \dots, z_t)$$
  
 $p(x \mid u_1, z_1, \dots, u_t, z_t)$ 

## **Agenda for Today**

- Motivation ✓
- Bayesian Probability Theory
- Bayes Filter
- Normal Distribution
- Kalman Filter

## The Axioms of Probability Theory

Notation: P(A) refers to the probability that proposition A holds

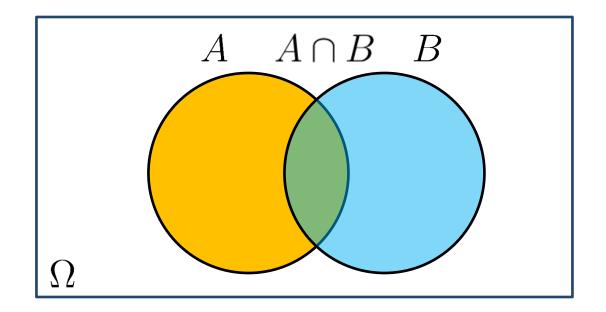
1. 
$$0 \le P(A) \le 1$$

**2.** 
$$P(\Omega) = 1$$
  $P(\emptyset) = 0$ 

3. 
$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

#### A Closer Look at Axiom 3

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



#### **Discrete Random Variables**

- X denotes a random variable
- X can take on a countable number of values in  $\{x_1, x_2, \dots x_n\}$
- $P(X = x_i)$  is the **probability** that the random variable X takes on value  $x_i$
- $P(\cdot)$  is called the **probability mass function**

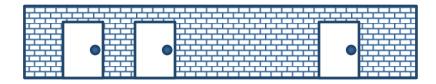
■ Example: P(Room) = < 0.7, 0.2, 0.08, 0.02 > $Room \in \{\text{office, corridor, lab, kitchen}\}$ 

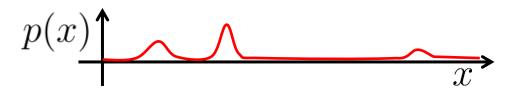
#### **Continuous Random Variables**

- X takes on continuous values
- p(X = x) or p(x) is called the **probability** density function (PDF)

$$P(x \in [a, b]) = \int_{a}^{b} p(x) dx$$

Example





## **Proper Distributions Sum To One**

Discrete case

$$\sum_{x} P(x) = 1$$

Continuous case

$$\int p(x)\mathrm{d}x = 1$$

### **Joint and Conditional Probabilities**

$$P(X = x \text{ and } Y = y) = P(x, y)$$

If X and Y are independent then

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

•  $P(x \mid y)$  is the probability of **x given y** 

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

If X and Y are independent then

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

## **Conditional Independence**

Definition of conditional independence

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

■ Equivalent to  $P(x \mid z) = P(x \mid y, z)$  $P(y \mid z) = P(x \mid x, z)$ 

Note: this does not necessarily mean that

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

## Marginalization

Discrete case

$$P(x) = \sum_{y} P(x, y)$$

Continuous case

$$p(x) = \int p(x, y) dy$$

## **Example: Marginalization**

	$\mathbf{x}_1$	$\mathbf{x}_2$	<b>x</b> <sub>3</sub>	$\mathbf{x}_4$	$p_{y}(Y) \downarrow$
<b>y</b> 1	1/8	1 16	1/32	1/32	$\frac{1}{4}$
<b>Y</b> 2	$\frac{1}{16}$	1/8	1/32	1 32	$\frac{1}{4}$
<b>У</b> 3	1 16	1 16	1 16	$\frac{1}{16}$	1/4
$Y_4$	$\frac{1}{4}$	0	0	0	1/4
$p_{x}(X)$	$\frac{1}{2}$	$\frac{1}{4}$	1/8	18	1

## **Law of Total Probability**

Discrete case

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

Continuous case

$$p(x) = \int p(x, y) dy$$
$$= \int p(x \mid y) p(y) dy$$

## **Expected Value of a Random Variable**

Discrete case

$$E[X] = \sum_{i} x_i P(x_i)$$

Continuous case

$$E[X] = \int xP(X=x)\mathrm{d}x$$

- The expected value is the weighted average of all values a random variable can take on.
- Expectation is a linear operator

$$E[aX + b] = aE[X] + b$$

#### **Covariance of a Random Variable**

 Measures the squared expected deviation from the mean

$$Cov[X] = E[X - E[X]]^2 = E[X^2] - E[X]^2$$

#### **Estimation from Data**

• Observations  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 

• Sample Mean 
$$\mu = \frac{1}{n} \sum_{i} \mathbf{x}_{i}$$

Sample Covariance

$$\Sigma = \frac{1}{n-1} \sum_{i} (\mathbf{x}_i - \boldsymbol{\mu}) (\mathbf{x}_i - \boldsymbol{\mu})^{\top}$$

#### The State Estimation Problem

We want to estimate the world state x from

- 1. Sensor measurements z and
- 2. Controls (or odometry readings) u

We need to model the relationship between these random variables, i.e.,

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

## Causal vs. Diagnostic Reasoning

- $P(x \mid z)$  is diagnostic
- $P(z \mid x)$  is causal
- Often causal knowledge is easier to obtain
- Bayes rule allows us to use causal knowledge:

observation likelihood prior on world state

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

prior on sensor observations

## **Bayes Formula**

Derivation of Bayes Formula

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

Our usage

$$P(x \mid z) = \frac{P(z \mid x)P(x)}{P(z)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

# **Bayes Formula**

Derivation of Bayes Formula

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

Our usage

$$P(x \mid z) = \frac{P(z \mid x)P(x)}{P(z)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

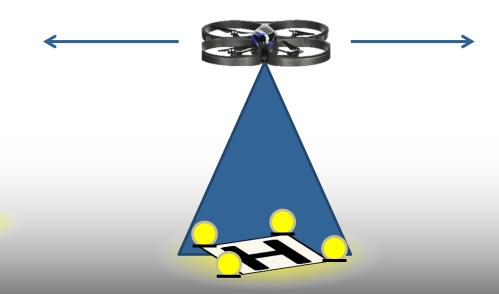
#### **Normalization**

- Direct computation of P(z) can be difficult
- Idea: Compute improper distribution, normalize afterwards
- Step 1:  $L(x \mid z) = P(z \mid x)P(x)$
- Step 2:  $P(z) = \sum_{x} P(z \mid x) P(x) = \sum_{x} L(x \mid z)$  (Law of total probability)
- Step 3:  $P(x \mid z) = L(x \mid z)/P(z)$

#### **Bayes Rule with Background Knowledge**

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

- Quadrocopter seeks the landing zone
- Landing zone is marked with many bright lamps
- Quadrocopter has a brightness sensor



- Binary sensor  $Z \in \{ bright, \neg bright \}$
- Binary world state  $X \in \{\text{home}, \neg \text{home}\}$
- Sensor model  $P(Z = \text{bright} \mid X = \text{home}) = 0.6$  $P(Z = \text{bright} \mid X = \neg \text{home}) = 0.3$
- Prior on world state P(X = home) = 0.5
- Assume: Robot observes light, i.e., Z = bright
- What is the probability  $P(X = \text{home} \mid Z = \text{bright})$  that the robot is above the landing zone?

- Sensor model  $P(Z = \text{bright} \mid X = \text{home}) = 0.6$  $P(Z = \text{bright} \mid X = \neg \text{home}) = 0.3$
- Prior on world state P(X = home) = 0.5
- Probability after observation (using Bayes)

$$P(X = \text{home} \mid Z = \text{bright})$$

$$= \frac{P(\text{bright} \mid \text{home})P(\text{home})}{P(\text{bright} \mid \text{home})P(\text{home}) + P(\text{bright} \mid \neg \text{home})P(\neg \text{home})}$$

$$= \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{0.3}{0.3 + 0.15} = 0.67$$

- Sensor model  $P(Z = \text{bright} \mid X = \text{home}) = 0.6$  $P(Z = \text{bright} \mid X = \neg \text{home}) = 0.3$
- Prior on world state P(X = home) = 0.5
- Probability after observation (using Bayes)

$$P(X = \text{home} \mid Z = \text{bright})$$

$$= \frac{P(\text{bright} \mid \text{home})P(\text{home})}{P(\text{bright} \mid \text{home})P(\text{home}) + P(\text{bright} \mid \neg \text{home})P(\neg \text{home})}$$

$$= \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{0.3}{0.3 + 0.15} = \frac{0.67}{0.67}$$

# **Combining Evidence**

- Suppose our robot obtains another observation  $z_2$  (either from the same or a different sensor)
- How can we integrate this new information?
- More generally, how can we estimate  $p(x \mid z_1, z_2, \dots)$ ?

# **Combining Evidence**

- Suppose our robot obtains another observation  $z_2$  (either from the same or a different sensor)
- How can we integrate this new information?
- More generally, how can we estimate  $p(x \mid z_1, z_2, \dots)$ ?
- Bayes formula gives us:

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

#### **Recursive Bayesian Updates**

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

#### **Recursive Bayesian Updates**

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

#### Markov Assumption:

 $z_n$  is independent of  $z_1, \ldots, z_{n-1}$  if we know x

#### **Recursive Bayesian Updates**

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

#### Markov Assumption:

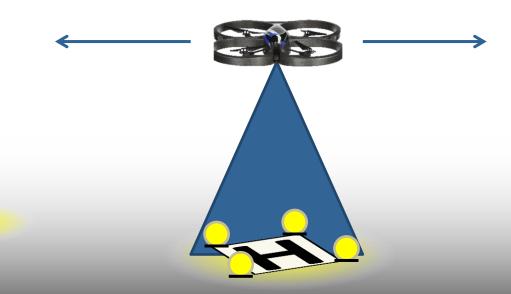
 $z_n$  is independent of  $z_1, \ldots, z_{n-1}$  if we know x

$$\Rightarrow P(x \mid z_{1}, \dots, z_{n}) = \frac{P(z_{n} \mid x) P(x \mid z_{1}, \dots, z_{n-1})}{P(z_{n} \mid z_{1}, \dots, z_{n-1})}$$

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

$$= \eta_{1:n} \prod_{i=1,\dots,n} P(z_{i} \mid x) P(x)$$

- Quadrocopter seeks the landing zone
- Landing zone is marked with many bright lamps and a visual marker



#### **Example: Second Measurement**

- Sensor model  $P(Z_2 = \text{marker} \mid X = \text{home}) = 0.8$   $P(Z_2 = \text{marker} \mid X = \neg \text{home}) = 0.1$
- Previous estimate  $P(X = \text{home} \mid Z_1 = \text{bright}) = 0.67$
- Assume robot does not observe marker
- What is the probability of being home?

$$P(X = \text{home} \mid Z_1 = \text{bright}, Z_2 = \neg \text{marker})$$

$$= \frac{P(\neg \text{marker} \mid \text{home})P(\text{home} \mid \text{bright})}{P(\neg \text{marker} \mid \text{home})P(\text{home} \mid \text{bright}) + P(\neg \text{marker} \mid \neg \text{home})P(\neg \text{home} \mid \text{bright})}$$

$$= \frac{0.2 \cdot 0.67}{0.2 \cdot 0.67 + 0.9 \cdot 0.33} = 0.31$$

#### **Example: Second Measurement**

- Sensor model  $P(Z_2 = \text{marker} \mid X = \text{home}) = 0.8$   $P(Z_2 = \text{marker} \mid X = \neg \text{home}) = 0.1$
- Previous estimate  $P(X = \text{home} \mid Z_1 = \text{bright}) = 0.67$
- Assume robot does not observe marker
- What is the probability of being home?

$$P(X = \text{home} \mid Z_1 = \text{bright}, Z_2 = \neg \text{marker})$$

$$= \frac{P(\neg \text{marker} \mid \text{home})P(\text{home} \mid \text{bright})}{P(\neg \text{marker} \mid \text{home})P(\text{home} \mid \text{bright}) + P(\neg \text{marker} \mid \neg \text{home})P(\neg \text{home} \mid \text{bright})}$$

$$= \frac{0.2 \cdot 0.67}{0.2 \cdot 0.67 + 0.9 \cdot 0.33} = \frac{0.31}{0.31}$$
The second observation lowers the probability that the robot is above the landing zone!

#### **Actions (Motions)**

- Often the world is dynamic since
  - actions carried out by the robot...
  - actions carried out by other agents...
  - or just time passing by...
  - ...change the world

How can we incorporate actions?

#### **Typical Actions**

- Quadrocopter accelerates by changing the speed of its motors
- Position also changes when quadrocopter does "nothing" (and drifts..)

- Actions are never carried out with absolute certainty
- In contrast to measurements, actions generally increase the uncertainty of the state estimate

#### **Action Models**

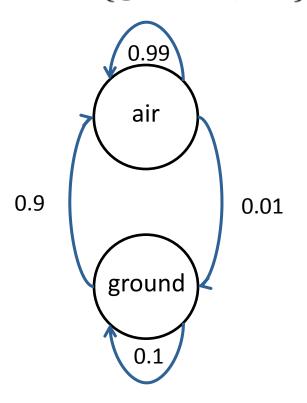
 To incorporate the outcome of an action u into the current state estimate ("belief"), we use the conditional pdf

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

 This term specifies the probability that executing the action u in state x will lead to state x'

# **Example: Take-Off**

- Action:  $u \in \{\text{takeoff}\}$
- World state:  $x \in \{\text{ground}, \text{air}\}$



#### Integrating the Outcome of Actions

Discrete case

$$P(x' \mid u) = \sum_{x} P(x' \mid u, x) P(x)$$

Continuous case

$$p(x' \mid u) = \int p(x' \mid u, x) p(x) dx$$

#### **Example: Take-Off**

- Prior belief on robot state: P(x = ground) = 1.0 (robot is located on the ground)
- Robot executes "take-off" action
- What is the robot's belief after one time step?

$$P(x' = \text{ground}) = \sum_{x} P(x' = \text{ground} \mid u, x) P(x)$$

$$= P(x' = \text{ground} \mid u, x = \text{ground}) P(x = \text{ground})$$

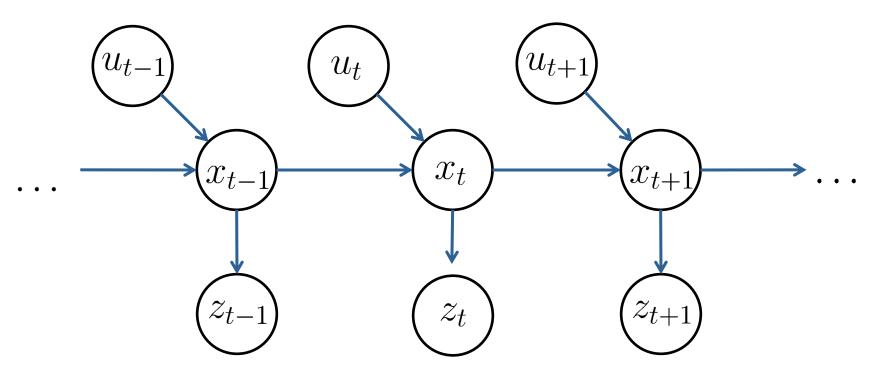
$$+ P(x' = \text{ground} \mid u, x = \text{air}) P(x = \text{air})$$

$$= 0.1 \cdot 1.0 + 0.01 \cdot 0.0 = 0.1$$

• Question: What is the probability at t=2?

#### **Markov Chain**

 A Markov chain is a stochastic process where, given the present state, the past and the future states are independent



# **Markov Assumption**

Observations depend only on current state

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

 Current state depends only on previous state and current action

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

- Underlying assumptions
  - Static world
  - Independent noise
  - Perfect model, no approximation errors

# **Bayes Filter**

#### Given:

• Stream of observations z and actions u:

$$\mathbf{d}_t = (u_1, z_1, \dots, u_t, z_t)^{\top}$$

- Sensor model  $P(z \mid x)$
- Action model  $P(x' \mid x, u)$
- Prior probability of the system state P(x)
- Wanted:
  - Estimate of the state x of the dynamic system
  - Posterior of the state is also called belief

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

# **Bayes Filter**

#### For each time step, do

1. Apply motion model

$$\overline{\operatorname{Bel}}(x_t) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t) \operatorname{Bel}(x_{t-1})$$

Apply sensor model

$$Bel(x_t) = \eta P(z_t \mid x_t) \overline{Bel}(x_t)$$

Note: Bayes filters also work on continuous state spaces (replace sum by integral)

# **Bayes Filter**

#### For each time step, do

1. Apply motion model

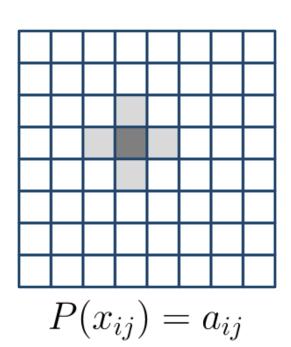
$$\overline{\operatorname{Bel}}(x_t) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t) \operatorname{Bel}(x_{t-1})$$

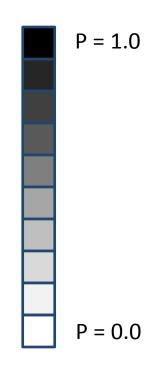
Apply sensor model

$$Bel(x_t) = \eta P(z_t \mid x_t) \overline{Bel}(x_t)$$

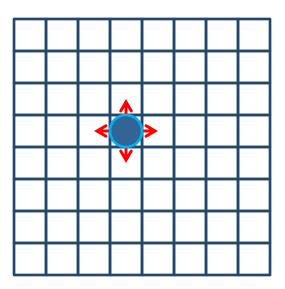
Second note: Bayes filter also works when actions and observations are asynchronous!

- Discrete state  $x \in \{1, 2, ..., w\} \times \{1, 2, ..., h\}$
- Belief distribution can be represented as a grid
- This is also called a histogram filter





- **Action**  $u \in \{\text{north}, \text{east}, \text{south}, \text{west}\}$
- Robot can move one cell in each time step
- Actions are not perfectly executed

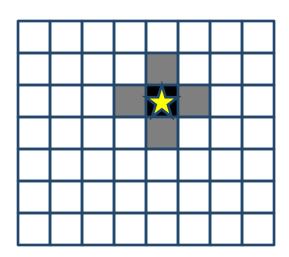


- **Action**  $u \in \{\text{north}, \text{east}, \text{south}, \text{west}\}$
- Robot can move one cell in each time step
- Actions are not perfectly executed
- Example: move east

$$x_{t-1} = \bigcirc$$
,  $u = \text{east} \Rightarrow \bigcirc$ 

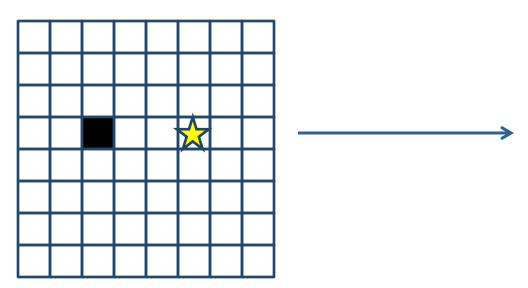
60% success rate, 10% to stay/move too far/move one up/move one down

- Binary observation  $z \in \{\text{marker}, \neg\text{marker}\}$
- One (special) location has a marker
- Marker is sometimes also detected in neighboring cells

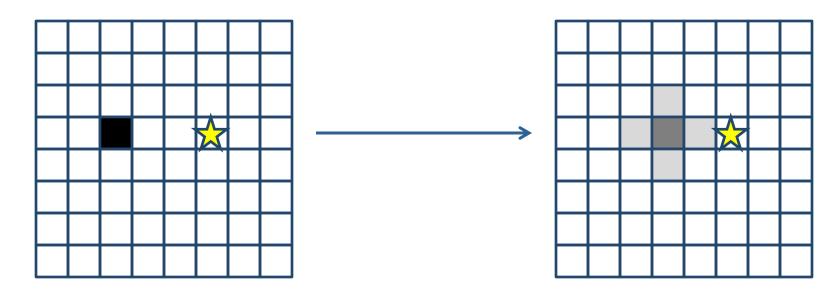


 Let's start a simulation run... (shades are handdrawn, not exact!)

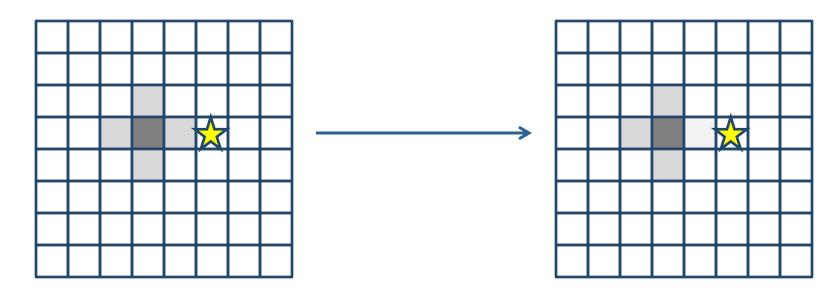
- t=0
- Prior distribution (initial belief)
- Assume we know the initial location (if not, we could initialize with a uniform prior)



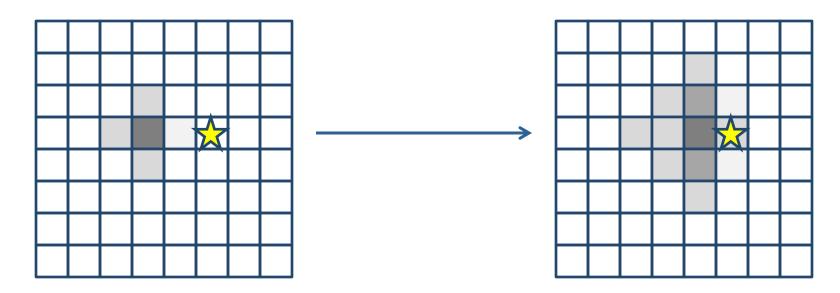
- t=1, u=east, z=no-marker
- Bayes filter step 1: Apply motion model



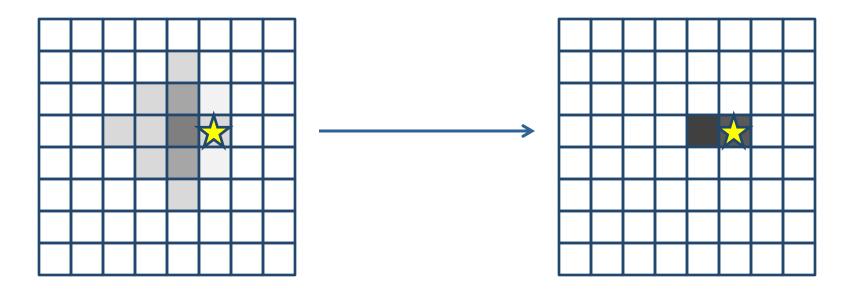
- t=1, u=east, z=no-marker
- Bayes filter step 2: Apply observation model



- t=2, u=east, z=marker
- Bayes filter step 2: Apply motion model



- t=2, u=east, z=marker
- Bayes filter step 1: Apply observation model
- Question: Where is the robot?



## **Bayes Filter - Summary**

- Markov assumption allows efficient recursive Bayesian updates of the belief distribution
- Useful tool for estimating the state of a dynamic system
- Bayes filter is the basis of many other filters
  - Kalman filter
  - Particle filter
  - Hidden Markov models
  - Dynamic Bayesian networks
  - Partially observable Markov decision processes (POMDPs)

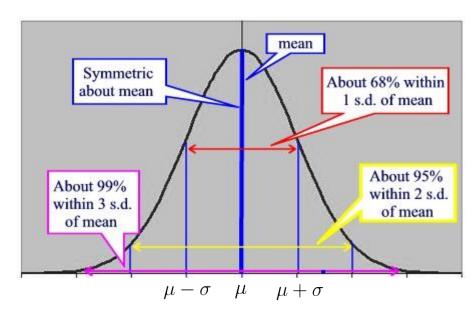
### Kalman Filter

- Bayes filter with continuous states
- State represented with a normal distribution
- Developed in the late 1950's
- Kalman filter is very efficient (only requires a few matrix operations per time step)
- Applications range from economics, weather forecasting, satellite navigation to robotics and many more
- Most relevant Bayes filter variant in practice
   → exercise sheet 2

### **Normal Distribution**

Univariate normal distribution

$$p(X = x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right)$$



### **Normal Distribution**

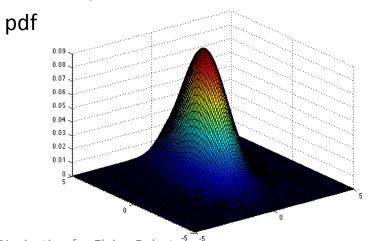
Multivariate normal distribution

$$X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

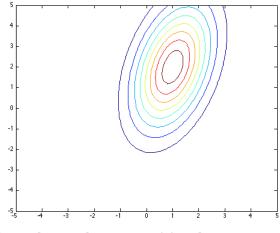
$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \frac{1}{(2\pi)^{d/2}} \sum_{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

Example: 2-dimensional normal distribution



iso lines



Dr. Jürgen Sturm, Computer Vision Group, TUM

## **Properties of Normal Distributions**

■ Linear transformation → remains Gaussian

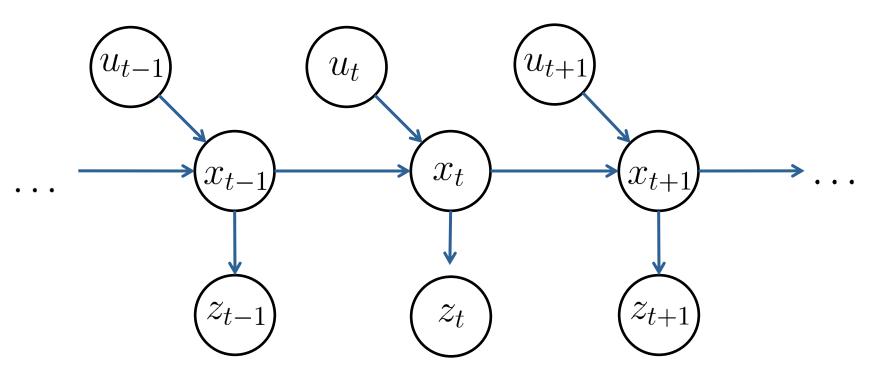
$$X \sim \mathcal{N}(\mu, \Sigma), Y \sim AX + B$$
  
 $\Rightarrow Y \sim \mathcal{N}(A\mu + B, A\Sigma A^{\top})$ 

■ Intersection of two Gaussians → remains Gaussian

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_1), X_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$$

$$\Rightarrow p(X_1, X_2) = \mathcal{N}\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}\right)$$

 Consider a time-discrete stochastic process (Markov chain)



- Consider a time-discrete stochastic process
- Represent the estimated state (belief) by a Gaussian  $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$

- Consider a time-discrete stochastic process
- Represent the estimated state (belief) by a Gaussian  $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$
- Assume that the system evolves linearly over time, then

$$x_t = Ax_{t-1}$$

- Consider a time-discrete stochastic process
- Represent the estimated state (belief) by a Gaussian  $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$
- Assume that the system evolves linearly over time and depends linearly on the controls

$$x_t = Ax_{t-1} + Bu_t$$

- Consider a time-discrete stochastic process
- Represent the estimated state (belief) by a Gaussian  $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$
- Assume that the system evolves linearly over time, depends linearly on the controls, and has zero-mean, normally distributed process noise

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

with 
$$\epsilon_t \sim \mathcal{N}(0, Q)$$

### **Linear Observations**

 Further, assume we make observations that depend linearly on the state

$$z_t = Cx_t$$

### **Linear Observations**

Further, assume we make observations that depend linearly on the state and that are perturbed by zero-mean, normally distributed observation noise

$$z_t = Cx_t + \delta_t$$

with 
$$\delta_t \sim \mathcal{N}(0,R)$$

### Kalman Filter

Estimates the state  $x_t$  of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

and (linear) measurements of the state

$$z_t = Cx_t + \delta_t$$

with  $\delta_t \sim \mathcal{N}(0,R)$  and  $\epsilon_t \sim \mathcal{N}(0,Q)$ 

### **Variables and Dimensions**

- State  $x \in \mathbb{R}^n$
- Controls  $u \in \mathbb{R}^l$
- Observations  $z \in \mathbb{R}^k$
- Process equation

$$x_t = \underbrace{A}_{n \times n} x_{t-1} + \underbrace{B}_{n \times l} u_t + \epsilon$$

Measurement equation

$$z_t = \underbrace{C}_{n \times k} x_t + \delta_t$$

### Kalman Filter

Initial belief is Gaussian

$$Bel(x_0) = \mathcal{N}(x_0; \mu_0, \Sigma_0)$$

Next state is also Gaussian (linear transformation)

$$x_t \sim \mathcal{N}(Ax_{t-1} + Bu_t, Q)$$

Observations are also Gaussian

$$z_t \sim \mathcal{N}(Cx_t, R)$$

## From Bayes Filter to Kalman Filter

#### For each time step, do

Apply motion model

$$\overline{\operatorname{Bel}}(x_t) = \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\mathcal{N}(x_t; Ax_t + Bu_t, Q)} \underbrace{\operatorname{Bel}(x_{t-1})}_{\mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} dx_{t-1}$$

## From Bayes Filter to Kalman Filter

#### For each time step, do

#### Apply motion model

$$\overline{\operatorname{Bel}}(x_t) = \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\mathcal{N}(x_t; Ax_{t-1} + Bu_t, Q)} \underbrace{\operatorname{Bel}(x_{t-1})}_{\mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})} dx_{t-1}$$

$$= \mathcal{N}(x_t; A\mu_{t-1} + Bu_t, A\Sigma A^{\top} + Q)$$

$$=\mathcal{N}(x_t;\bar{\mu}_t,\bar{\Sigma}_t)$$

## From Bayes Filter to Kalman Filter

#### For each time step, do

#### 2. Apply sensor model

$$Bel(x_t) = \eta \underbrace{p(z_t \mid x_t)}_{\mathcal{N}(z_t; Cx_t, R)} \underbrace{\overline{Bel}(x_t)}_{\mathcal{N}(x_t; \bar{\mu}_t, \bar{\Sigma}_t)}$$

$$= \mathcal{N}(x_t; \bar{\mu}_t + K_t(z_t - C\bar{\mu}), (I - K_tC)\bar{\Sigma})$$

$$= \mathcal{N}(x_t; \mu_t, \Sigma_t)$$

with 
$$K_t = \bar{\Sigma}_t C^{\top} (C\bar{\Sigma}_t C^{\top} + R)^{-1}$$

### Kalman Filter

#### For each time step, do

#### Apply motion model

For the interested readers: See Probabilistic Robotics for full derivation (Chapter 3)

$$\bar{\mu}_t = A\mu_{t-1} + Bu_t$$
$$\bar{\Sigma}_t = A\Sigma A^\top + Q$$

### 2. Apply sensor model

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\bar{\mu}_t)$$
  
$$\Sigma_t = (I - K_t C)\bar{\Sigma}_t$$

with 
$$K_t = \bar{\Sigma}_t C^{\top} (C \bar{\Sigma}_t C^{\top} + R)^{-1}$$

### Kalman Filter

Highly efficient: Polynomial in the measurement dimensionality k and state dimensionality n:

$$O(k^{2.376} + n^2)$$

- Optimal for linear Gaussian systems!
- Most robotics systems are nonlinear!

## **Nonlinear Dynamical Systems**

- Most realistic robotic problems involve nonlinear functions
- Motion function

$$x_t = g(u_t, x_{t-1})$$

Observation function

$$z_t = h(x_t)$$

## **Taylor Expansion**

- Solution: Linearize both functions
- Motion function

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1})$$
$$= g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1})$$

Observation function

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t} (x_t - \mu_t)$$
$$= h(\bar{\mu}_t) + H_t(x_t - \mu_t)$$

### **Extended Kalman Filter**

#### For each time step, do

#### Apply motion model

For the interested readers: See Probabilistic Robotics for full derivation (Chapter 3)

$$ar{\mu}_t = g(\mu_{t-1}, u_t)$$
 $ar{\Sigma}_t = G_t \Sigma G_t^\top + Q$  with  $G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$ 

#### 2. Apply sensor model

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$
  
$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

with 
$$K_t = \bar{\Sigma}_t H_t^{\top} (H_t \bar{\Sigma}_t H_t^{\top} + R)^{-1}$$
 and  $H_t = \frac{\partial h(\mu_t)}{\partial x_t}$ 

- 2D case
- State  $\mathbf{x} = \begin{pmatrix} x & y & \psi \end{pmatrix}^{\mathsf{T}}$
- Odometry  $\mathbf{u} = (\dot{x} \ \dot{y} \ \dot{\psi})^{\top}$
- Observations of visual marker  $\mathbf{z} = \begin{pmatrix} x & y & \psi \end{pmatrix}^{\top}$  (relative to robot pose)
- Fixed time intervals  $\Delta t$

Motion function

$$g(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} x + (\cos(\psi)\dot{x} - \sin(\psi)\dot{y})\Delta t \\ y + (\sin(\psi)\dot{x} + \cos(\psi)\dot{y})\Delta t \\ \psi + \dot{\psi}\Delta t \end{pmatrix}$$

Derivative of motion function

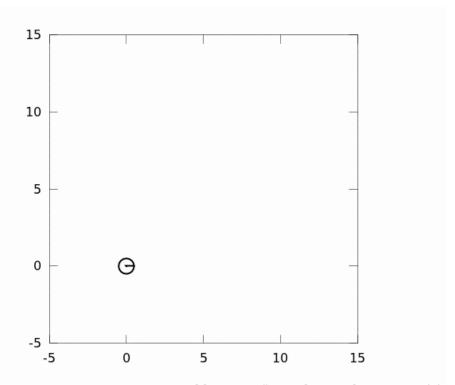
$$G = \frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} = \begin{pmatrix} 1 & 0 & (-\sin(\psi)\dot{x} - \cos(\psi)\dot{y})\Delta t \\ 0 & 1 & (\cos(\psi)\dot{x} + \sin(\dot{\psi})\dot{y})\Delta t \\ 0 & 0 & 1 \end{pmatrix}$$

■ Observation Function (→ Sheet 2)

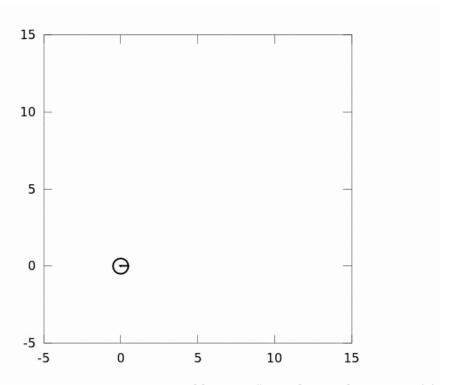
$$h(\mathbf{x}) = \dots$$

$$H = \frac{\partial h(\mathbf{x})}{\partial x} = \dots$$

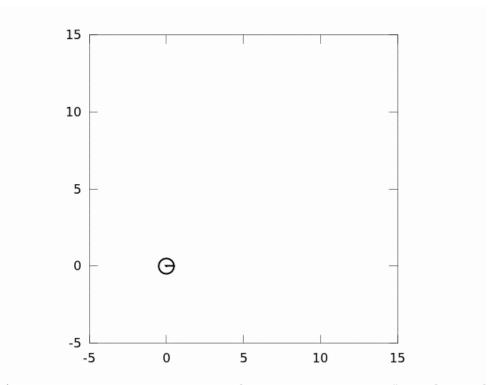
- Dead reckoning (no observations)
- Large process noise in x+y



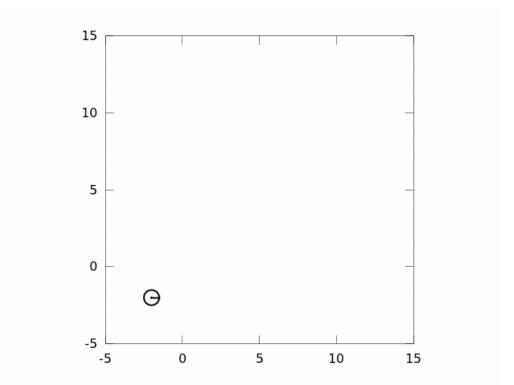
- Dead reckoning (no observations)
- Large process noise in x+y+yaw



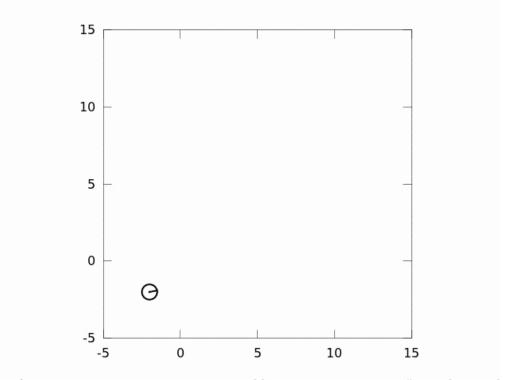
- Now with observations (limited visibility)
- Assume robot knows correct starting pose



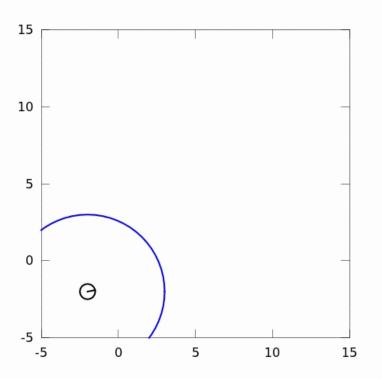
What if the initial pose (x+y) is wrong?

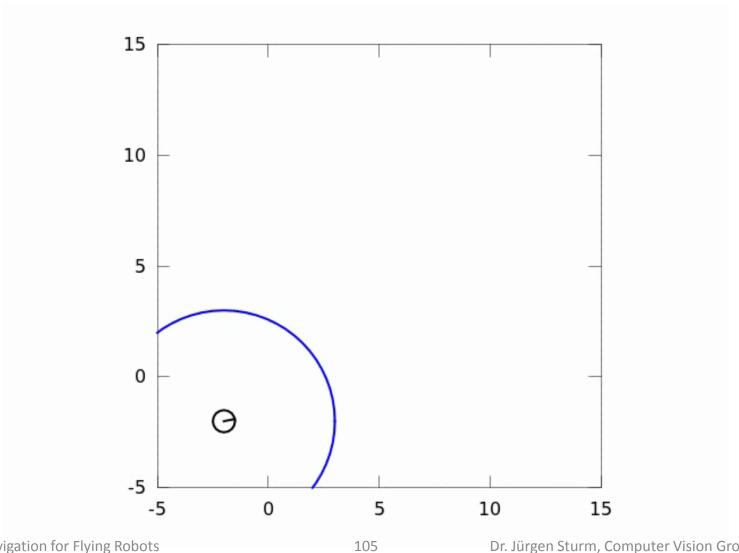


What if the initial pose (x+y+yaw) is wrong?



 If we are aware of a bad initial guess, we set the initial sigma to a large value (large uncertainty)





## **Lessons Learned Today**

- Observations and actions are inherently noisy
- Knowledge about state is inherently uncertain
- Probability theory
- Probabilistic sensor and motion models
- Bayes Filter, Histogram Filter, Kalman Filter, Examples