Introduction to Bayesian tracking

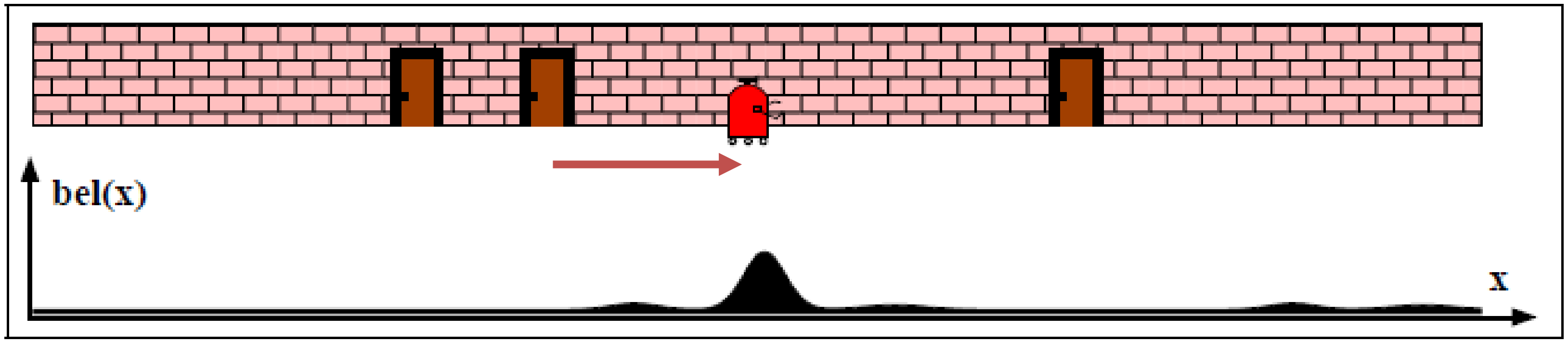
Doz. G. Bleser

Prof. Stricker

Computer Vision: Object and People Tracking

Reminder: introductory example

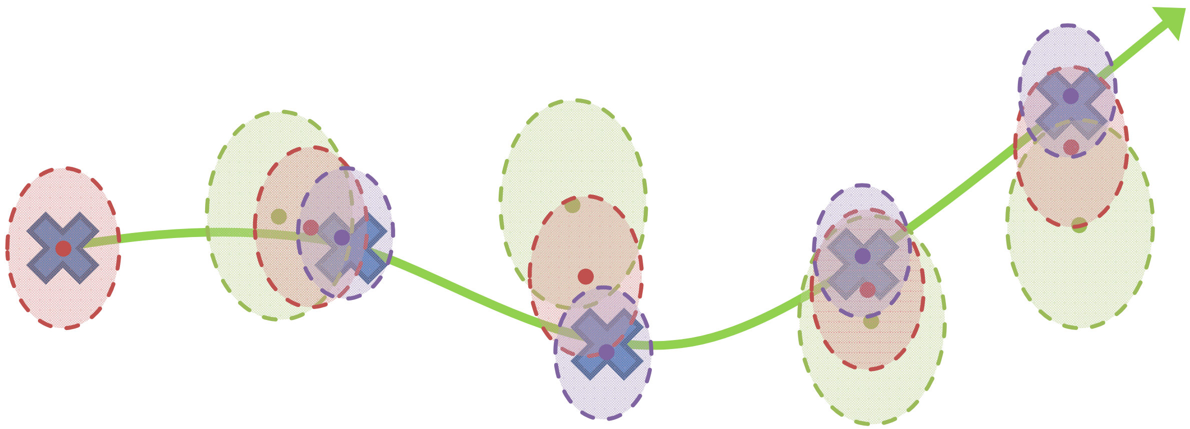
(e)



- Goal:
  - Estimate **posterior belief** or **state distribution** based on **control data** and **sensor measurements**
  - Information represented as **probability density function**
- State: position of robot in corridor
- Control data: odometry
- Measurements: door sensors

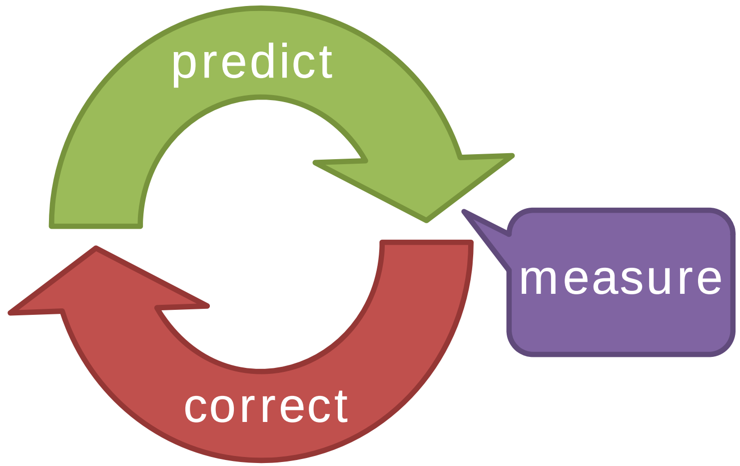
Recursive Bayesian filtering

- **Key idea 1**: Probability distributions represent our belief about the state of the dynamical system



- **Key idea 2**: Recursive cycle
    1. Predict from motion model
    2. Sensor measurement
    3. Correct the prediction

    ...repeat



Outline

- Reminder: Basic concepts in probability
- Terminology, notation, probabilistic laws
- Bayes filters

Joint and Conditional Probability

- Let X and Y denote two random variables:

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

- If X and Y are independent (carry no information about each other) then:

$$\mathrm{p}(x, y) = p(x)p(y)$$
$$p(x|y) = p(x)$$

Joint and Conditional Probability: example

- Ideal cube, dice toss: G={2, 4, 6}, A={4, 5, 6}
- $p(G) =?$
- $p(A) =?$
- $p(G, A) =?$
- $p(G|A) =?$

Joint and Conditional Probability: example

- Ideal cube, dice toss: G={2, 4, 6}, A={4, 5, 6}
- $p(G) = \frac{1}{2}$
- $p(A) = \frac{1}{2}$
- $p(G, A) = p(\{4,6\}) = \frac{1}{3}$
- $p(G|A) = \frac{p(G,A)}{p(A)} = 2\, p(\{4,6\}) = \frac{2}{3}$

Theorem of Total Probability, Marginals

Discrete case

$$\sum_x p(x) = 1$$

$$p(x) = \sum_y p(x, y)$$

$$p(x) = \sum_y p(x|y)p(y)$$

Continuous case

$$\int_{-\infty}^{\infty} p(x) = 1$$

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

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

Bayes Theorem

- In the context of state estimation:
  - Assume x is a quantity that we want to infer from y
    ➔ Think of x as state and y as sensor measurement

Generative model: how state variables cause sensor measurements

$$\boxed{p(x|y)} = \frac{\boxed{p(y|x)}\,p(x)}{\boxed{p(y)}} = \frac{likelihood \cdot prior}{evidence}$$

Posterior probability

Independent of x
➔ denoted as normalizer η

Important: in this lecture, we will freely use η in different equations to denote normalizers, even if their actual values differ!

Conditional Independence

- All rules presented so far can be conditioned on an arbitrary random variable, e.g. Z

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

$$p(x,y|z) = p(x|y,z)p(y|z)$$

- Conditional Independence: x and y are independent, given that z is known

y carries no information about x, if z is known

$$p(x,y|z) = p(x|z)p(y|z)$$
$$\Leftrightarrow p(x|z) = p(x|z,y), p(y|z) = p(y|z,x)$$

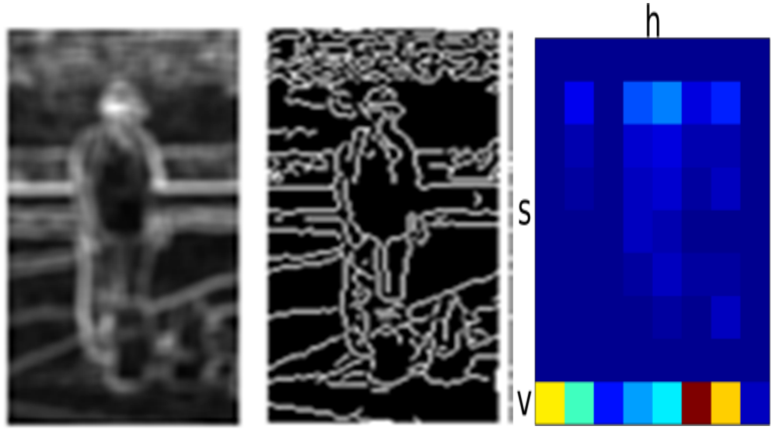
➔ does <u>not</u> necessarily imply absolute independence

Towards the state estimation problem: terminology, notation, probabilistic laws

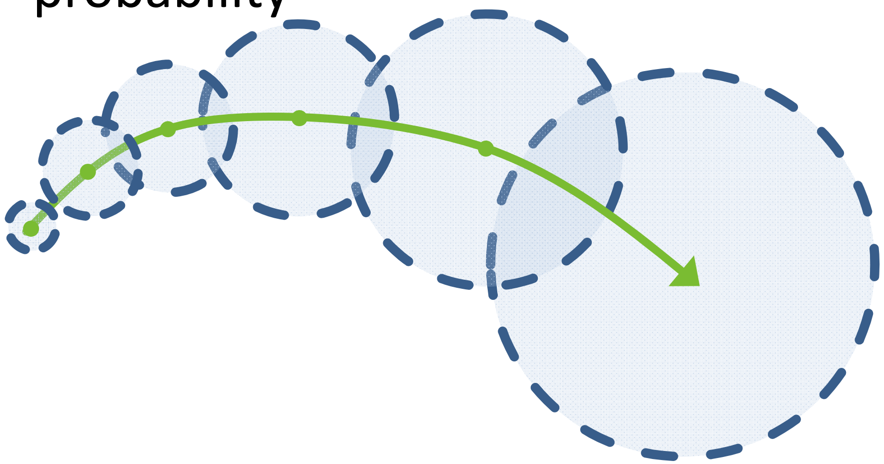
Terminology: overview

- State

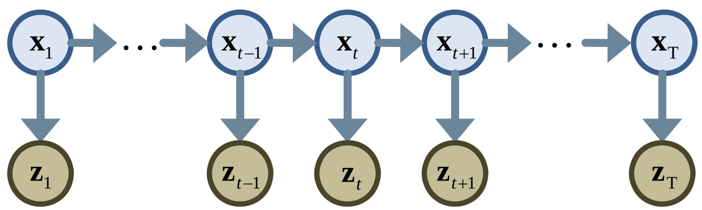
- Measurement model/likelihood



- Motion model/transition probability



- Inference



State

- The environment or world is considered a dynamical system
- The state contains all information that we want to know about the system
- Notation: x_t denotes the state at time t
- A state is called complete, if it is the best predictor for the future
  ➔ knowledge of past states, measurements, or controls carries no information about evolution of the state in the future

Markov chain

- Typical examples of states:
  - Object pose/velocity in global coordinate system ➔ continuous, dynamic
  - 3D positions of landmarks ➔ continuous, stationary
  - Whether a sensor is broken or not ➔ discrete, dynamic
  - Sensor biases ➔ discrete, stationary

State: examples

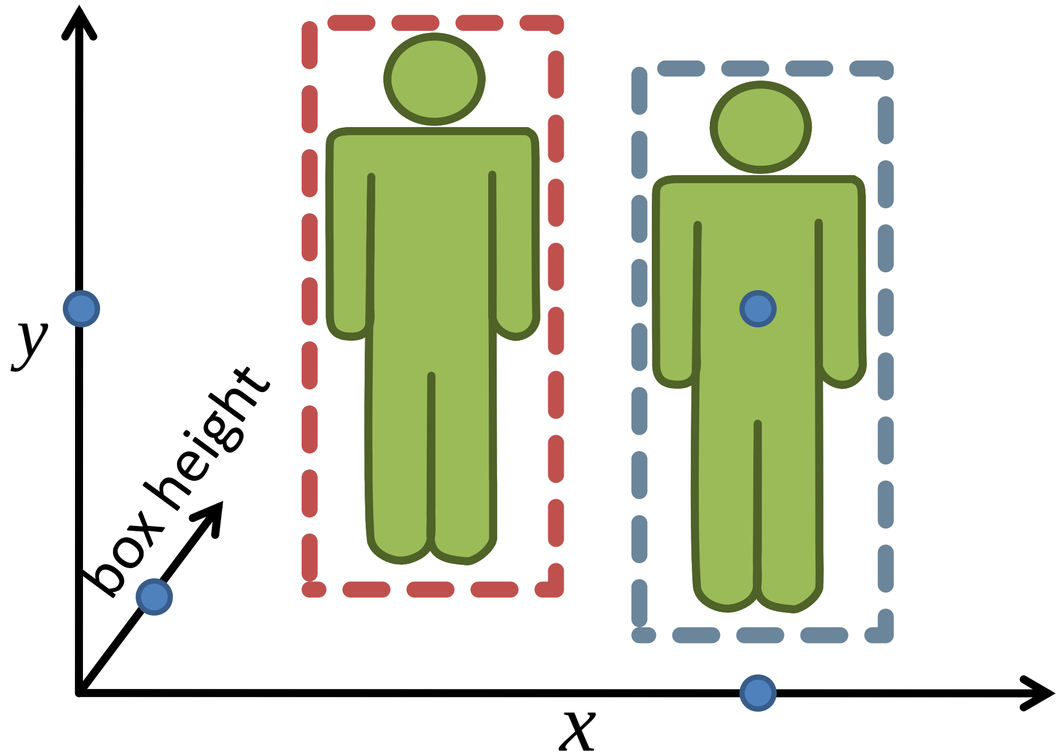
- Example for people tracking in 2D images

$$\mathbf{x}_t = x$$

$$\mathbf{x}_t = (x, y)$$

$$\mathbf{x}_t = (x, y, h)$$

$$\mathbf{x}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2\}$$



State: example

- Object defined by a point in an image
  - position
  - velocity
  - acceleration

$$\mathbf{x}_t = (x, y)$$

$$\mathbf{x}_t = (x, y, \dot{x}, \dot{y})$$

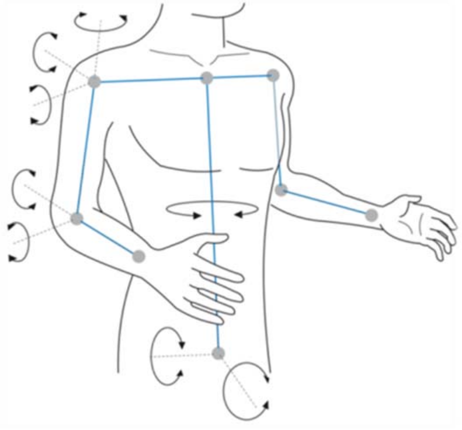
$$\mathbf{x}_t = (x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y})$$

Saad Ali and Mubarak Shah, [Floor Fields for Tracking in High Density Crowd Scenes](), European Conference on Computer Vision (ECCV), 2008.

State: examples

- Camera/object pose (rotation, translation)

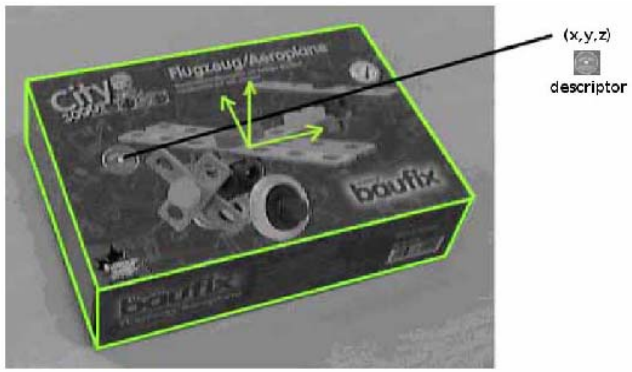
- Joint angles



Measurements

- Sensor measurements provide noisy (indirect) information about the state of the dynamical system under consideration
- Notation:
  - z_t denotes a measurement at time t
  - $z_{t_1:t_2}$ denotes the set of all measurements acquired from time t_1 to t_2

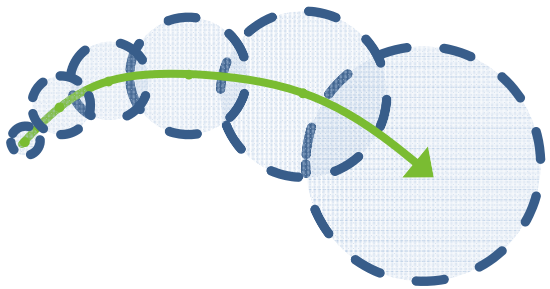
- Typical examples of sensor measurements:
  - Camera images (pixel-/feature-/object-level)
  - Inertial measurements
  - GPS coordinates



Control inputs

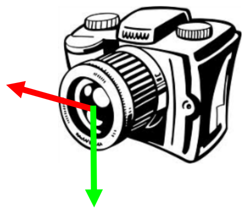
- Control inputs carry noisy information about the change of the dynamic system under consideration
- Notation:
  - u_t denotes control data at time t
  - u_t corresponds to the change of the state in time interval $(t-1; t]$
  - $u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, \dots, u_{t_2}$ denotes sequences of control data

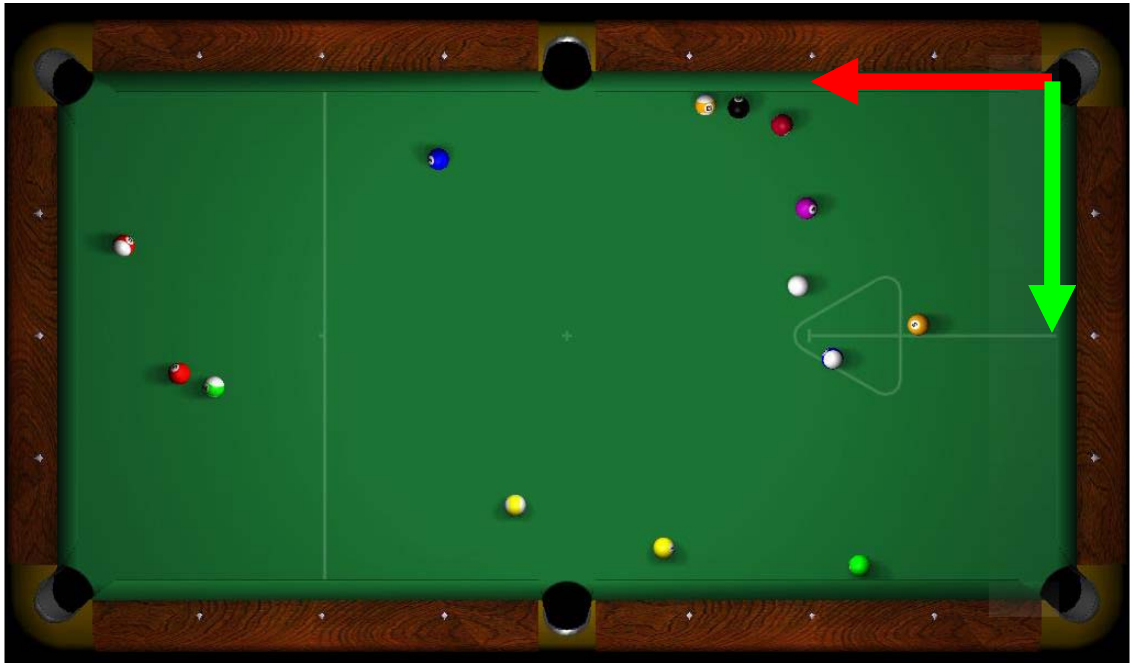
- Typical examples of control inputs:
  - Velocity: setting the velocity of a robot to 10 cm/s for the duration of 5 seconds suggests that the robot is 50 cm ahead of its pose before
  - Odometry: odometers measure the revolution of wheels
  - **No input** (often the case in visual tracking)



State estimation example

- **Dynamical system:** tracking of billard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite:** camera pose known with respect to table

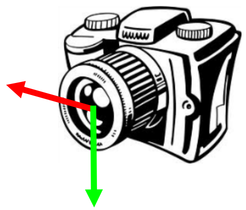


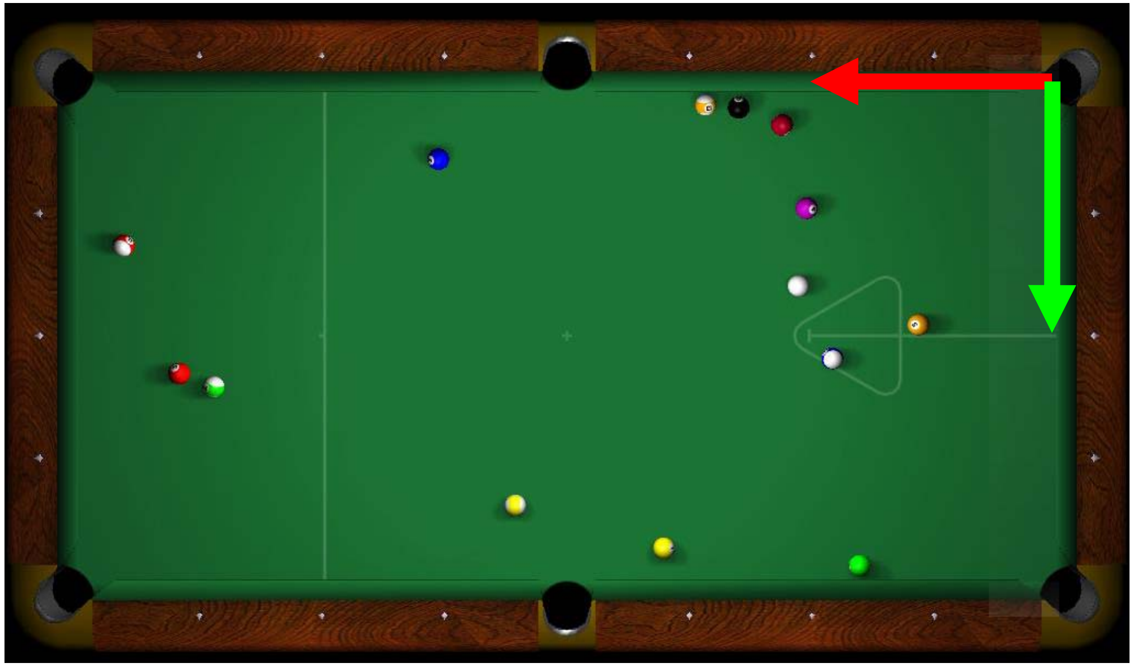


Courtesy of U. Frese

State estimation example

- **Dynamical system:** tracking of billard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite:** camera pose known with respect to table
- Which components are contained in:
  - State x_t:
  - Measurement z_t:
  - Control input u_t:





Courtesy of U. Frese

State estimation example

- **Dynamical system:** tracking of billard balls by means of a camera looking from above (without spin, collision, etc.)
- **Pre-requisite:** camera pose known with respect to table
- Which components are contained in (simple model):
  - State $x_t = (p_x, p_y, \dot{p}_x, \dot{p}_y)$ [m]
    ➔ position and velocity in reference frame of billard table
  - Measurement $z_t = (i_x, i_y)$ [Pixel]
    ➔ pixel position of ball in camera image
  - Control input $u_t = ()$
    ➔ empty, however, we can assume constant velocity during a time interval
- Question: how could the Markov assumption be violated here?
  - E.g. badly calibrated camera
  - Interaction with other balls or table (collisions)
  - Physical aspects: spin, friction, …

Probabilistic generative laws

- The evolution of state and measurements is governed by probabilistic laws
- State x_t is generated stochastically from state x_{t-1}:

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

- Assuming that the state is complete:

$$\boxed{p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t)}$$

Markov assumption:

example of conditional independence

Probabilistic generative laws

- Measurement z_t is generated stochastically from state x_t:

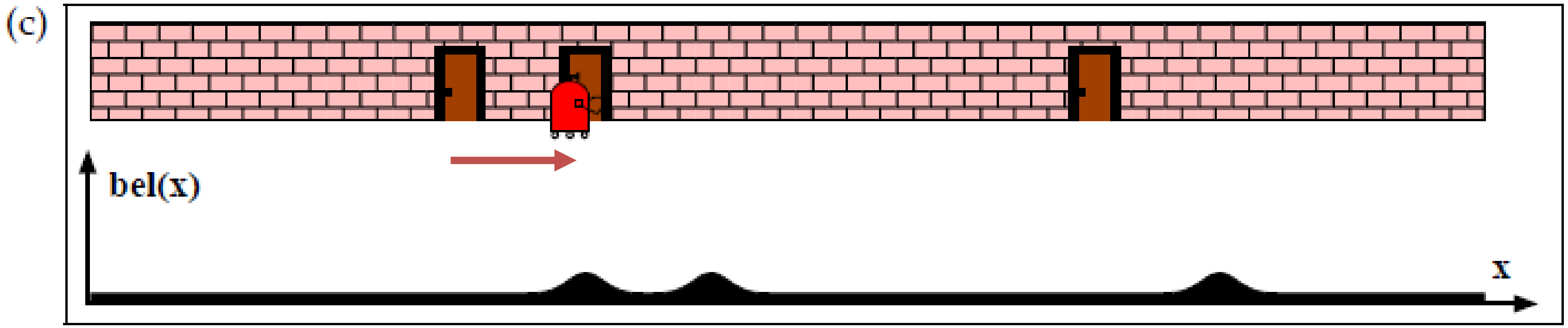
$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t})$$

- Assuming that the state is complete:

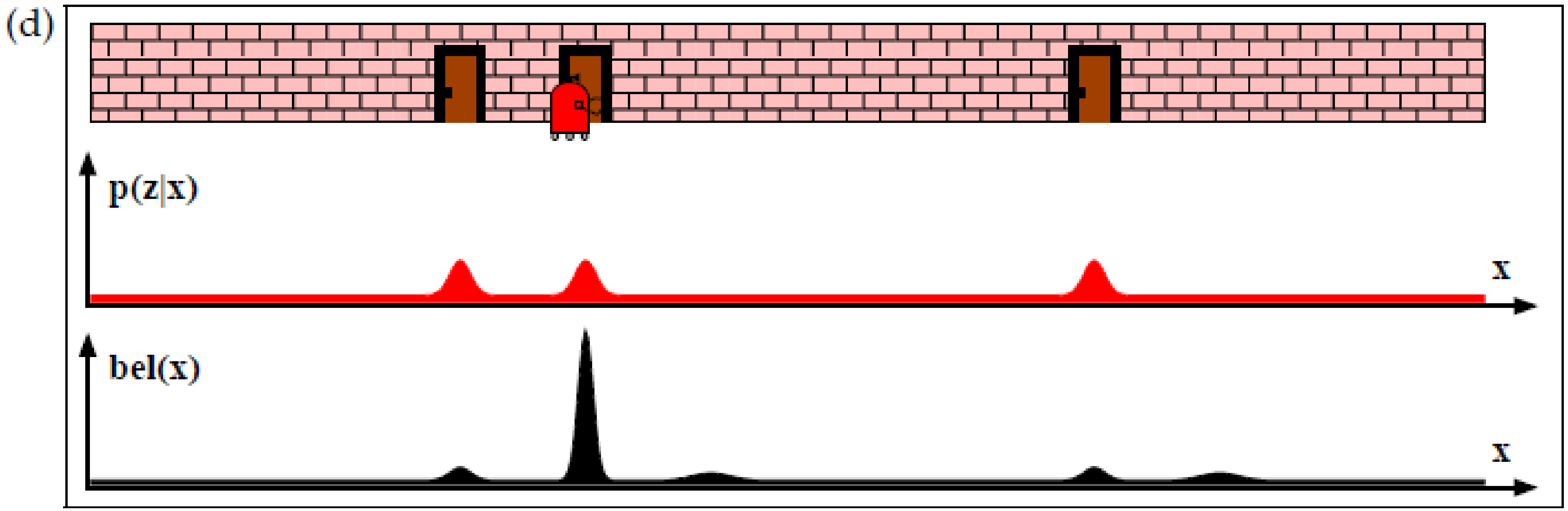
$$\boxed{p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)}$$

Another Markov assumption
(conditional independence)

Introductory example



Odometry input u_1: 1m forward.



Measurement z_2: here is a door.

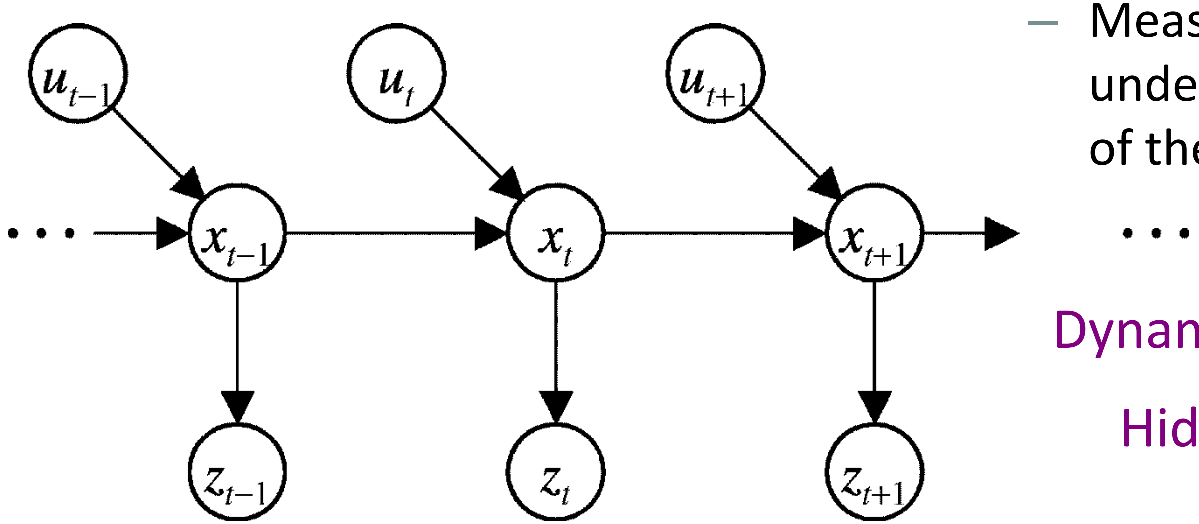
Probabilistic generative laws

Motion model

$p(x_t|x_{t-1}, u_t)$ This is what we model! $p(z_t|x_t)$

Measurement model

- State transition probability
  - Specifies, how the state evolves over time as a function of the previous state and the current control data
- Measurement probability/likelihood
  - Specifies how measurements are generated as function of the state
  - Measurements can be understood as noisy projections of the state



Dynamic Bayesian network/

Hidden Markov model

Belief

- In **Bayesian inference**, we usually want to estimate the state x_t given sequences of measurements $z_{1:t}$ and control data $u_{1:t}$ and the respective state transition $p(x_t|x_{t-1}, u_t)$ and measurement probabilities $p(z_t|x_t)$
- Our estimate of the true state x_t is also called belief:

Measurement update/correction: calculation of posterior from predicted state

$$\boxed{bel(x_t) = p(x_t|z_{1:t}, u_{1:t})}$$

Posterior distribution of x_t conditioned on all available data
➔ after including the current measurement z_t

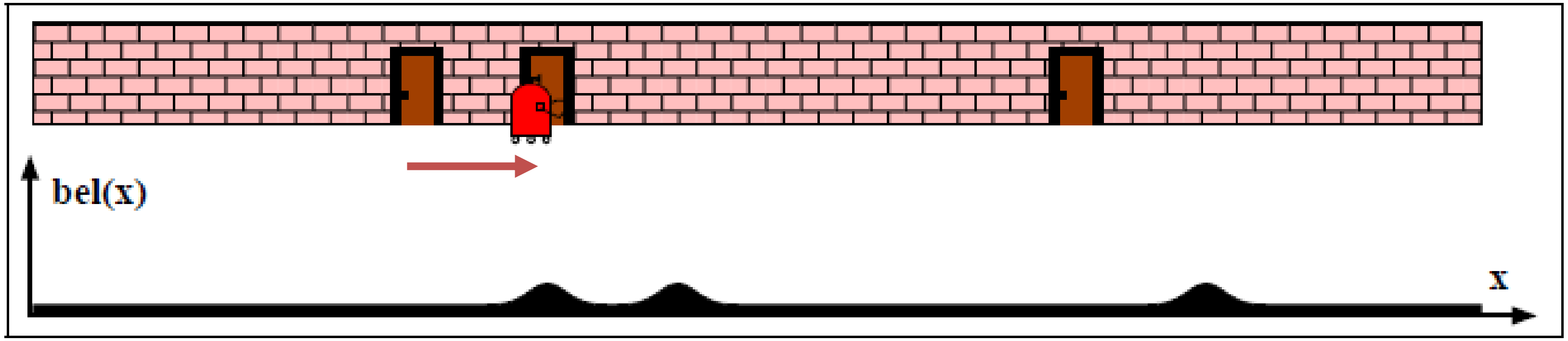
Time update: calculation of predicted state from current state and control input

$$\boxed{\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})}$$

Prediction of x_t before including the current measurement z_t

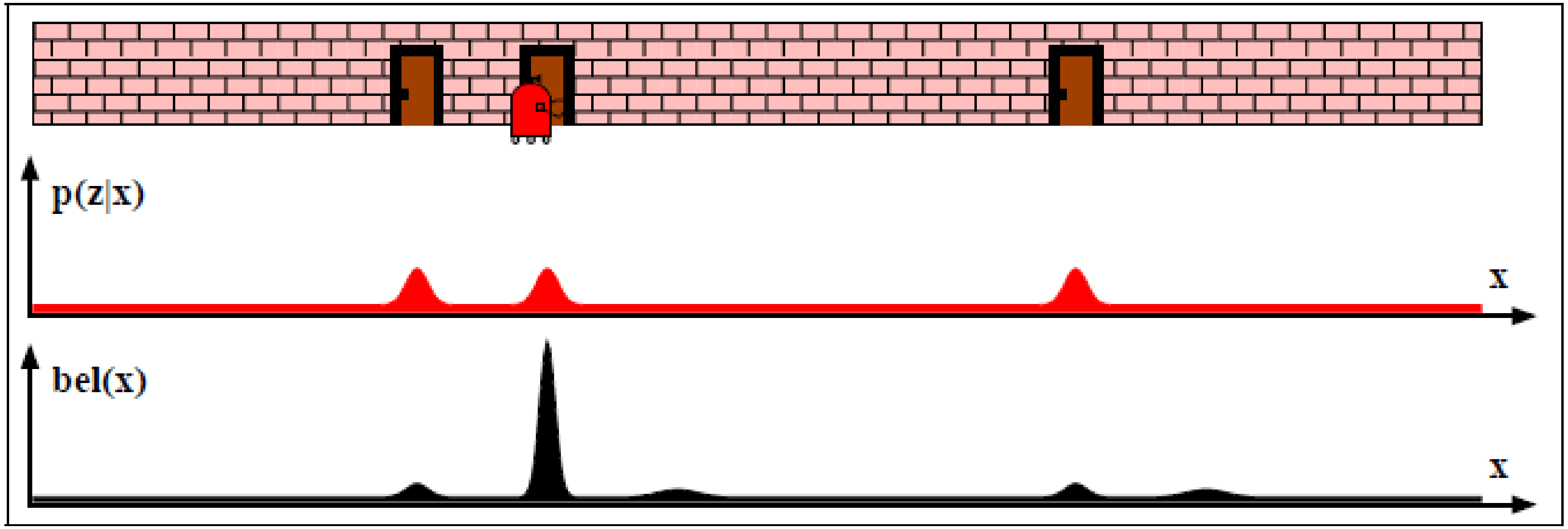
Introductory example

(c)



Odometry input u_1: 1m forward.

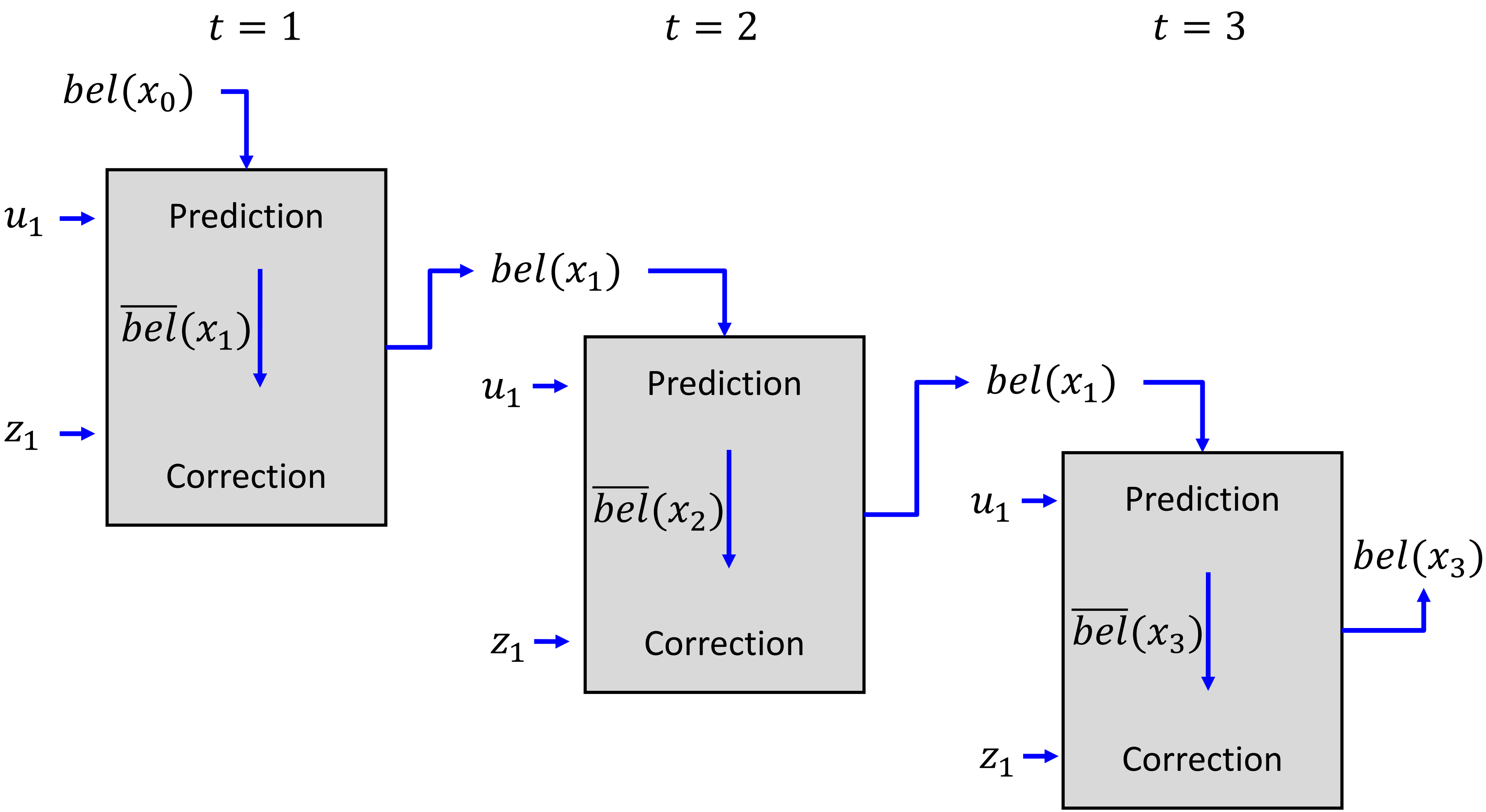
(d)



Measurement z_2: here is a door.

A general algorithm for state estimation (inference): Bayes filter

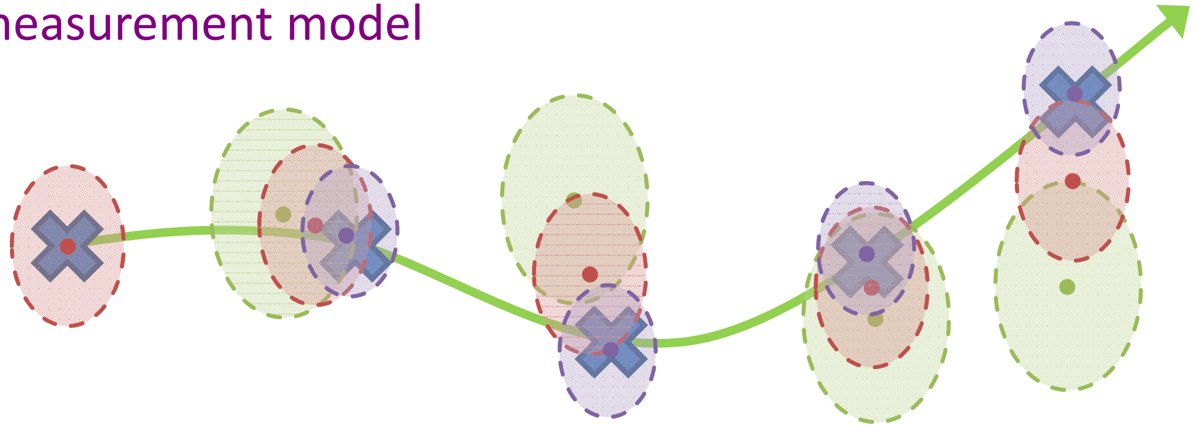
Recursive Bayes filter algorithm



All entities are modelled as random variables with PDFs

Recursive Bayesian filtering

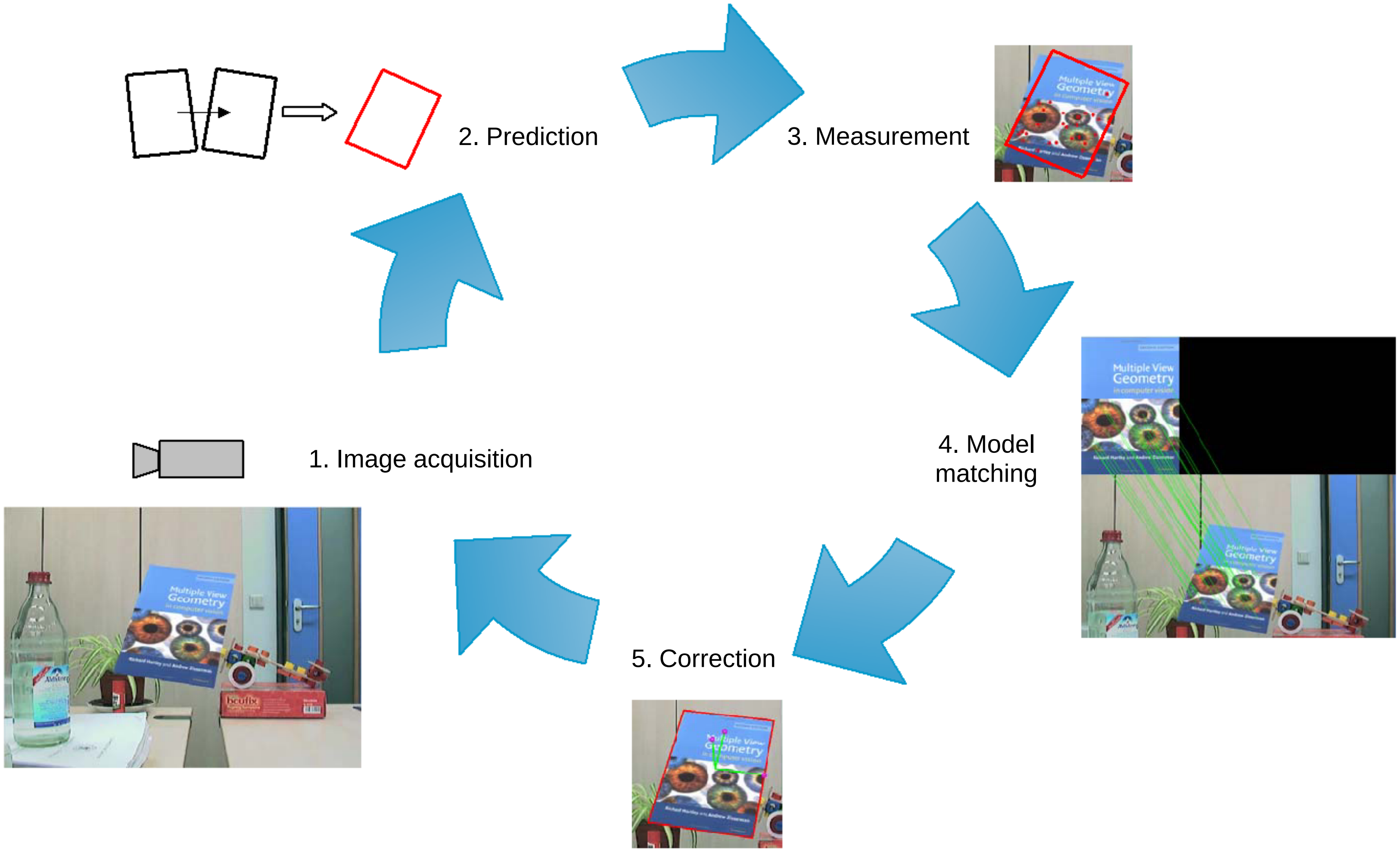
- Use probability distributions to model the estimation problem
  - Prediction/time update: calculate prior belief based on dynamic model
  - Correction/measurement update: calculate posterior belief based on measurement model



correct measure predict

$$\underbrace{bel(x_t)}_{\text{posterior}} = \eta\, \underbrace{p(z_t|x_t)}_{\text{likelihood}} \int \underbrace{p(x_t|x_{t-1},u_t)}_{\text{motion model}}\, \underbrace{bel(x_{t-1})}_{\text{posterior at } t\text{-}1} dx_{t-1}$$

Tracking pipeline



Recursive Bayes filter algorithm

Bayes_filter($bel(x_{t-1}), u_t, z_t$):

for all x do

$\overline{bel}(x_t)$ = Time_update($bel(x_{t-1}), u_t$)

$bel(x_t)$ = Measurement_update($\overline{bel}(x_t), z_t$)

endfor

return $bel(x_t)$

Measurement update step derived

$bel(x_t)$ `= Measurement_update(` $\overline{bel}(x_t)$, z_t `)`

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

Bayes

$$= \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t})\, p(x_t | z_{1:t-1}, u_{1:t})}{p(z_t | z_{1:t-1}, u_{1:t})}$$

Markov,

normalizer

$$= \eta\, p(z_t | x_t)\, p(x_t | z_{1:t-1}, u_{1:t})$$

$$= \eta\, p(z_t | x_t)\, \overline{bel}(x_t)$$

Time update step derived

$\overline{bel}(x_t)$ = `Time_update(` $bel(x_{t-1}),\ u_t$ `)`

Expand using marginalization

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$$

$$= \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})\, p(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1}$$

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

Markov

$$= \int p(x_t|x_{t-1}, u_t)\, bel(x_{t-1}) dx_{t-1}$$

For a finite state space, the integral turns into a sum

Bayes update rule

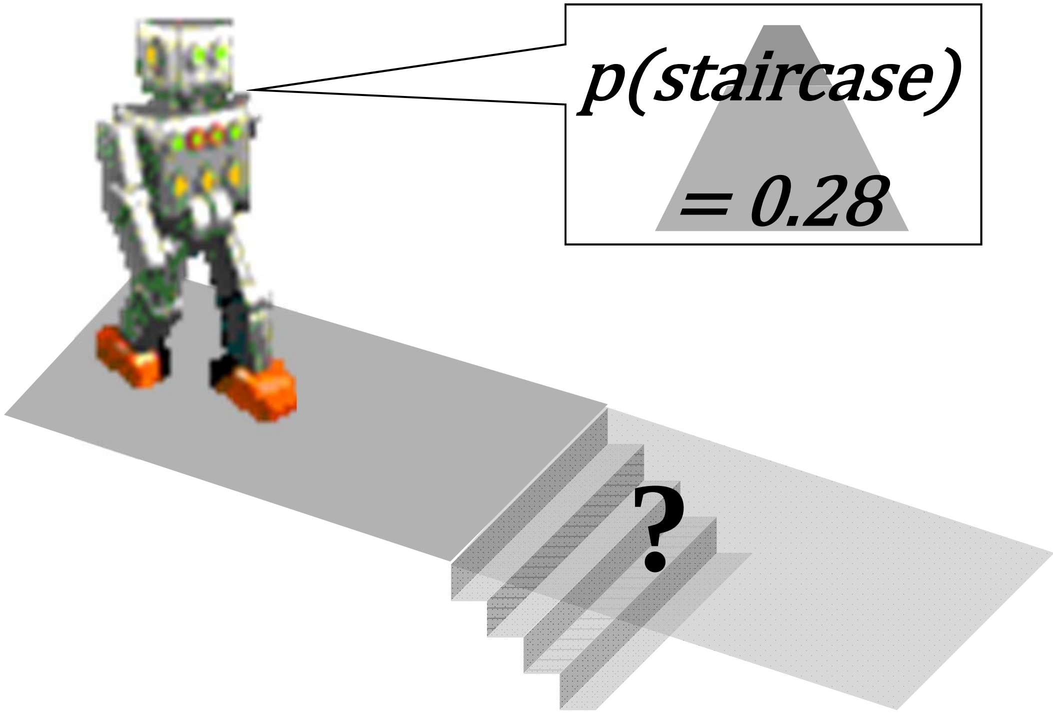
$$\underbrace{bel(x_t)}_{\substack{\text{Posterior} \\ \text{at time } t}} = \eta \underbrace{p(z_t|x_t)}_{\substack{\text{Measurement likelihood} \\ \text{Measurement model}}} \int \overbrace{\underbrace{p(x_t|x_{t-1}, u_t)}_{\substack{\text{State transition probability} \\ \text{Dynamic model}}} \underbrace{bel(x_{t-1})}_{\substack{\text{Posterior at} \\ \text{time } t-1}} dx_{t-1}}^{\overline{bel}(x_t)}$$

Bayes filter algorithm

Prerequisites:

- Assumption: the world is Markov, i.e. the state is complete
- Given: 3 probability density functions:
  - Initial belief: $p(x_0)$
  - Measurement probability: $p(z_t|x_t)$
  - State transition probability: $p(x_t|x_{t-1}, u_t)$

Hands-on example of Bayesian inference



Sensor model

$p(image \mid staircase) = 0.7$

$p(image \mid no\ staircase) = 0.2$

Prior belief

$p(staircase) = 0.1$

```
for all x do
    bel(x_t) = η p(z_t|x_t) bel‾(x_t)
endfor
```

$$\text{for all } x \text{ do} \quad bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t) \quad \text{endfor}$$

Bayesian inference (measurement update)

$$p(staircase \mid image) = \frac{p(image \mid staircase)\, p(staircase)}{p(im \mid stair)\, p(stair) + p(im \mid no\ stair)\, p(no\ stair)}$$

$$= 0.7 \bullet 0.1 / (0.7 \bullet 0.1 + 0.2 \bullet 0.9) = 0.28$$

Tip: how to calculate the normalization

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

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y \mid x) P(x)}$$

Algorithm:

$$\forall x : \text{aux}_{x|y} = P(y \mid x)\ P(x)$$

$$\eta = \frac{1}{\sum_x \text{aux}_{x|y}}$$

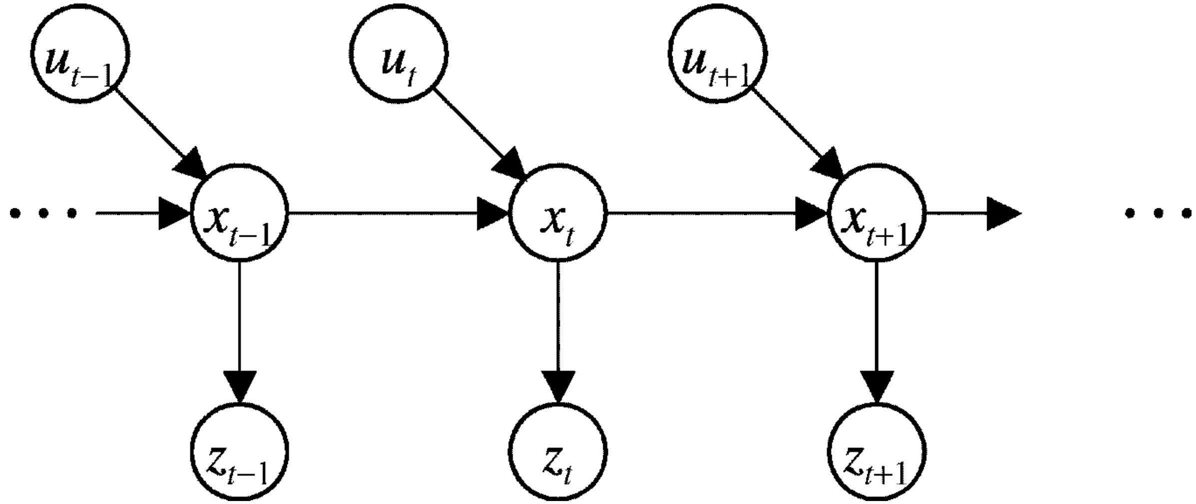
$$\forall x : P(x \mid y) = \eta\ \text{aux}_{x|y}$$

The resulting distribution must integrate to 1

Summary: Bayes filter framework

- **Given:**
  - Stream of measurements $z_{1:t}$ and control data $u_{1:t}$
  - Measurement model $p(z_t|x_t)$
  - Dynamic model $p(x_t|x_{t-1}, u_t)$
  - Prior/Initial probability of the system state $p(x_0)$
- **Wanted:**
  - Estimate of the state x_t of a dynamical system
  - The posterior of the state is also called belief: $\mathrm{bel}(x_t) = p(x_t|u_{1:t}, z_{1:\mathrm{t}})$

Markov Assumption

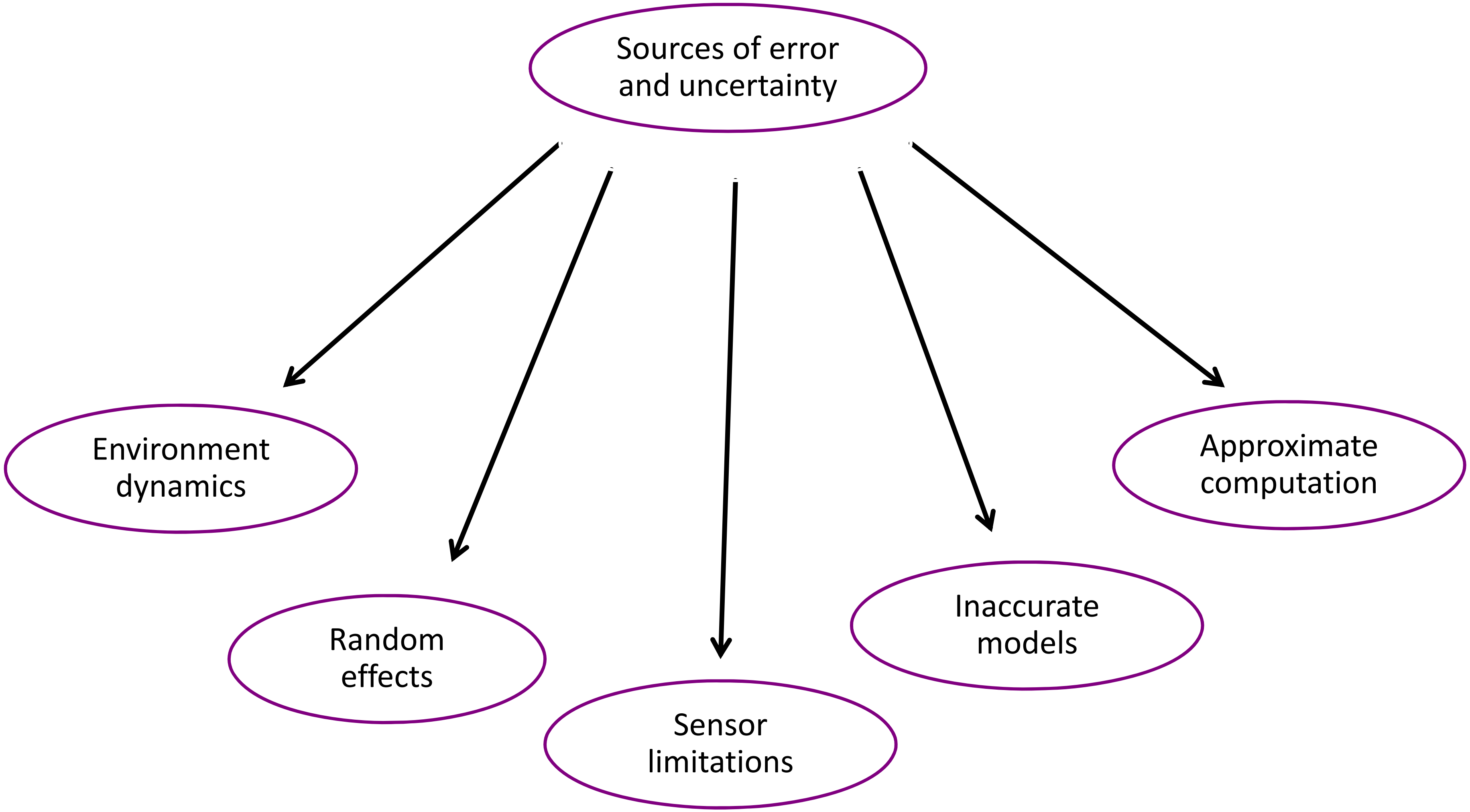


$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t)$$
$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$$

Underlying Assumptions

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

Reality



Sources of error and uncertainty
Environment dynamics
Random effects
Sensor limitations
Inaccurate models
Approximate computation

Summary: Bayes filters

- Probabilistic tool for **recursively** estimating the **state** of a **dynamical system** from **noisy measurements** and **control inputs**.
- Based on probabilistic concepts such as the **Bayes theorem**, **marginalization**, and **conditional independence**.
- Make a **Markov assumption** according to which the state is a complete summary of the past. In real-world problems, this assumption is usually an approximation!

- Can in the presented form only be implemented for simple estimation problems, requires either…or…
  - closed form solutions for multiplication and integral
  - restriction to finite state spaces

Outlook

- What is missing:
  - Concrete representations for belief
  - Concrete representations for probability density functions
  - Implementable and tractable filter approximations
  - Applicability to complex and continuous estimation problems
  - Hands-on experience

- Readings:
  - Kalman Filtering book by Peter Maybeck, chapter 1: http://www.cs.unc.edu/~welch/kalman/maybeck.html
- Next lectures:
  - Filters: (Extended) Kalman filter
  - Measurement and motion models

Exercise 1

- Available at: http://av.dfki.de/images/stories/lectures/opt-ss12/exercise1.pdf
  - Simple computations (probabilistic concepts, Bayes filter)
  - Handling of Gaussians (preparation for next lecture)
- If you want feedback, hand in solutions until June 12
- Tutorial session: Thursday, 14.06.2012, 14:00-15:30, DFKI, room 2.04 (second floor)
  - Discussion of solutions
  - Preparation for next lecture (Bayes filter with Gaussians)

- Any questions?