Computer Vision I - Tracking

Carsten Rother

11/12/2015

[slide credits: Alex Krull]

What is Tracking?

- „Tracking an object in an image sequence means continuously identifying its location when either the object or the camera are moving“ [Lepetit and Fua 2005]
- This can mean estimating in each frame:
  - 2D location
    - **Autostereoscopic Display [1] : eye detection (0.1s) / tracking (40ms)**

CeBIT Highlight 1999

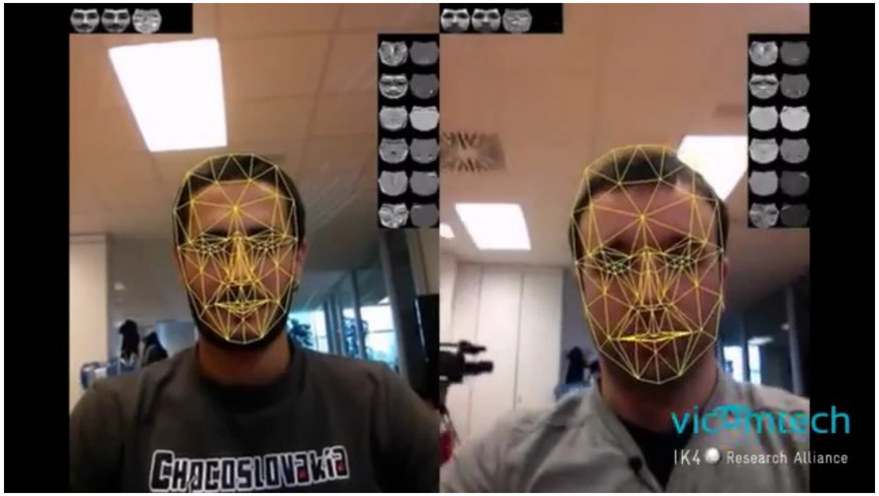
1st prize of innovation of Saxony 2001

[1] H. Heidrich, et al. “Eye position detection system,” *Stereoscop. Displ. Virt. Real. Syst.*, 2000.

What is Tracking?

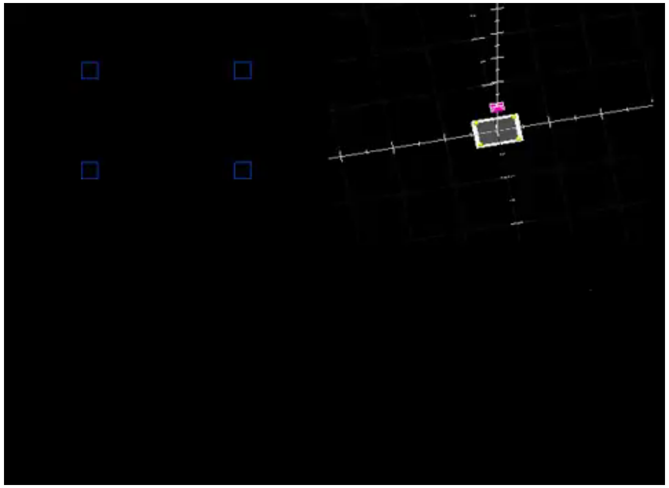
- „Tracking an object in an image sequence means continuosly identifying its location when either the object or the camera are moving“ [Lepetit and Fua 2005]

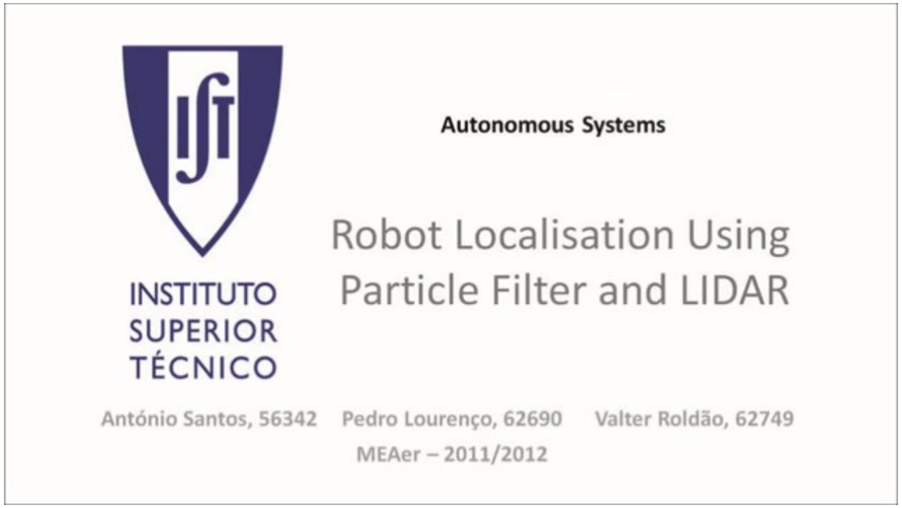
- This can mean estimating in each frame:
  - 2D location or window
  - 6D rigid body transformation
  - More complex parametric models
    - Active Appearance Models
    - Skeleton for human pose etc.



Tracking vs Localization

- Tracking of objects is closely related to:
  - camera pose estimation in a known environment
  - localization of agents (eg. Robots) in a known environment
  - Reminder: SLAM has unknown location (agent, camera) and unknown environment



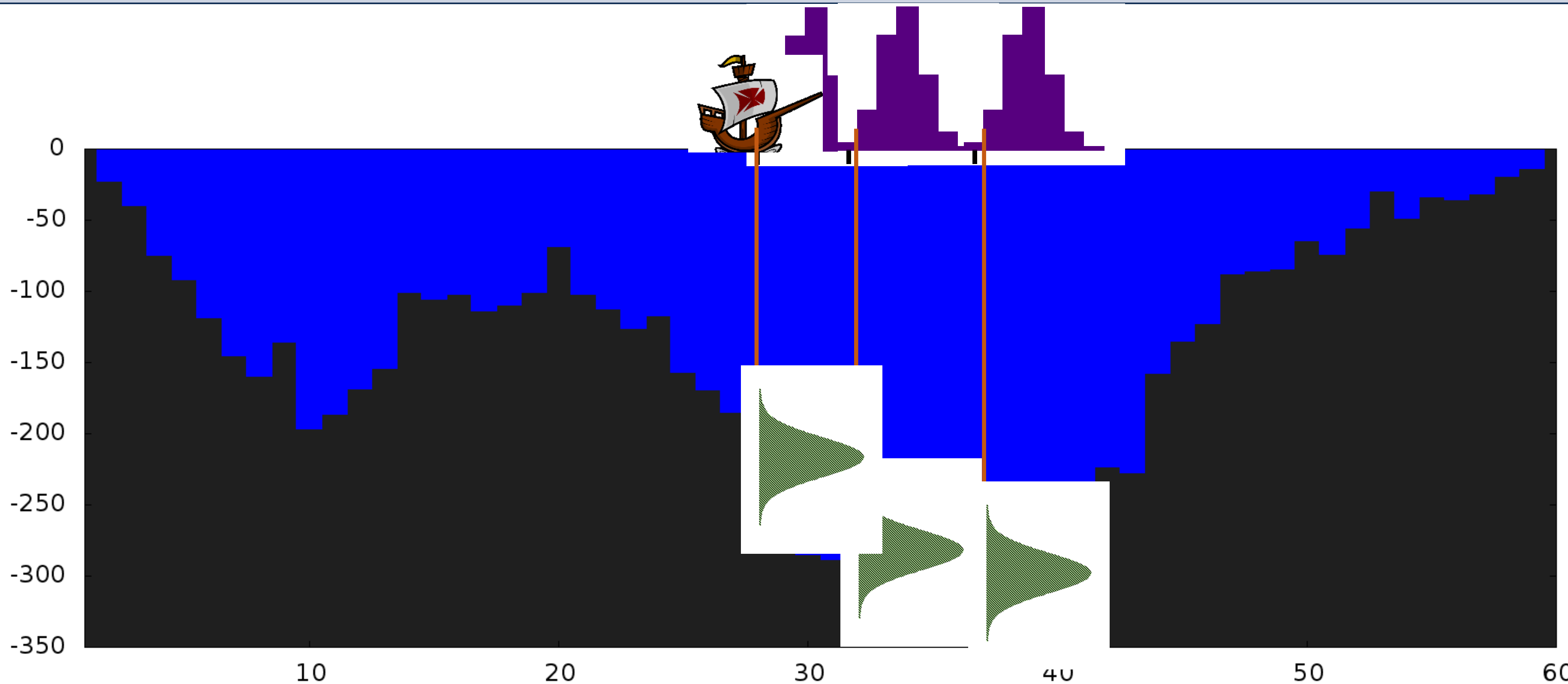


Outline

- This lecture
  - The Bayes Filter
    - Explained for localization

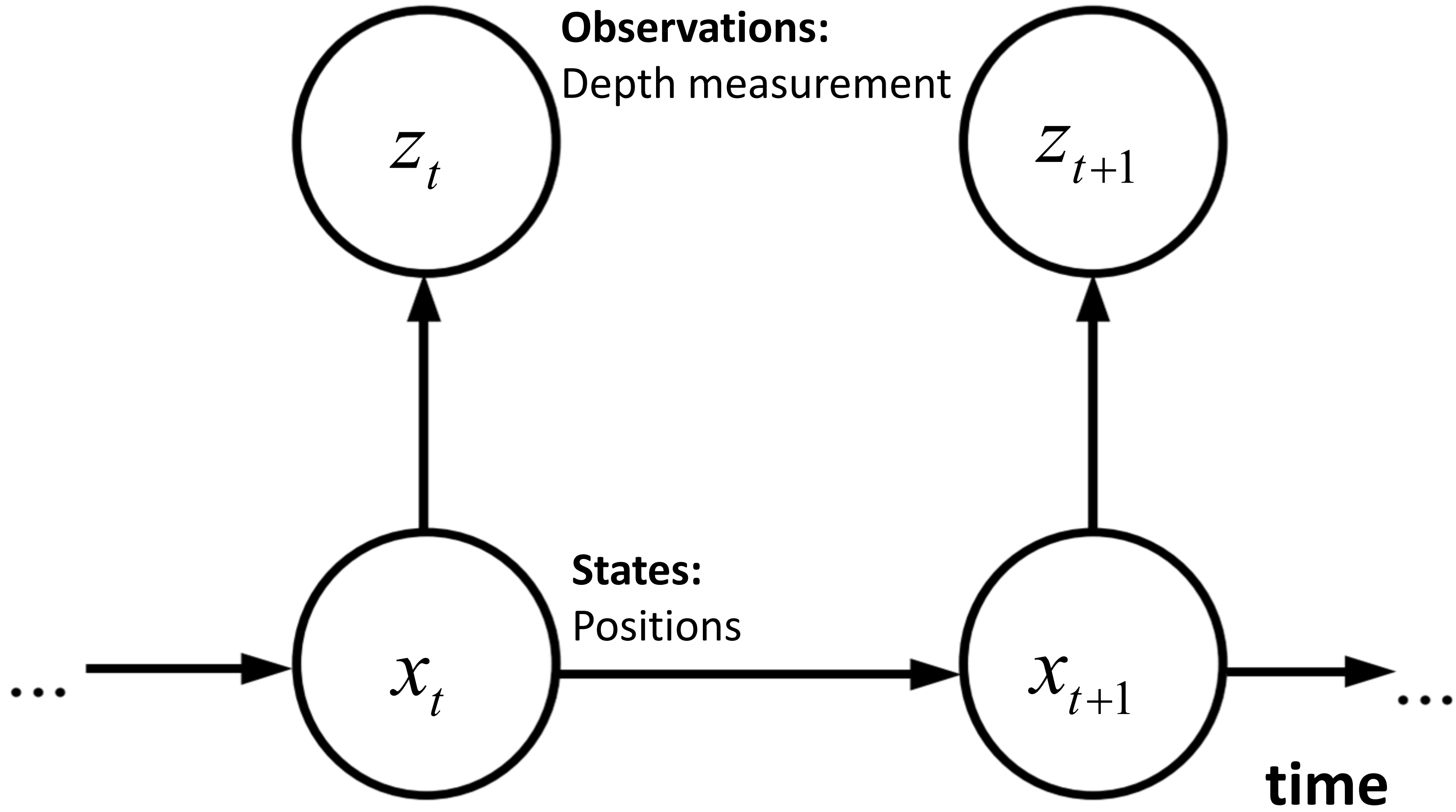
- Next lecture
  - The Particle Filter
  - The Kalman Filter
  - Pros and Cons
  - Pose Estimation & Pose Tracking [Eric]:
    - 6-DOF Model Based Tracking via Object Coordinate Regression

The Bayes Filter / Convolute and Multiply

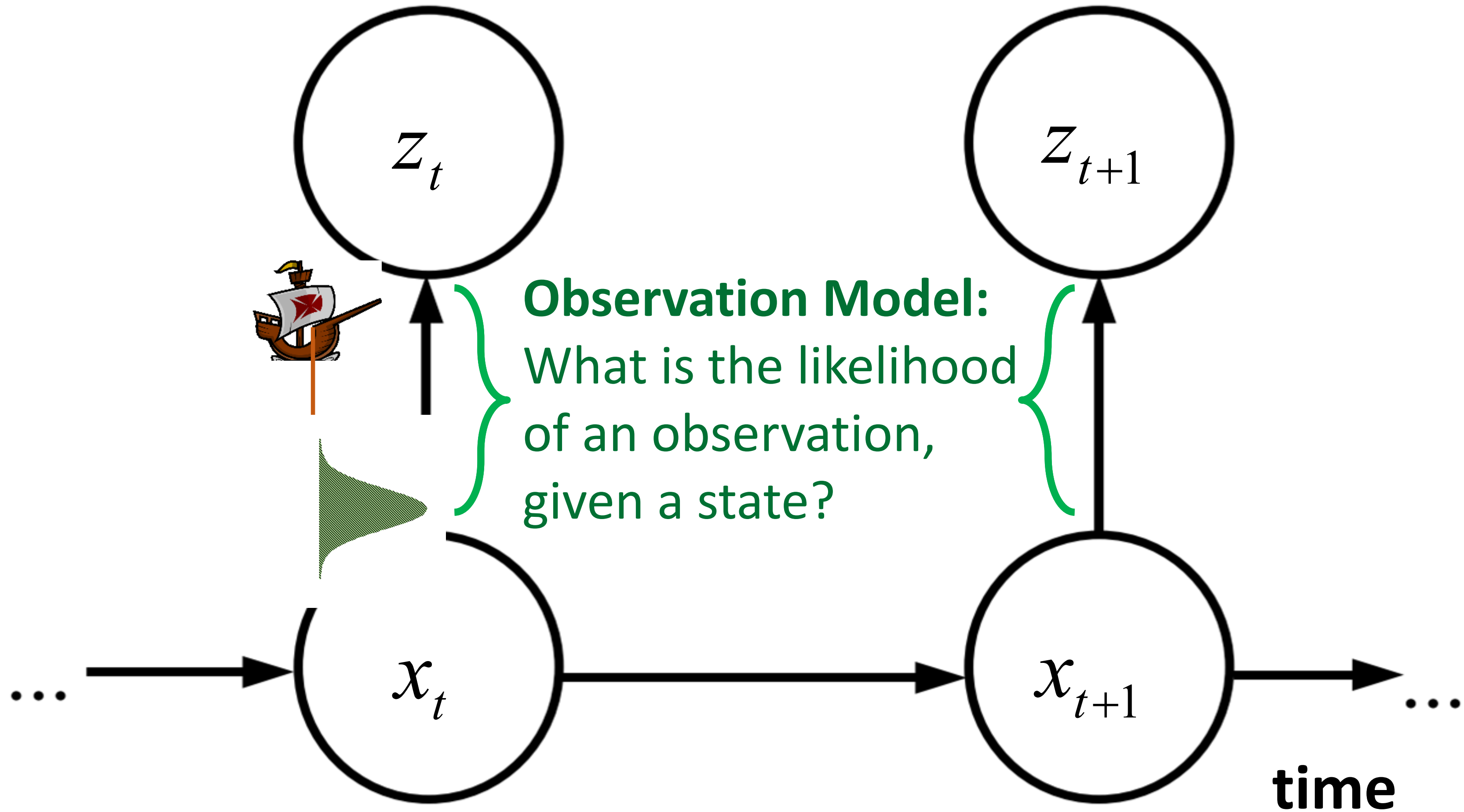


- We have:
  - Probabilistic model for movement
  - Probabilistic model for measurement
    - Based on map of the environment
- Where is the ship?
  - Using all previous and current observations

The Hidden Markov Model



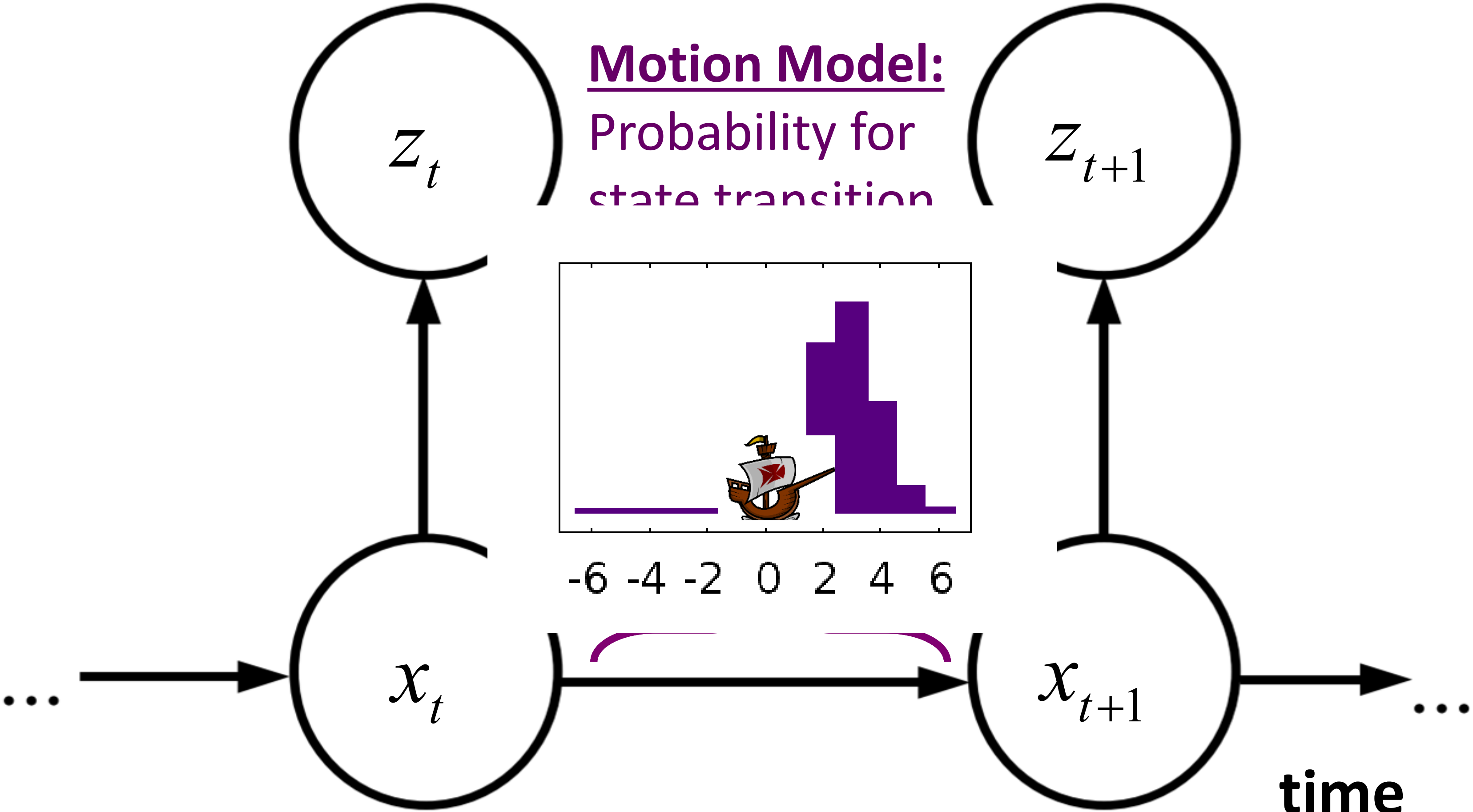
The Hidden Markov Model



$$p(z_t|x_t) = \frac{1}{c} e^{-\left(z_t - d(x_t)\right)^2/(2\sigma^2)}$$

$d(x_t)$ is the known true depth

The Hidden Markov Model



$$p(x_{t+1}|x_t)$$

Probabilities - Reminder

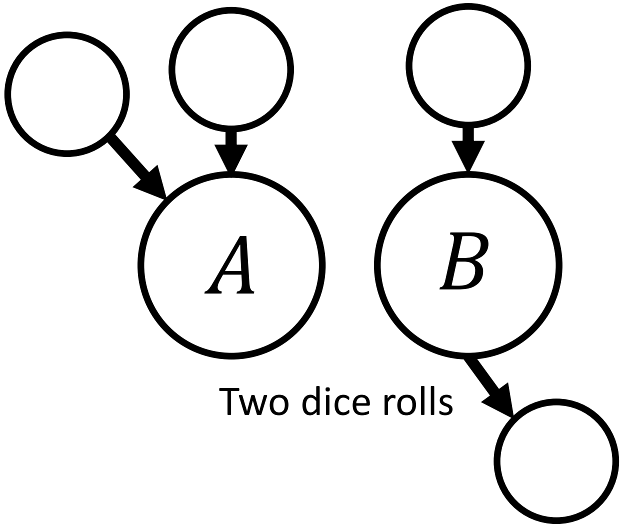
- A random variable is denoted with $x \in \{0, \dots, K\}$
- Discrete probability distribution: $p(x)$ satisfies $\sum_x p(x) = 1$
- Joint distribution of two random variables: $p(x, z)$
- Conditional distribution: $p(x|z)$
- Sum rule (marginal distribution): $p(z) = \sum_x p(x, z)$
- Independent probability distribution: $p(x, z) = p(z)p(x)$
- Product rule: $p(x, z) = p(z|x)p(x)$
- Bayes' rule: $p(x|z) = \frac{p(z|x)p(x)}{p(z)}$

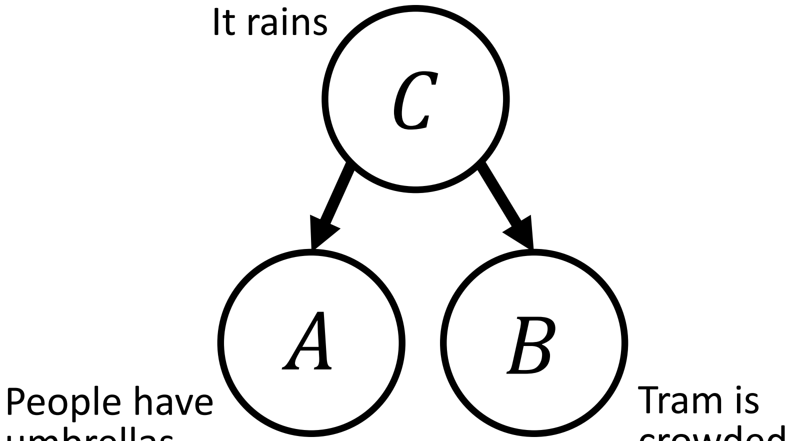
Probabilities - Reminder

- Sum rule (marginal distribution): $p(z) = \sum_x p(x, z)$
- Product rule: $p(x, z) = p(z|x)p(x)$
- Bayes' rule: $p(x|z) = \frac{p(z|x)p(x)}{p(z)}$

- Sum rule (marginal distribution): $p(z|A) = \sum_x p(x, z|A)$
- Product rule: $p(x, z|A) = p(z|x, A)p(x|A)$
- Bayes' rule: $p(x|z, A) = \frac{p(z|x,A)p(x|A)}{p(z|A)}$

Probabilities - Reminder





Independence

A and B are not connected in graph

- A does not contain information about B
- $p(A, B) = p(A)p(B)$
- $p(A|B) = p(A)$
- $p(B|A) = p(B)$

Conditional Independence

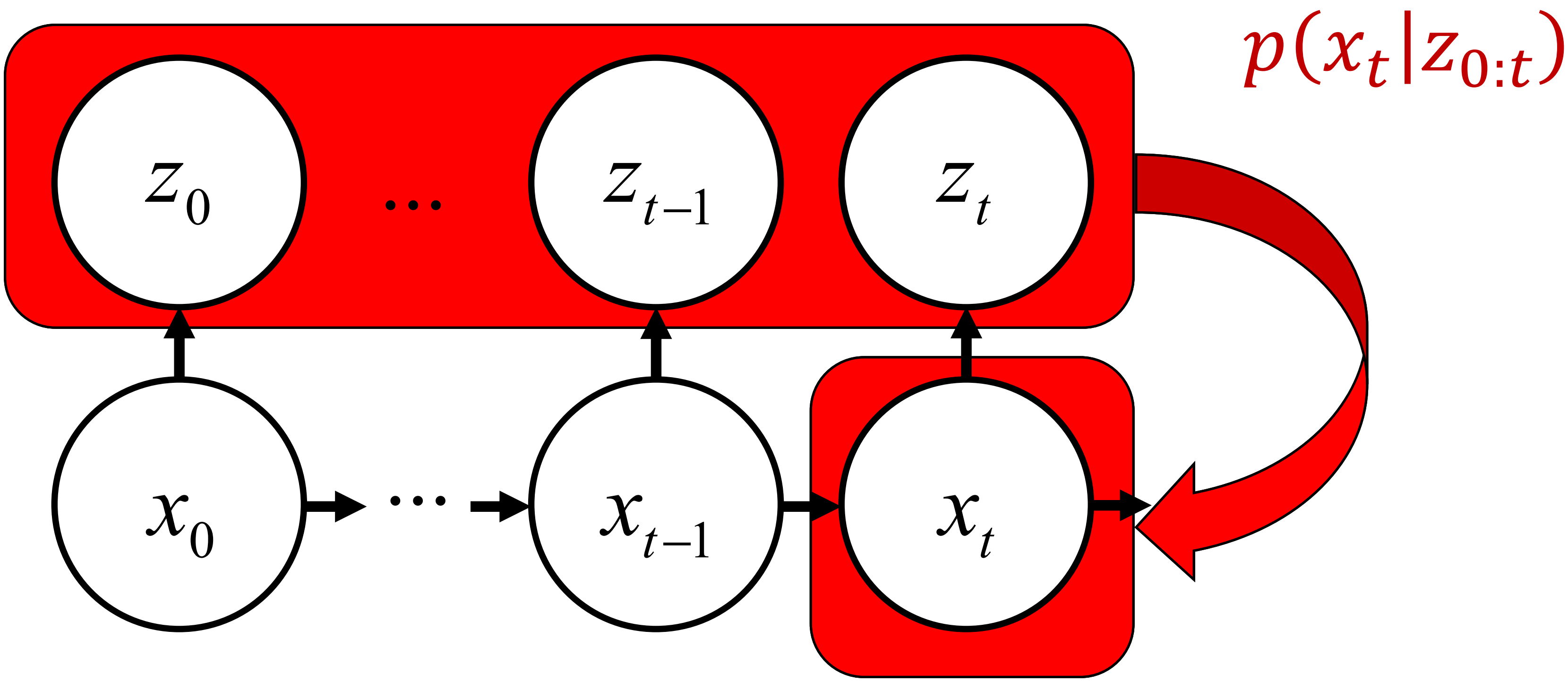
A and B are connected **only via** C

- A does not contain information about B **when** C **is known**
- $p(A, B|C) = p(A|C)p(B|C)$
- $p(A|B, C) = p(A|C)$
- $p(B|A, C) = p(B|C)$

The **Posterior** distribution

Probability distribution for the state given all **previous and current** observations:

- This is what we are interested in
- Eg. use maximum as current estimate $x_t^* = armax_{x_t}\, p(x_t|z_{0:t})$

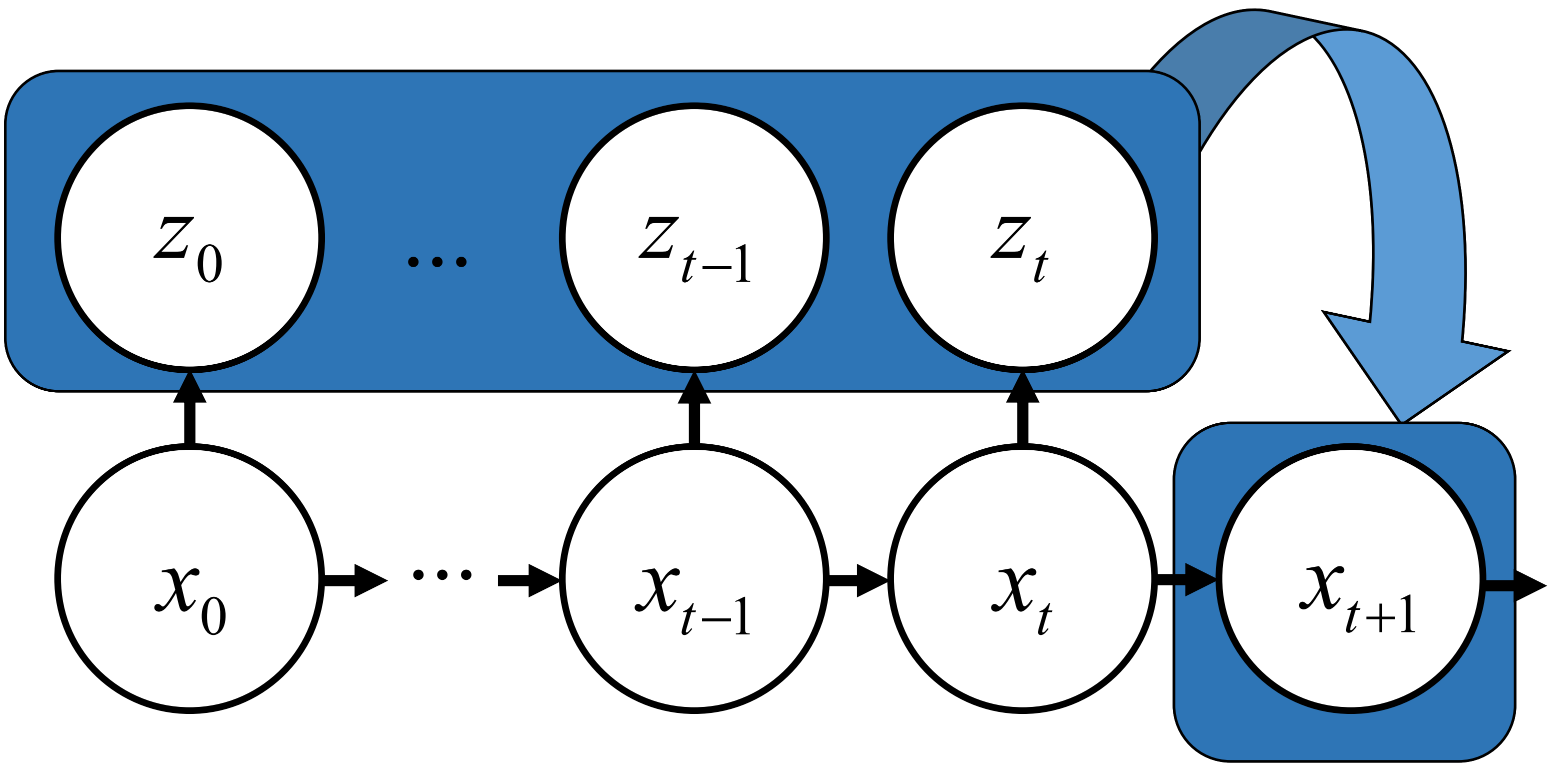


The **Prior** distribution

Probability distribution for the next state given **only previous** observations:

- Intermediate step for calculating the next posterior

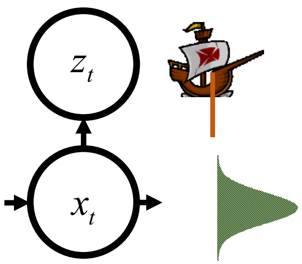
$$p(x_{t+1}|z_{0:t})$$



Important Distributions

Observation Model:

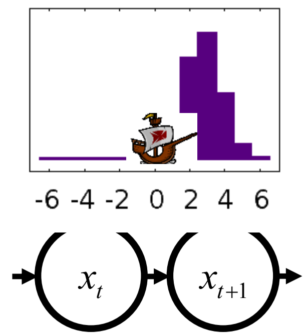
$p(z_t|x_t)$



- Likelihood of observation given state
- Continuous Gaussian around real depth

Motion Model:

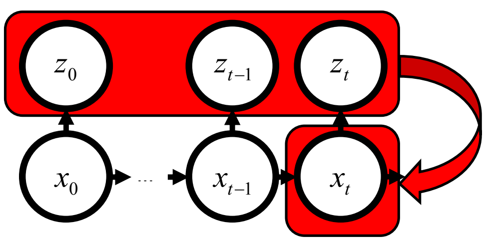
$p(x_{t+1}|x_t)$



- Probability of new state given old one
- Discrete Gaussian

Posterior:

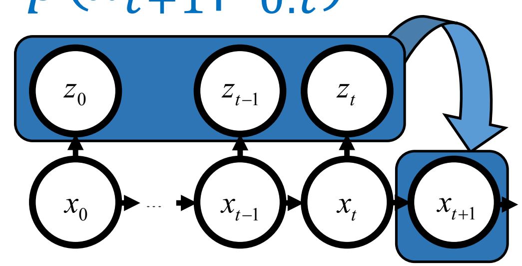
$p(x_t|z_{0:t})$



- Probability of state given previous and current observations

Prior:

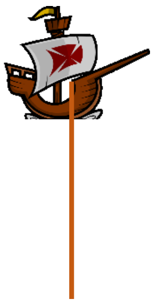
$p(x_{t+1}|z_{0:t})$



- Probability of state given only previous observations

Step by Step

$t = 0$



- Assume **prior** for first frame: $p(x_0)$
- Make first measurement: z_0

Step by Step

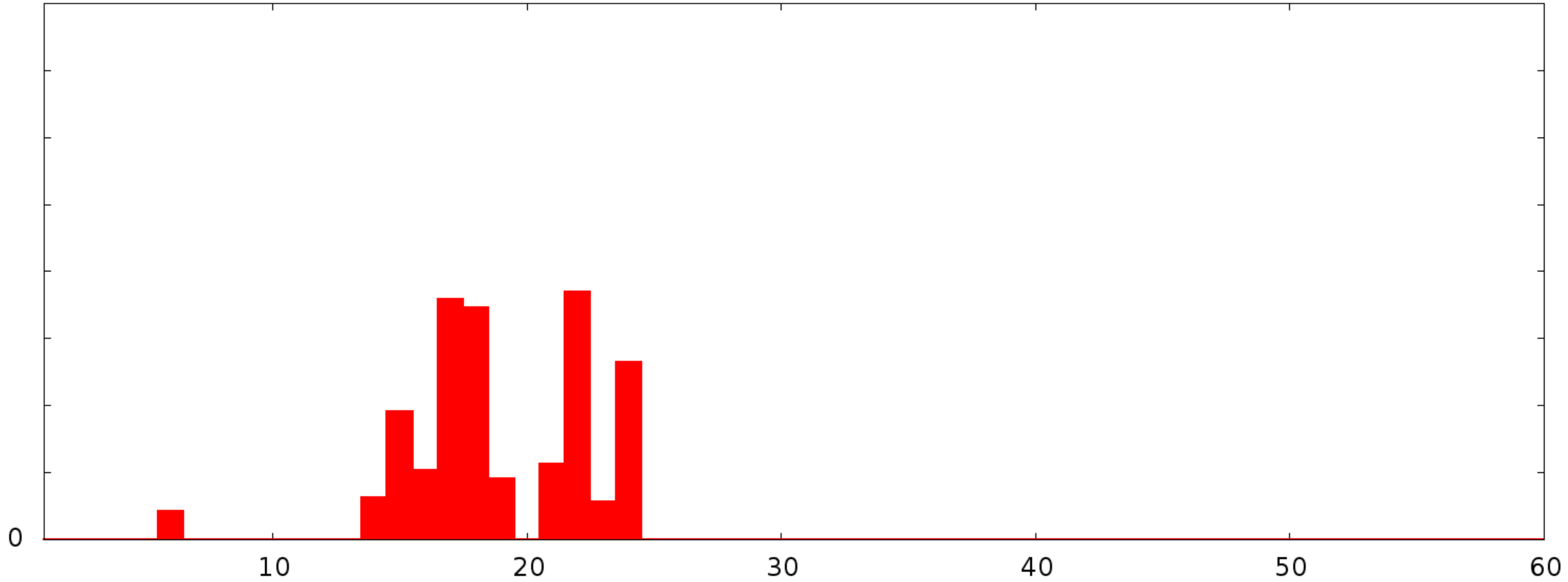
$t = 0$

- Calculate likelihood $p(z_0|x_0)$ for every possible state x_0:

Step by Step

- Calculate the **posterior** by
  - **multiplying** with prior
  - Normalizing
- Reducing uncertainty

$$p(x_0|z_0) = \frac{p(x_0)p(z_0|x_0)}{\sum_{\hat{x}} p(x_0 = \hat{x})p(z_0|x_0 = \hat{x})}$$

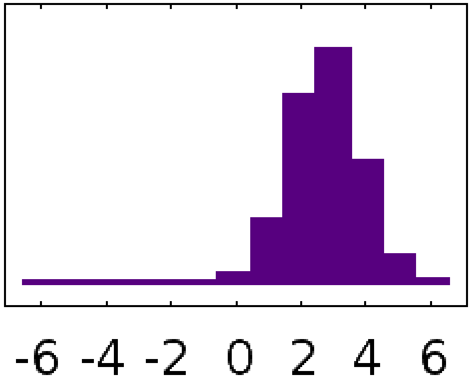


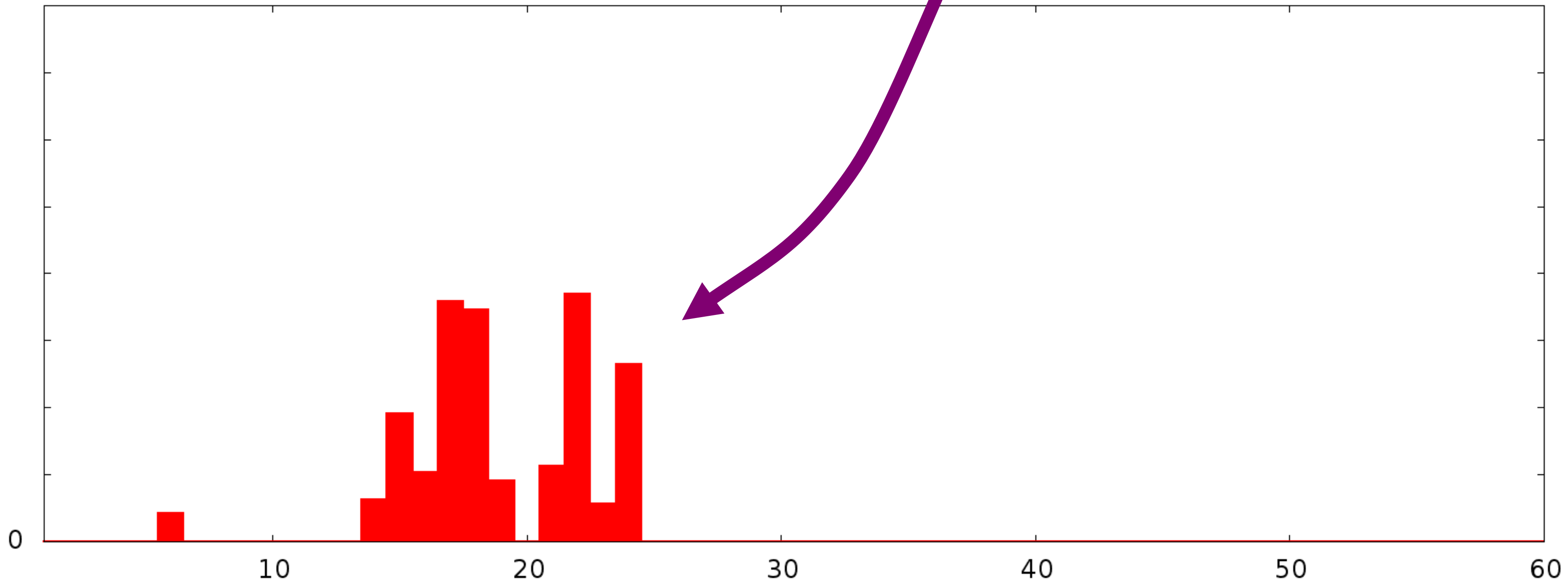
The Bayes Filter / Convolute and Multiply

$t = 0$

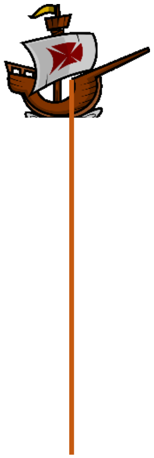
- Calculate the **prior** by **Convolution** with motion model
- Adding uncertainty

$$p(x_1|z_0) = \sum_{\hat{x}} p(x_0 = \hat{x}|z_0)\, p(x_1|x_0 = \hat{x})$$

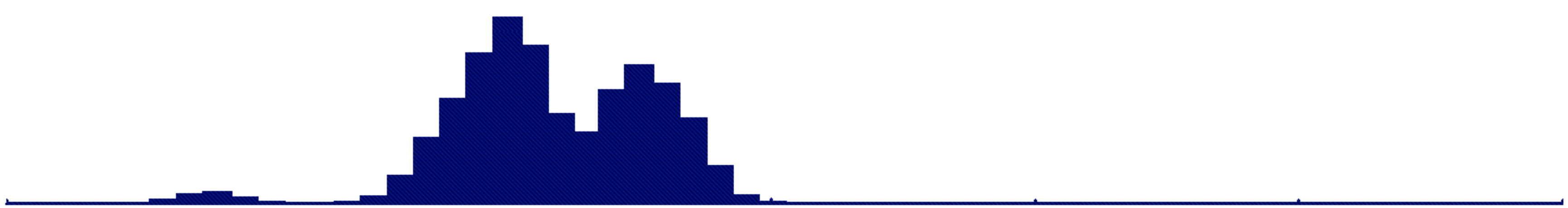




The Bayes Filter / Convolute and Multiply $t = 1$



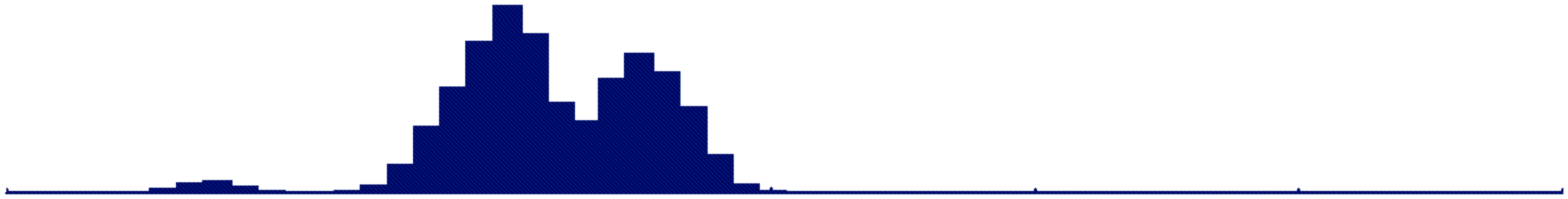
- Make new measurement: z_1



The Bayes Filter / Convolute and Multiply

$t = 1$

- Calculate likelihood $p(z_1|x_1)$ for every possible state x_1:

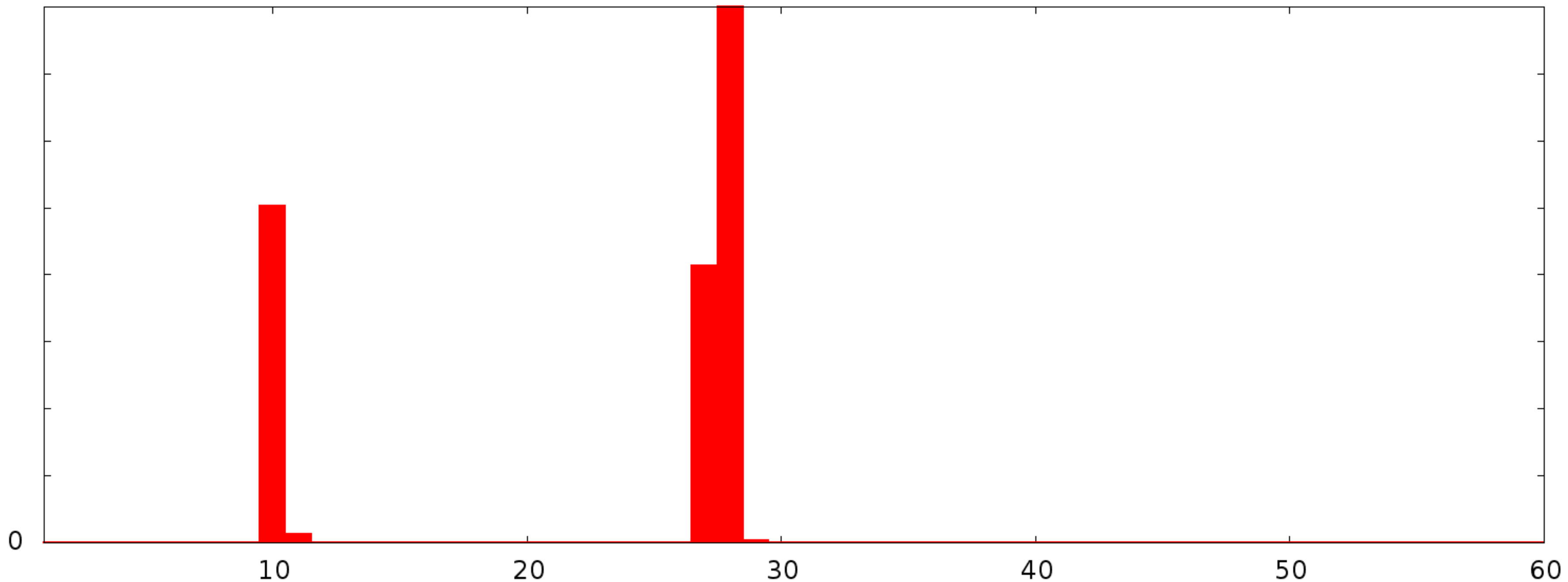


The Bayes Filter / Convolute and Multiply

$t = 1$

- Calculate the **posterior** by
  - **multiplying** with prior
  - Normalizing
- Reducing uncertainty

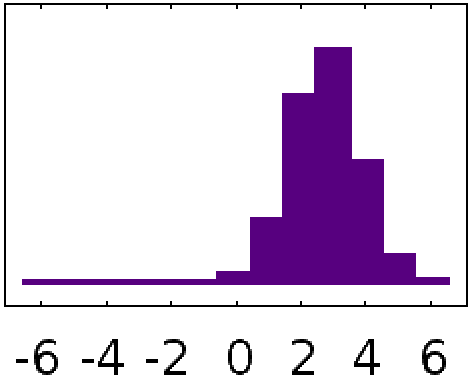
$$p(x_1|z_{0:1}) = \frac{p(x_1|z_0)p(z_1|x_1)}{\sum_{\hat{x}} p(x_1 = \hat{x}|z_0)p(z_1|x_1 = \hat{x})}$$

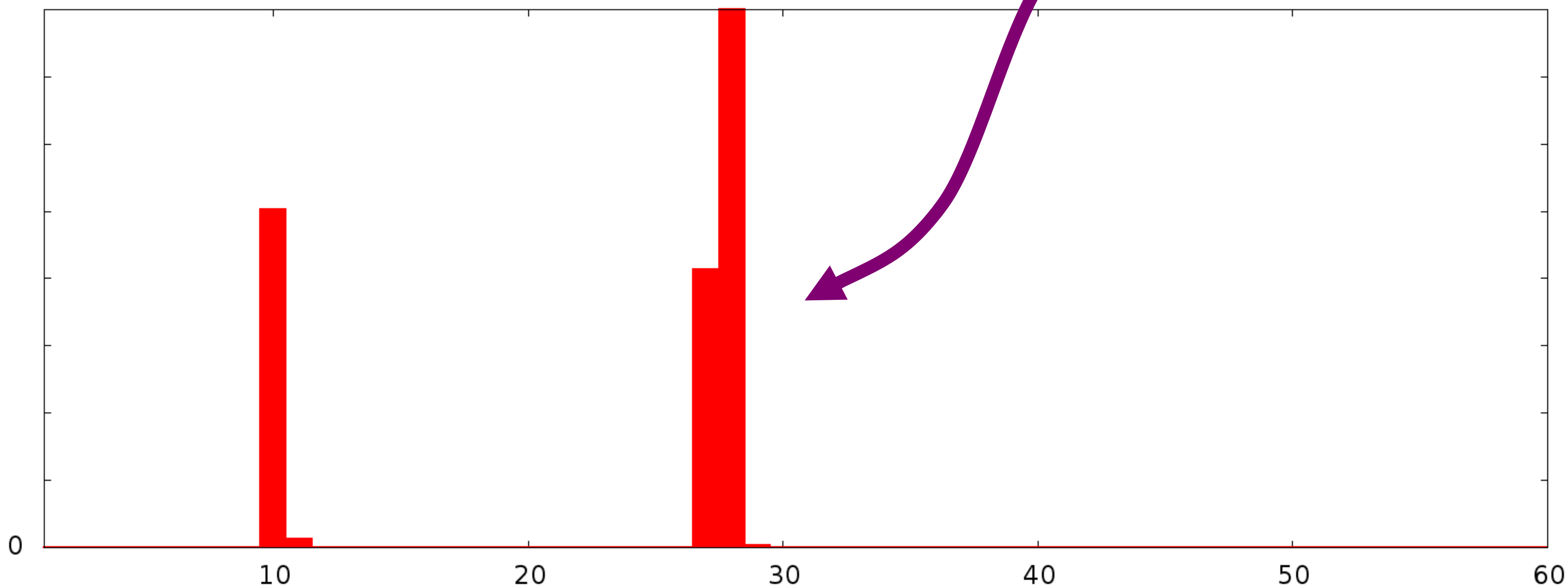


The Bayes Filter / Convolute and Multiply

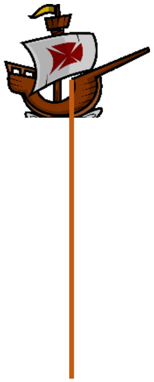
- Calculate the **prior** by **Convolution** with motion model
- Adding uncertainty

$$p(x_2|z_{0:1}) = \sum_{\hat{x}} p(x_1 = \hat{x}|z_{0:1})\, p(x_2|x_1 = \hat{x})$$

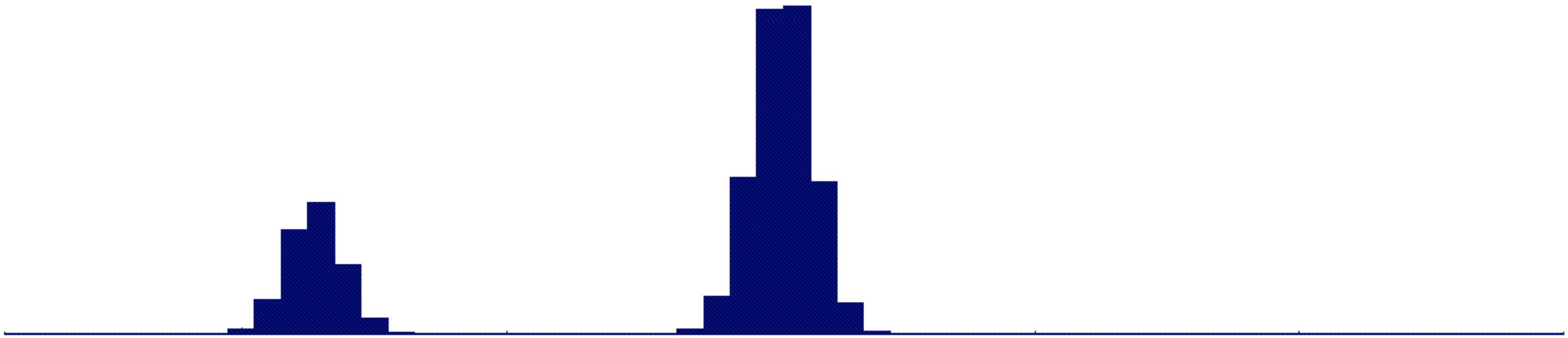




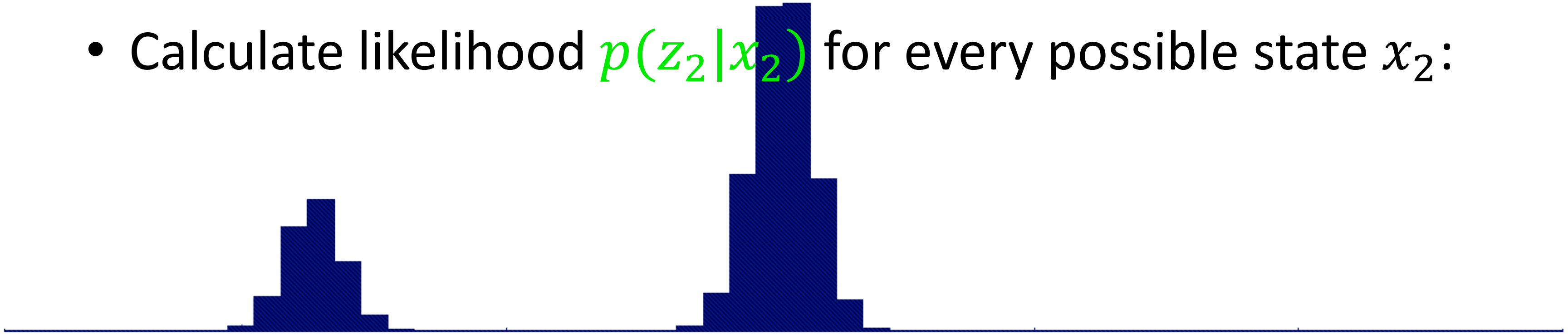
The Bayes Filter / Convolute and Multiply



- Make new measurement: z_2

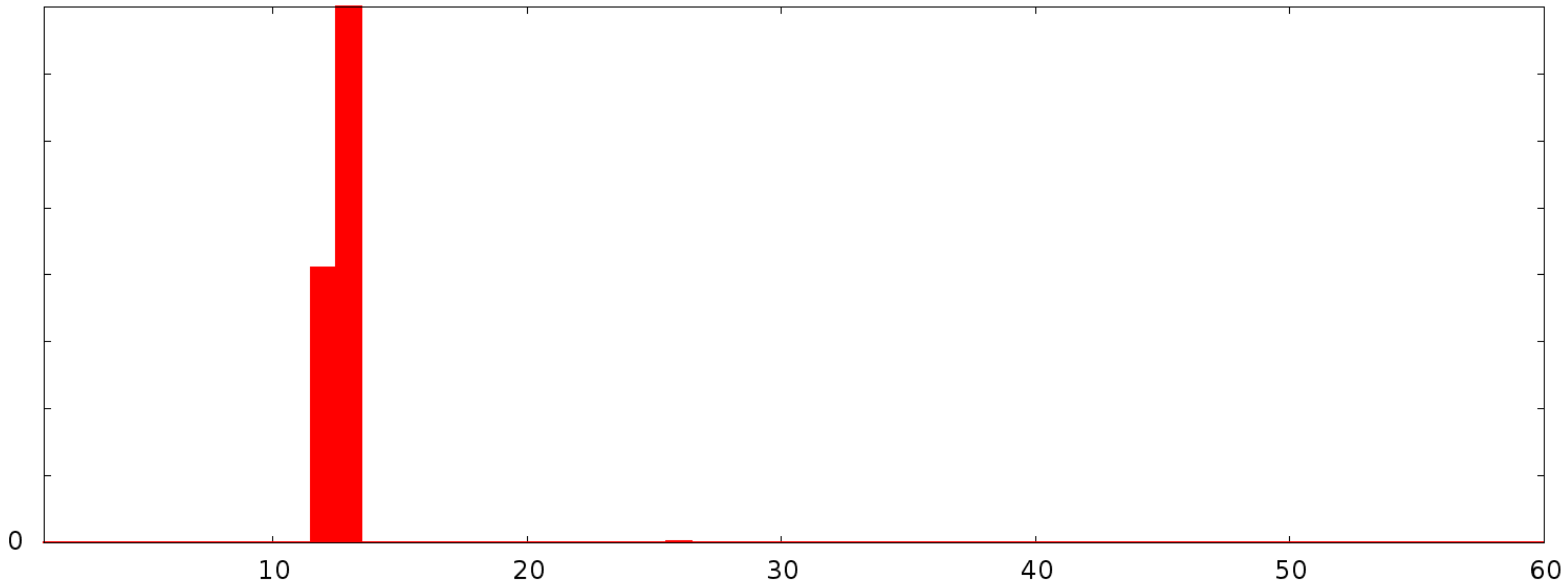


- Calculate likelihood $p(z_2|x_2)$ for every possible state x_2:



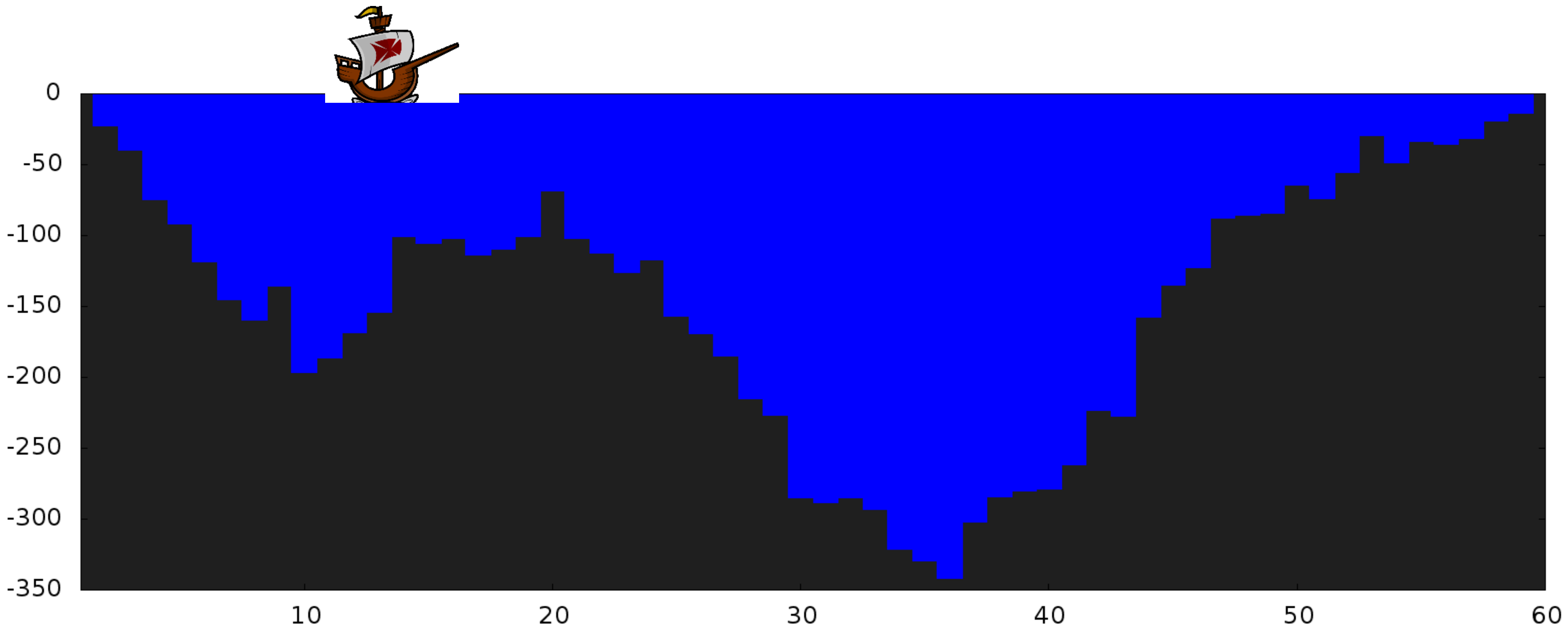
- Calculate the **posterior** by
  - **multiplying** with prior
  - Normalizing
- Reducing uncertainty

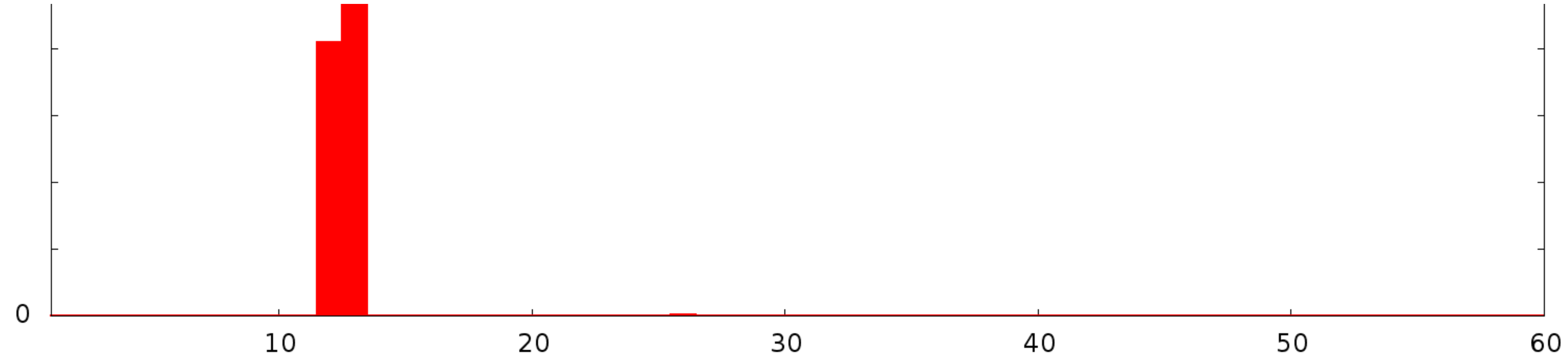
$$p(x_2|z_{0:2}) = \frac{p(x_2|z_{0:1})p(z_2|x_2)}{\sum_{\hat{x}} p(x_2 = \hat{x}|z_{0:1})p(z_2|x_2 = \hat{x})}$$



The Bayes Filter / Convolute and Multiply

$t = 2$





The Bayes Filter / Convolute and Multiply

Algorithm:

1. Make observation

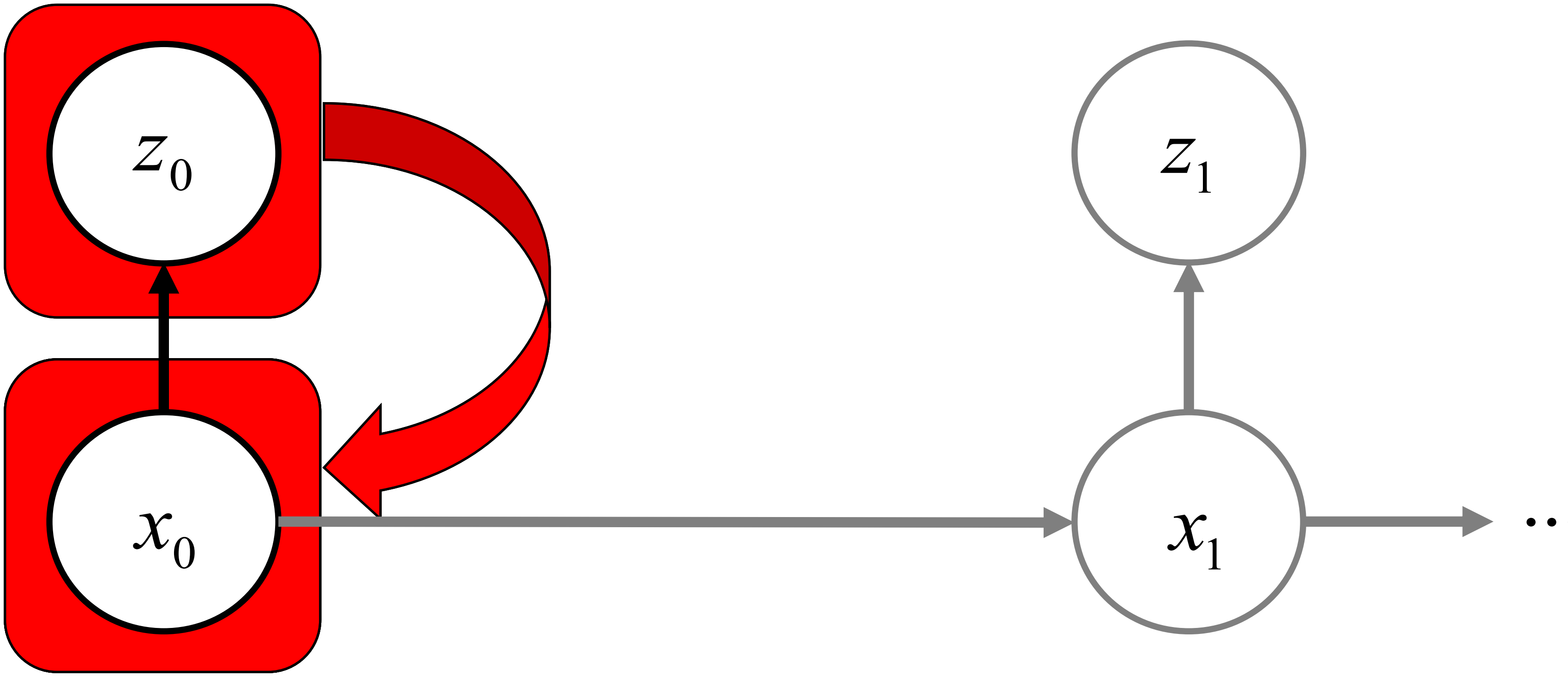
2. Calculate likelihood for every position

3. **Multiply** with last prior and normalize
   - Calculate posterior

4. **Convolution** with motion model
   - Calculate new prior

5. Go to 1.

Calculating the Posterior



$p(x_0)$

Observation model

(Multiply and normalize)

$$\boxed{p(x_0|z_0) = \frac{p(x_0)p(z_0|x_0)}{\sum_{\hat{x}} p(x_0 = \hat{x})p(z_0|x_0 = \hat{x})}}$$

(see proof 1)

Proof 1

Proof 1.:

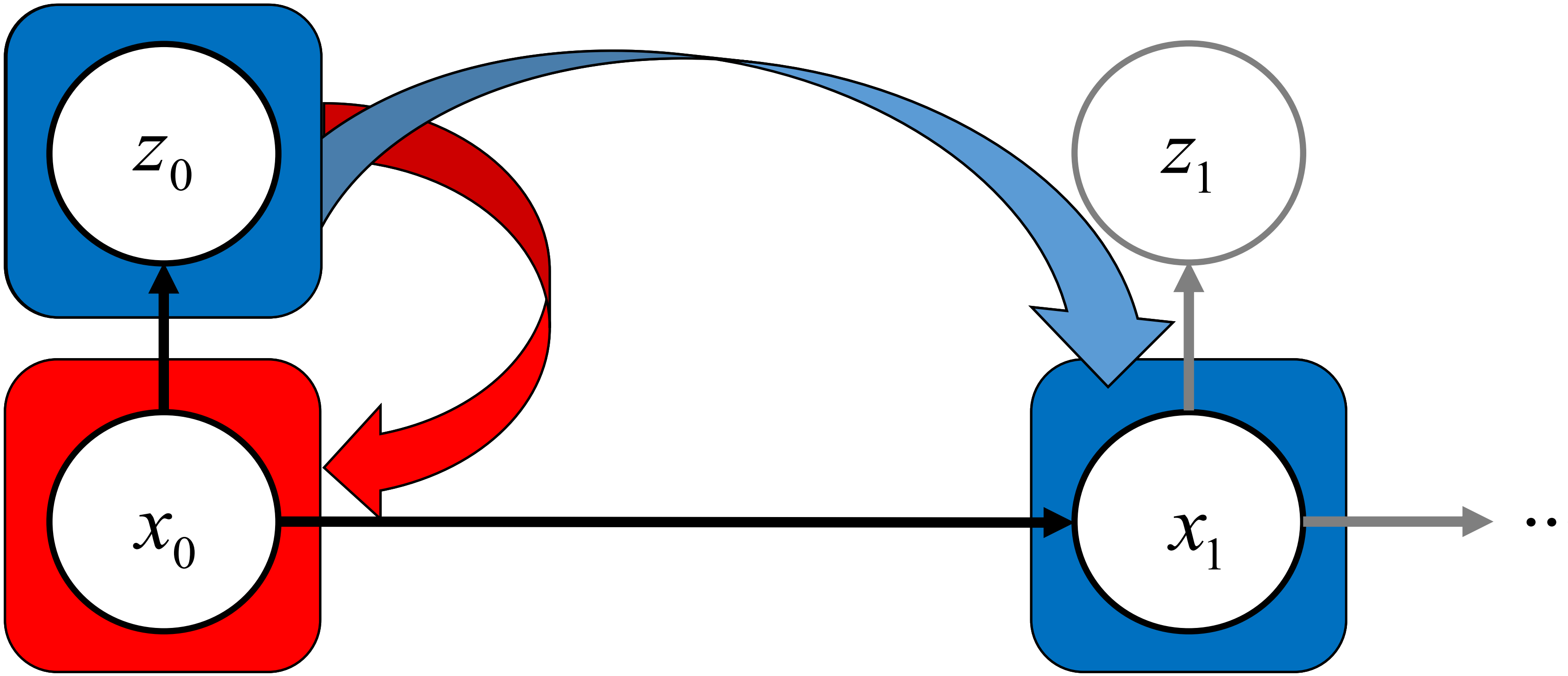
$$p(x_0|z_0) \overset{!}{=} \frac{\overbrace{p(x_0)\,p(z_0|x_0)}^{p(x_0,z_0)}}{\sum_{\hat{x}} \underbrace{p(x_0=\hat{x})\,p(z_0|x_0=\hat{x})}_{p(x_0=\hat{x},z_0)}}$$

$$= \frac{p(x_0,z_0)}{\underbrace{\sum_{\hat{x}} p(x_0=\hat{x},z_0)}_{p(z_0)}} \qquad |\ \text{marginalization}$$

$$= \frac{p(x_0,z_0)}{p(z_0)}$$

$$= p(x_0|z_0)$$

Calculating the next Prior



$p(x_0)$

Observation model

$p(x_0|z_0)$

Motion model

$$p(x_1|z_0) = \sum_{\hat{x}} p(x_0 = \hat{x}|z_0)\, p(x_1|x_0 = \hat{x})$$

(Convolution) (see proof 2)

Proof 2

Proof 2:

$$p(x_1|z_0) \overset{!}{=} \sum_{\hat{x}} p(x_0=\hat{x}|z_0)\underbrace{p(x_1|x_0=\hat{x})}_{p(x_1|x_0=\hat{x},z_0)}$$

x_1 and z_0 are cond. independent given x_0.

$\Downarrow$ it follows

$p(x_1|x_0=\hat{x}) = p(x_1|x_0=\hat{x},z_0)$

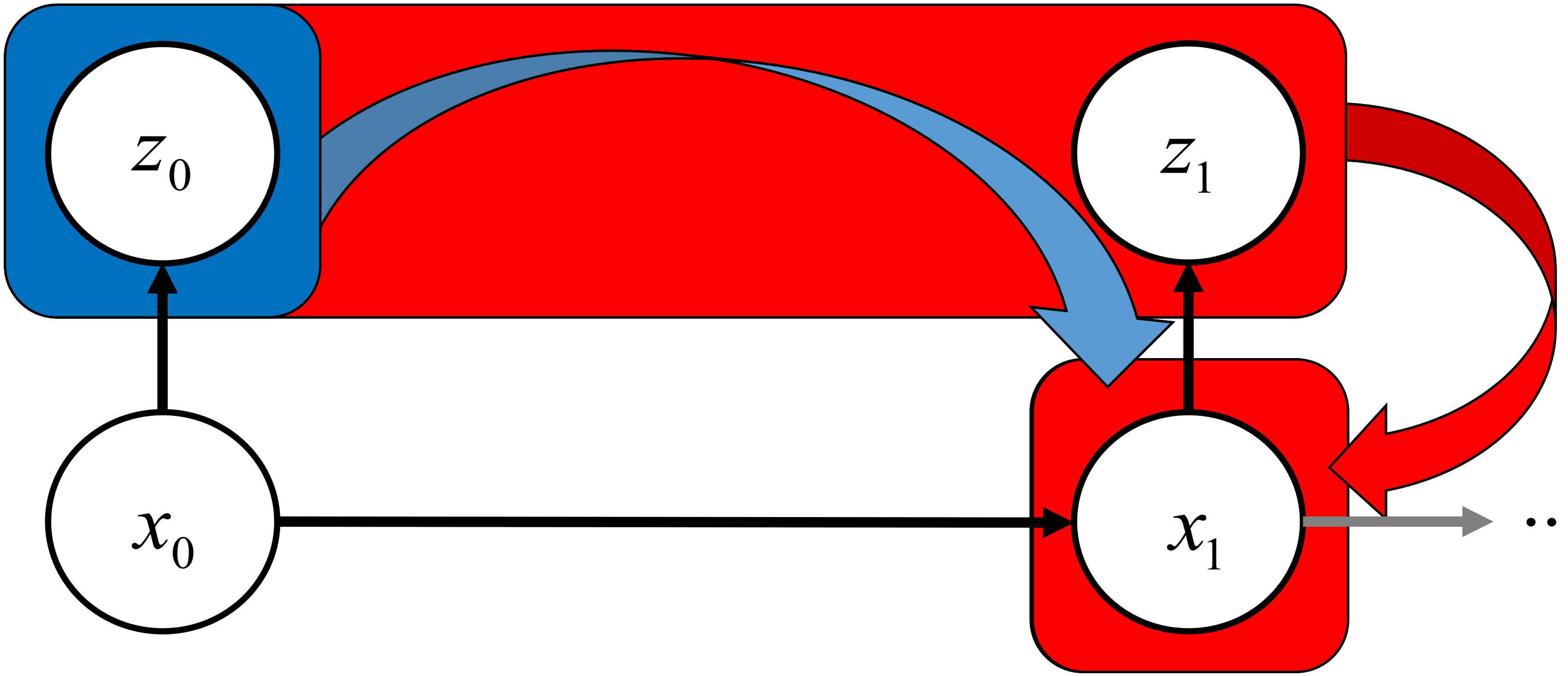
$$= \sum_{\hat{x}} p(x_0=\hat{x}|z_0)\, p(x_1|x_0=\hat{x},z_0)$$

both probs are conditioned on z_0..

$$= \sum_{\hat{x}} p(x_1,x_0=\hat{x}|z_0)$$

$$= p(x_1|z_0)$$

Calculating the Posterior



$p(x_0)$

Observation model

$p(x_0|z_0)$

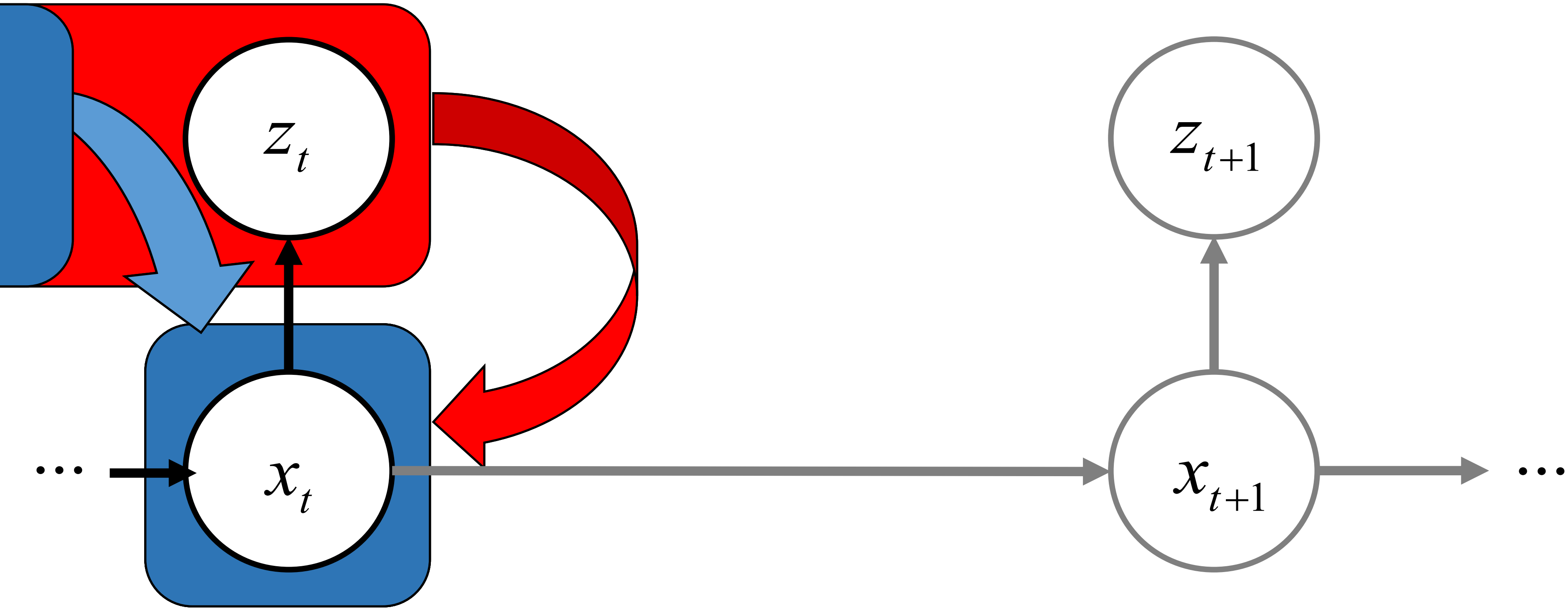
Motion model

$p(x_1|z_0)$

Observation model

$$p(x_1|z_{0:1}) = \frac{p(x_1|z_0)p(z_1|x_1)}{\sum_{\hat{x}} p(x_1 = \hat{x})p(z_1|x_1 = \hat{x})}$$

Calculating the Posterior (General Case)



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

(**Multiply** and normalize)

Observation model

$$\boxed{p(x_t|z_{0:t}) = \frac{p(x_t|z_{0:t-1})p(z_t|x_t)}{\sum_{\hat{x}} p(x_t = \hat{x}|z_{0:t-1})p(z_t|x_t = \hat{x})}}$$

(see proof 3)

Proof 3.:

$$P(x_t|z_{0:t}) \overset{!}{=} \frac{P(x_t|z_{0:t-1})\overbrace{P(z_t|x_t)}^{P(z_t|x_t,z_{0:t-1})}}{\sum_{\hat{x}} P(x_t=\hat{x}|z_{0:t-1})\underbrace{P(z_t|x_t=\hat{x})}_{P(z_t|x_t=\hat{x},z_{0:t-1})}}$$

| z_t and $z_0,\dots,z_{t-1}$ are cond. independent given x_t.
$\Downarrow$ it follows
$P(z_t|x_t) = P(z_t|x_t,z_{0:t-1})$ and
$P(z_t|x_t=\hat{x}) = P(z_t|x_t=\hat{x},z_{0:t-1})$

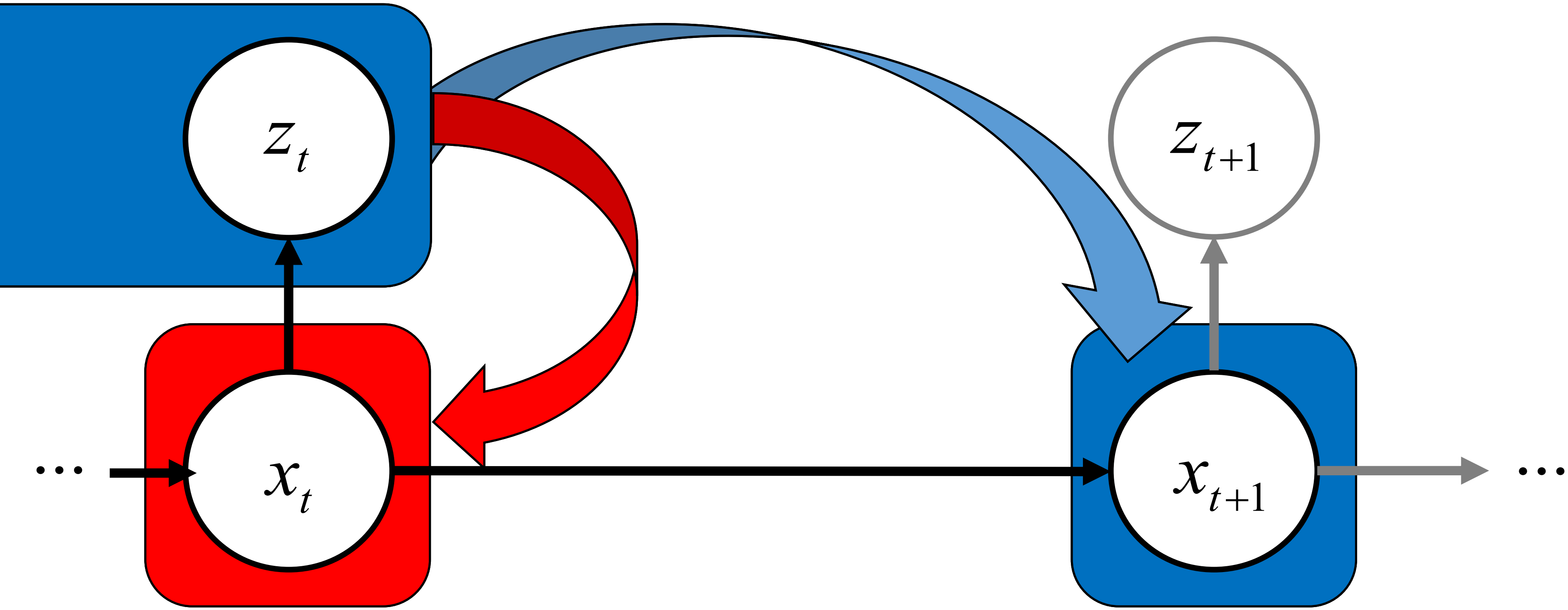
$$= \frac{P(x_t|z_{0:t-1})\,P(z_t|x_t,z_{0:t-1})}{\sum_{\hat{x}} P(x_t=\hat{x}|z_{0:t-1})P(z_t|x_t=\hat{x},z_{0:t-1})}$$

$$= \frac{P(x_t,z_t|z_{0:t-1})}{\underbrace{\sum_{\hat{x}} P(x_t=\hat{x},z_t|z_{0:t-1})}_{P(z_t|z_{0:t-1})}} \quad \Big|\ \text{(marginalization)}$$

$$= \frac{P(x_t,z_t|z_{0:t-1})}{P(z_t|z_{0:t-1})}$$

$$= P(x_t|z_{0:t})$$

Calculating the next Prior (General Case)



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

Observation model

$p(x_t | z_{0:t})$

Motion model

$$p(x_{t+1} | z_{0:t}) = \sum_{\hat{x}} p(x_t = \hat{x} | z_{0:t}) p(x_{t+1} | x_t = \hat{x})$$

(Convolution)

(see proof 4)

Proof 4

Proof 4:

$$p(x_{t+1} \mid z_{0:t}) \overset{!}{=} \sum_{\hat{x}} p(x_t = \hat{x} \mid z_{0:t}) \underbrace{p(x_{t+1} \mid x_t = \hat{x})}_{p(x_{t+1} \mid x_t = \hat{x}, z_{0:t})}$$

x_{t+1} and $z_0, \ldots, z_t$ ~~$z_0, \ldots, z_t$~~ are cond. independent given x_t

$\Downarrow$ it follows:

$$p(x_{t+1} \mid x_t = \hat{x}) = p(x_{t+1} \mid x_t = \hat{x}, z_{0:t})$$

$$= \sum_{\hat{x}} p(x_t = \hat{x} \mid z_{0:t})\, p(x_{t+1} \mid x_t = \hat{x}, z_{0:t})$$

$$= \sum_{\hat{x}} p(x_t = \hat{x}, x_{t+1} \mid z_{0:t})$$

| (marginalization)

$$= p(x_{t+1} \mid z_{0:t})$$

Overview Continuous Approaches

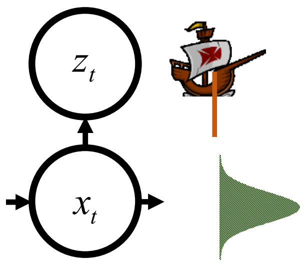
- How to apply it in continuous space?

- Two popular alternatives:
  - Particle filter
    - Represent **prior** and **posterior** with samples
  - Kalman Filter
    - Represent **prior** and **posterior** distributions as Gaussians

Important Distributions (Particle Filter)

Observation Model:

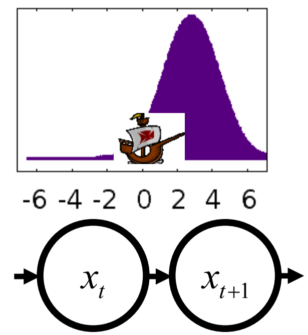
$p(z_t|x_t)$



- Likelihood of observation given state
- Continuous Gaussian around real depth

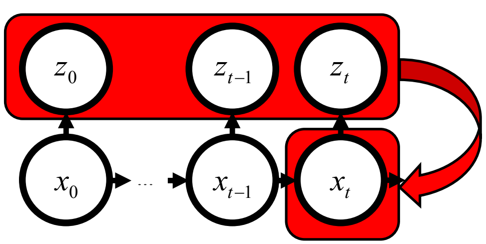
Motion Model:

$p(x_{t+1}|x_t)$



- Probability of new state given old one
- **Continuous Gaussian**

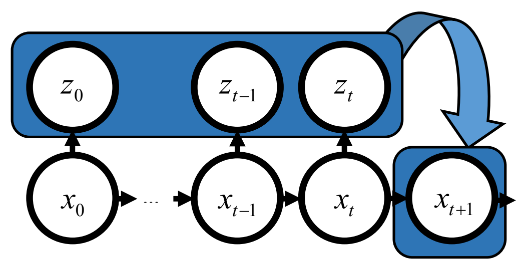
Posterior:



$p(x_t|z_{0:t})$

- Probability of state given previous and current observations
- **Continuous, represented as set of Samples (Particles)**

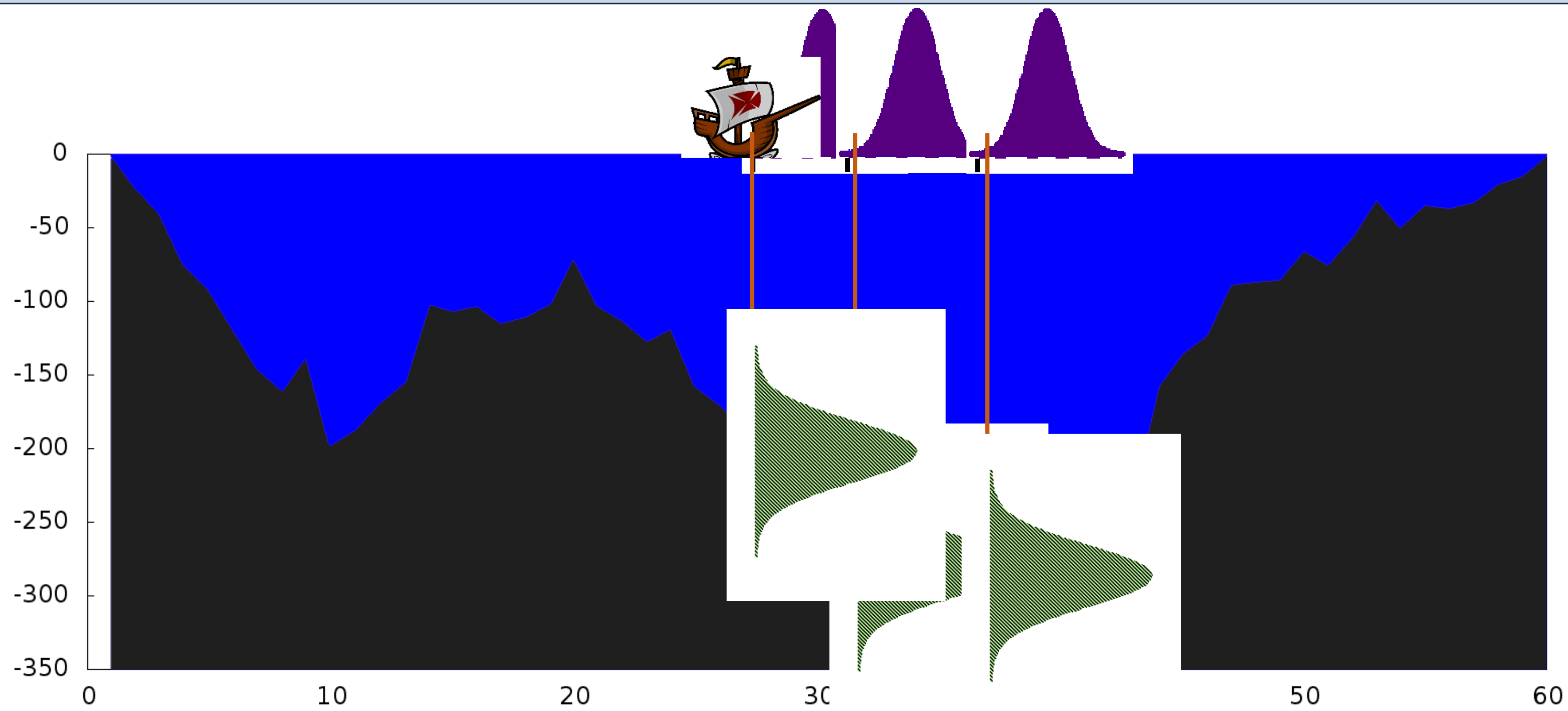
Prior:



$p(x_{t+1}|z_{0:t})$

- Probability of state given only previous observations
- **Continuous, represented as set of Samples (Particles)**

Particle Filter



Discrete Bayes Filter vs. Particle Filter

- Discrete Bayes Filter:

1. Make observation

2. Calculate likelihood for every position

3. **Multiply** with last prior and normalize

4. **Convolution** with motion model

5. Go to 1.

- Particle Filter:

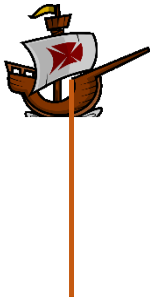
1. Make observation

2. Calculate likelihood for every **sample** -> weights

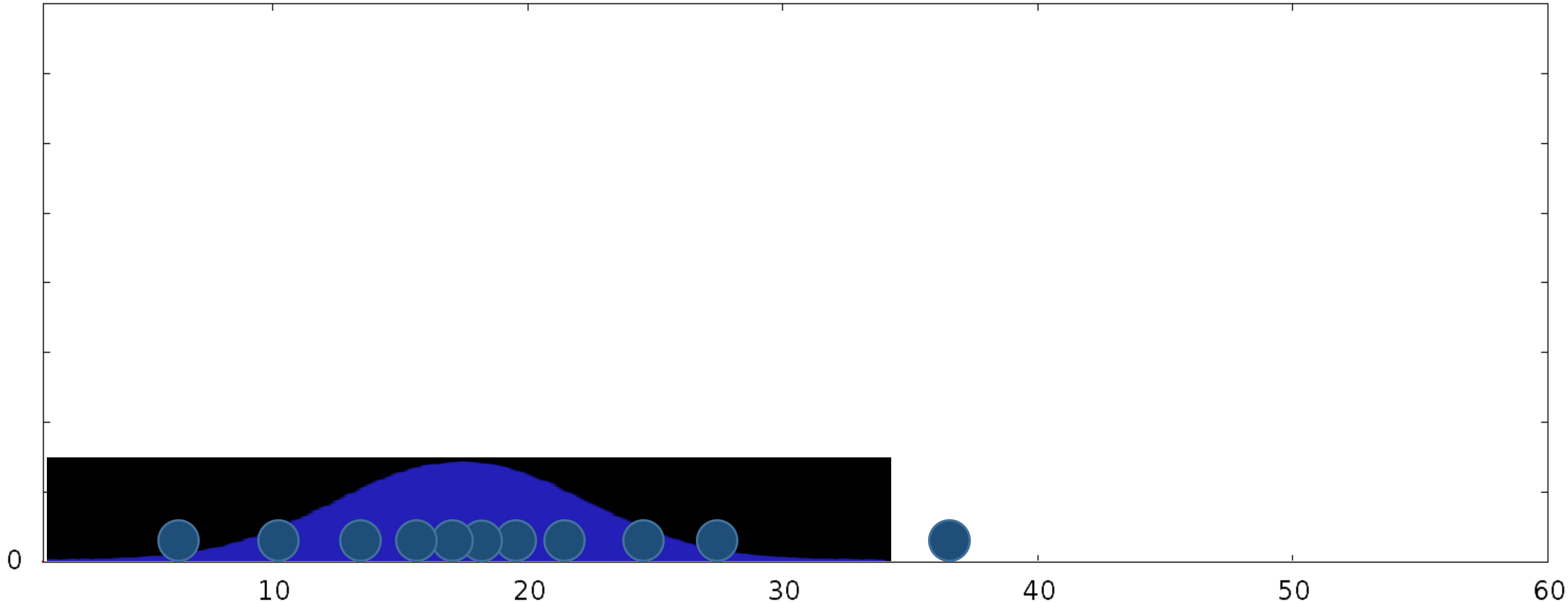
3. **Resampling** according to weights

4. Randomly move samples according to motion model (**Sampling**)

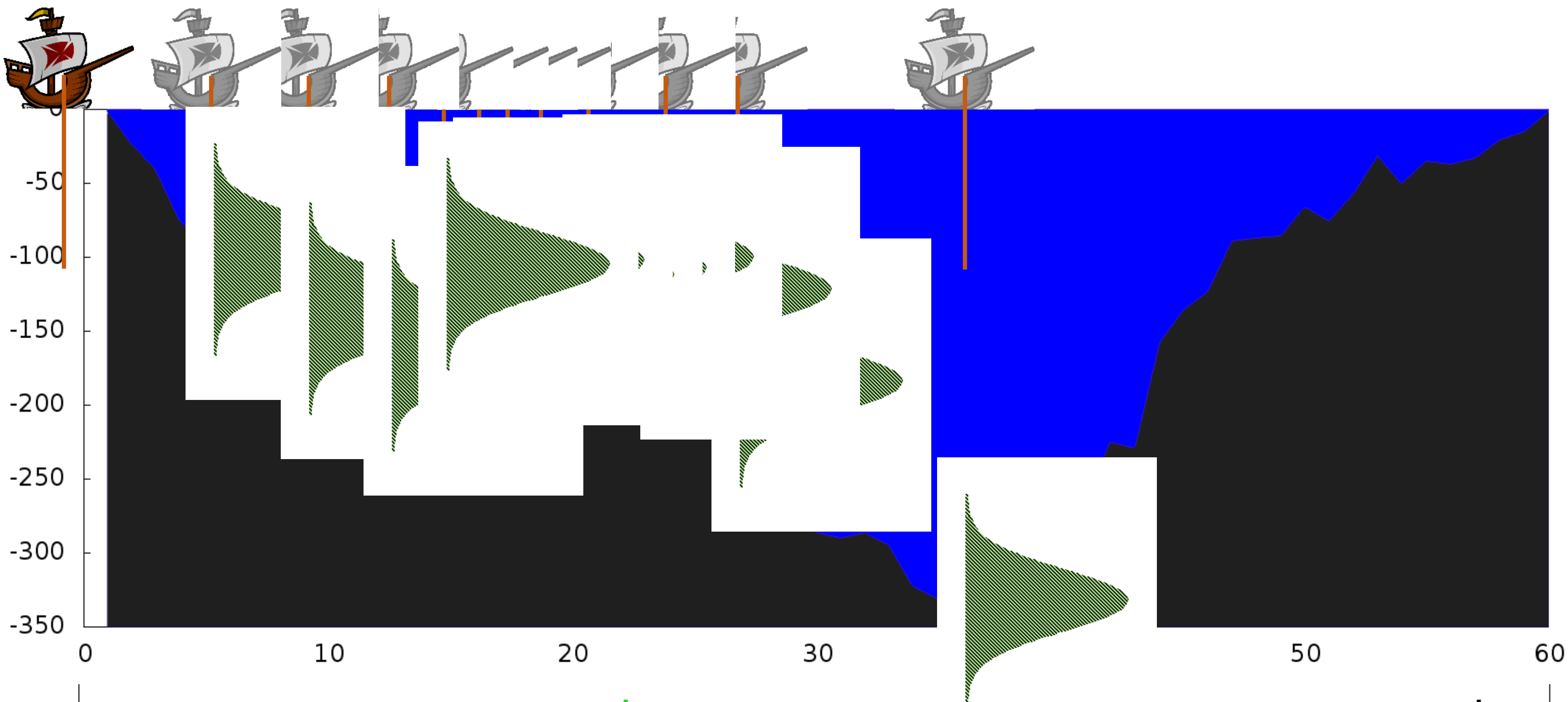
5. Go to 1.



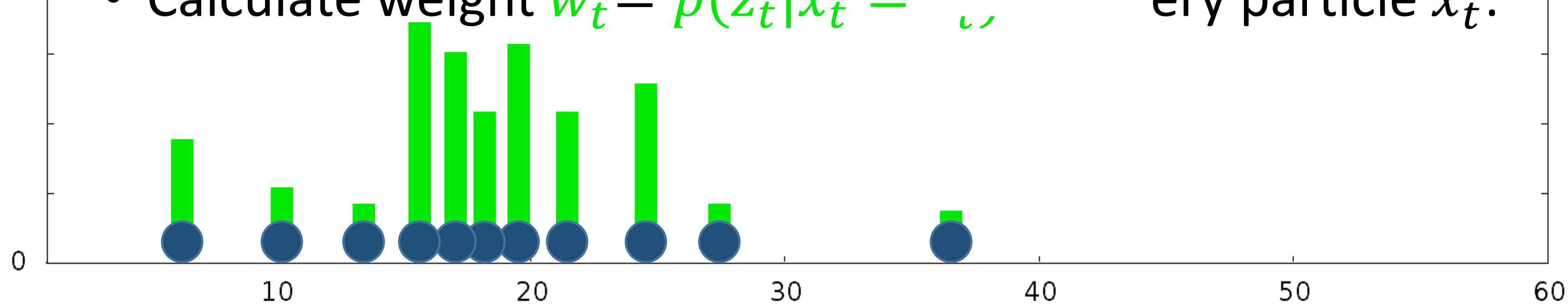
- Represent continuous prior with particles $\tilde{x}_t^1, \ldots, \tilde{x}_t^n$
- Make measurement: z_t



Particle Filter / Resampling

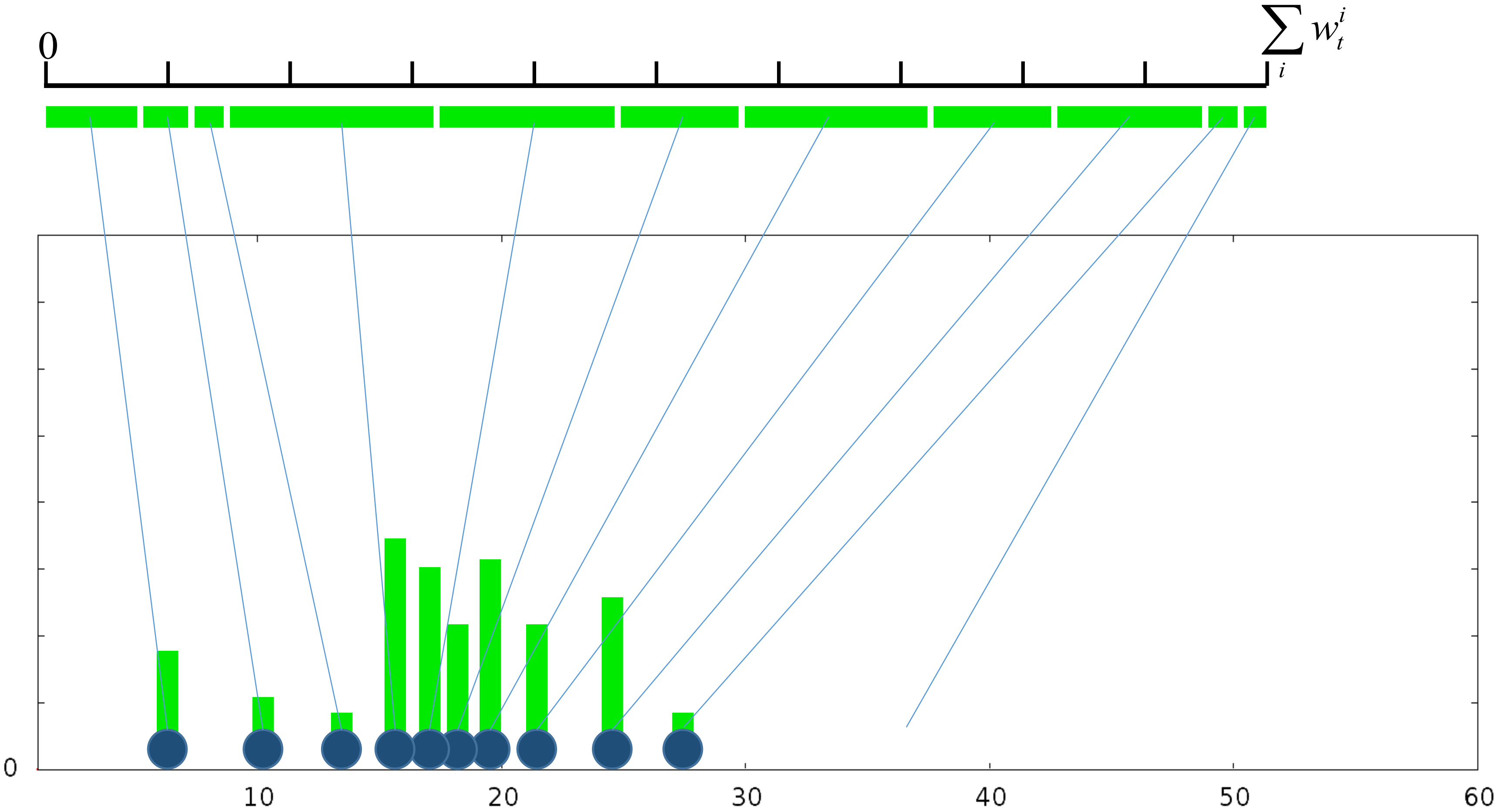


- Calculate weight $w_t^i = p(z_t|x_t =$ … ery particle $\tilde{x}_t^i$:



Particle Filter / Resampling

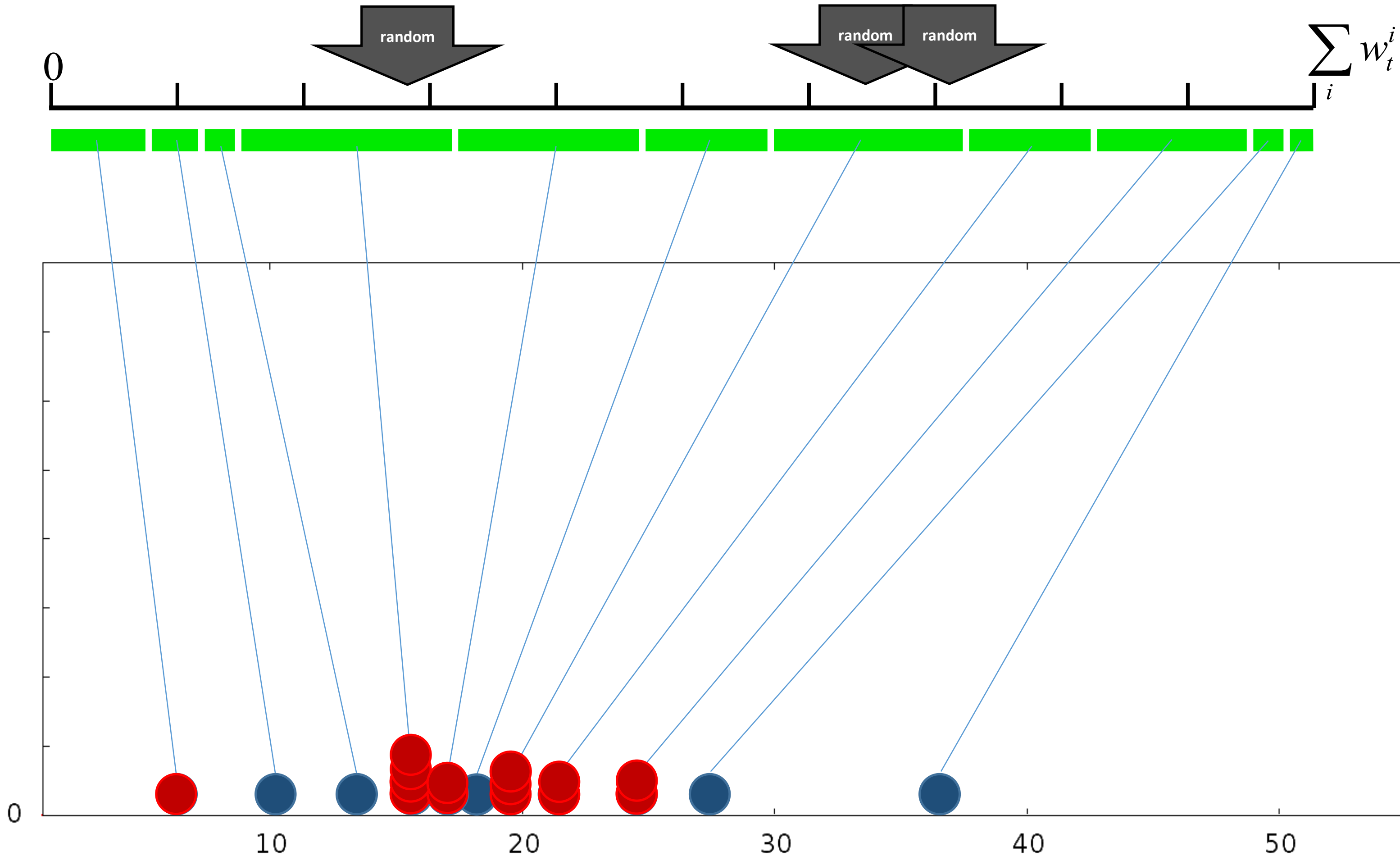
- Draw samples $x_t^1, \dots, x_t^n$ from **posterior** by resampling from $\tilde{x}_t^1, \dots, \tilde{x}_t^n$ using the weights w_t^i
- Reducing uncertainty



Particle Filter / Resampling

$t = 0$

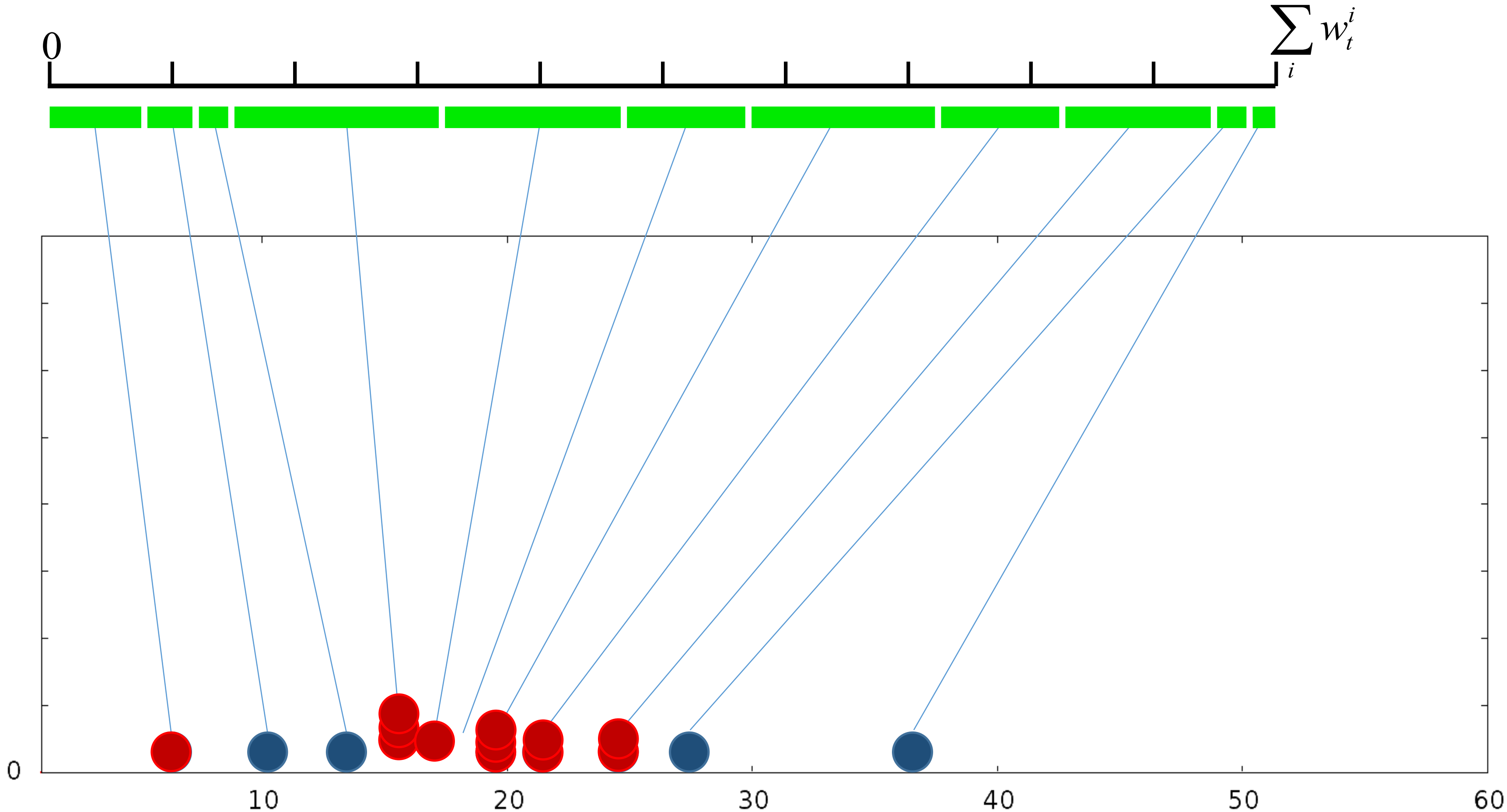
- Draw samples $x_t^1, \ldots, x_t^n$ from **posterior** by resampling from $\tilde{x}_t^1, \ldots, \tilde{x}_t^n$ using the weights w_t^i
- Reducing uncertainty



Particle Filter / Resampling

$t = 0$

- Draw samples $x_t^1, \ldots, x_t^n$ from **posterior** by resampling from $\tilde{x}_t^1, \ldots, \tilde{x}_t^n$ using the weights w_t^i
- Reducing uncertainty



Why is this allowed?

Resampling is like **multiplication** and **normalization**:

- Sample density in **posterior** depends linearly on
  - Density of **prior** samples $\tilde{x}_t^1, \ldots, \tilde{x}_t^n$
  - Likelihood w_t^i

Discrete Bayes Filter vs. Particle Filter

- Discrete Bayes Filter:

1. Make observation

2. Calculate likelihood for every position

3. **Multiply** with last prior and normalize

4. **Convolution** with motion model

5. Go to 1.

- Particle Filter:

1. Make observation

2. Calculate likelihood for every **sample** -> weights

3. **Resampling** according to weights

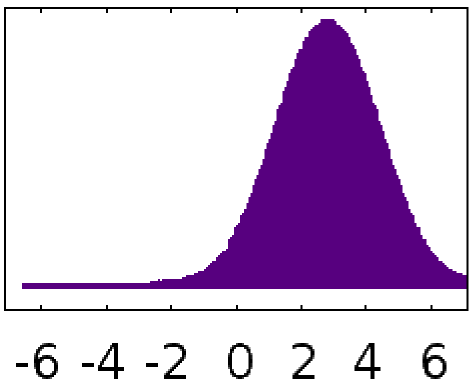
4. Randomly move samples according to motion model (**Sampling**)

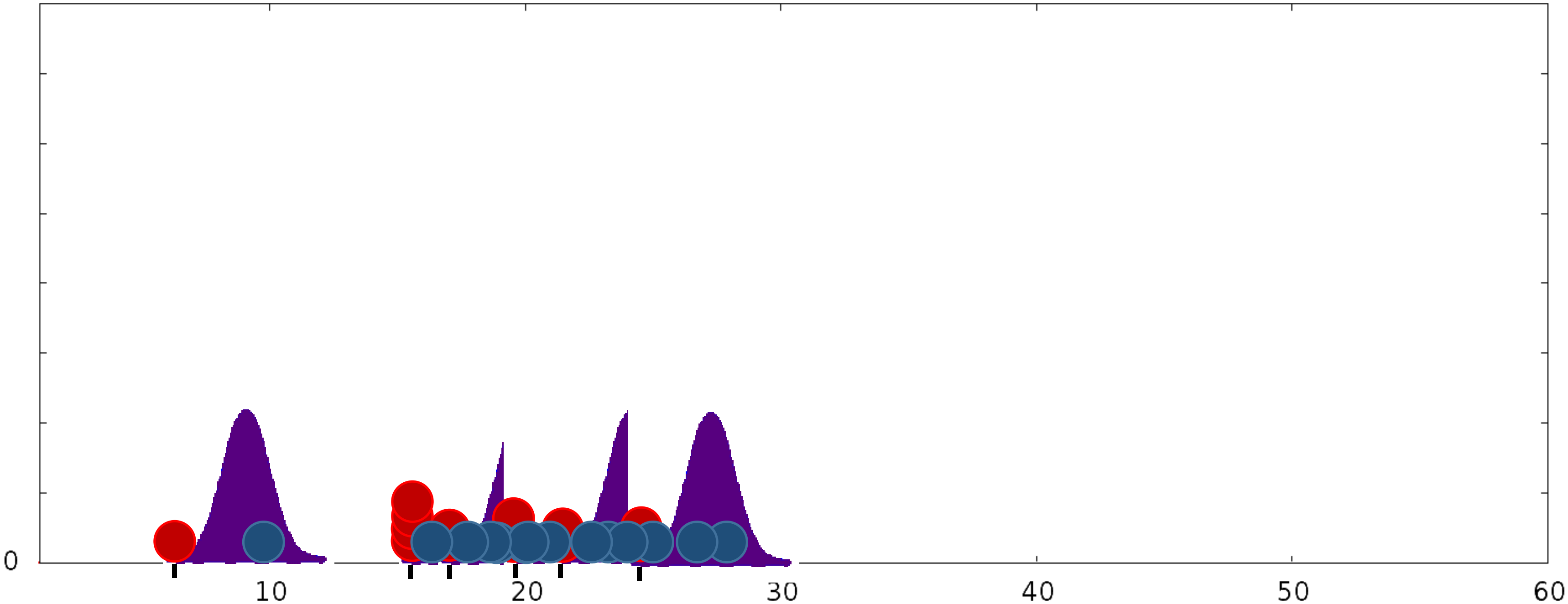
5. Go to 1.

Particle Filter / Sampling

$t = 0$

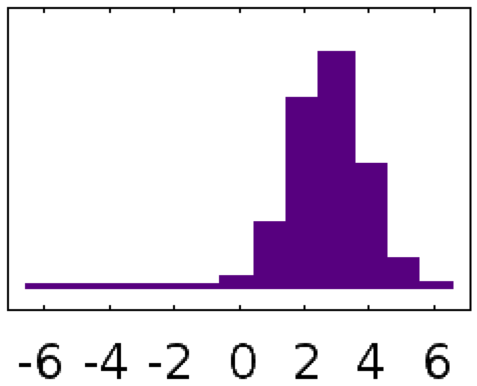
- Obtain samples $\tilde{x}_{t+1}^{1}, \dots, \tilde{x}_{t+1}^{n}$ from the **new prior** by moving each particle according to motion model
- Adding uncertainty

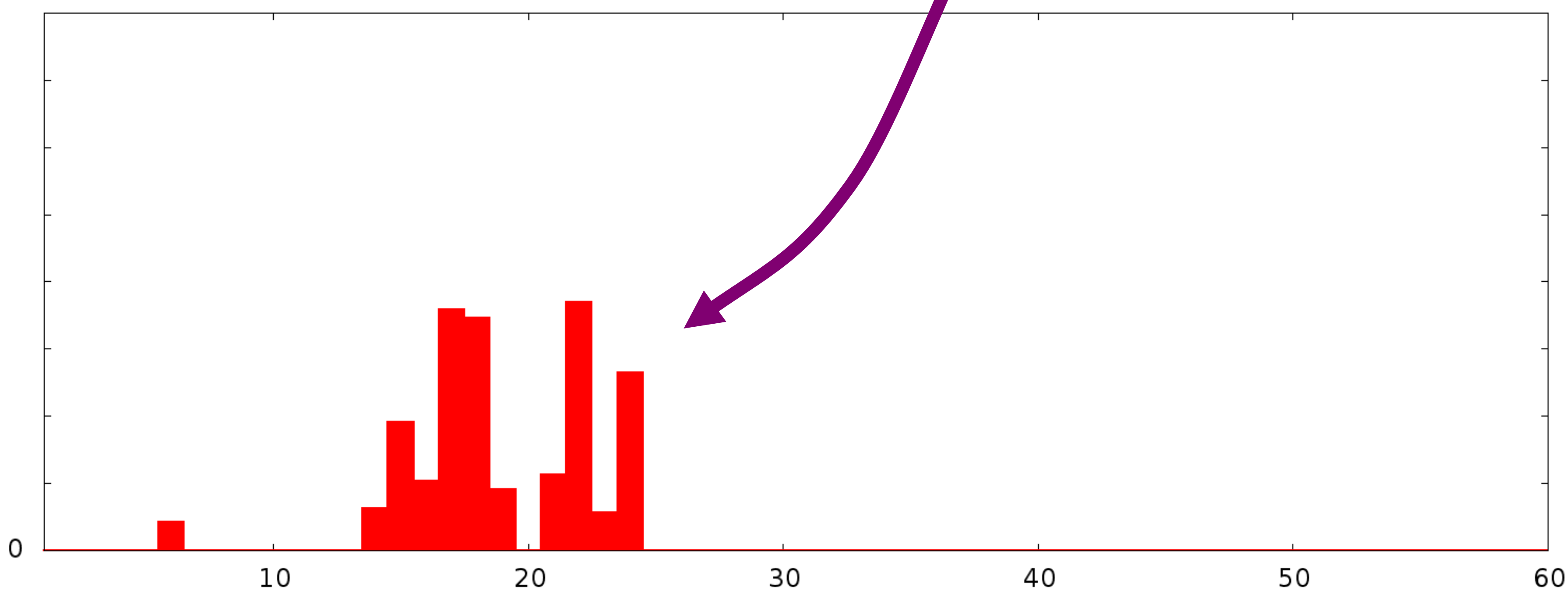




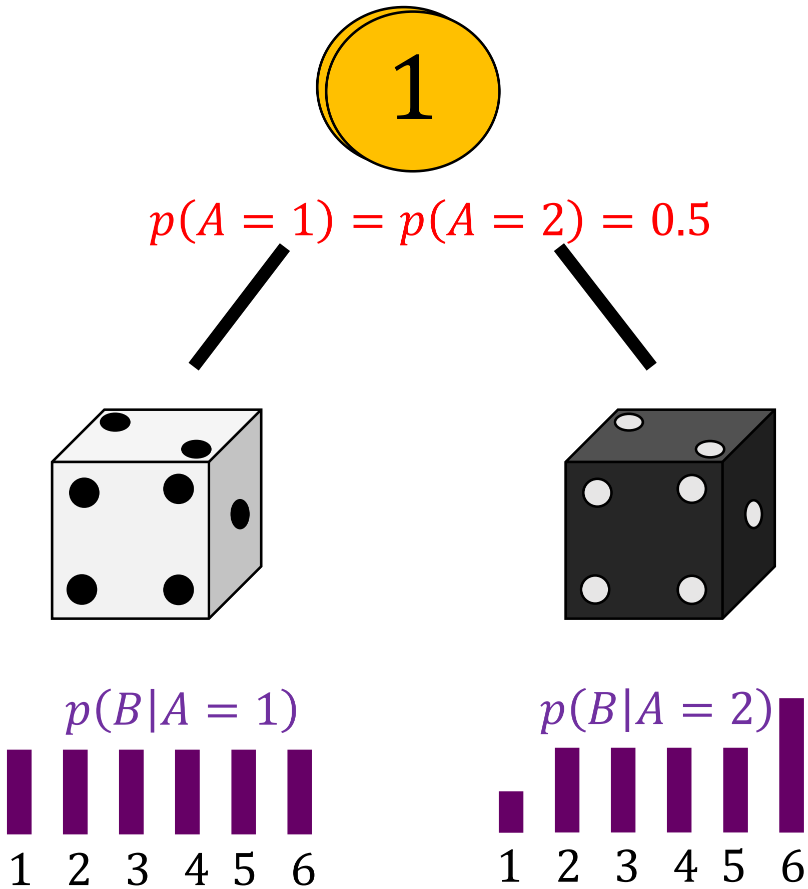
- Calculate the **prior** by **Convolution** with motion model
- Adding uncertainty

$$p(x_1|z_0) = \sum_{\hat{x}} p(x_0 = \hat{x}|z_0)p(x_1|x_0 = \hat{x})$$

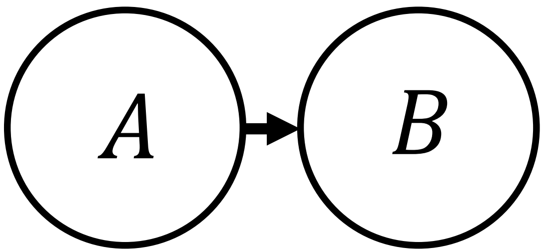




Why is this allowed?

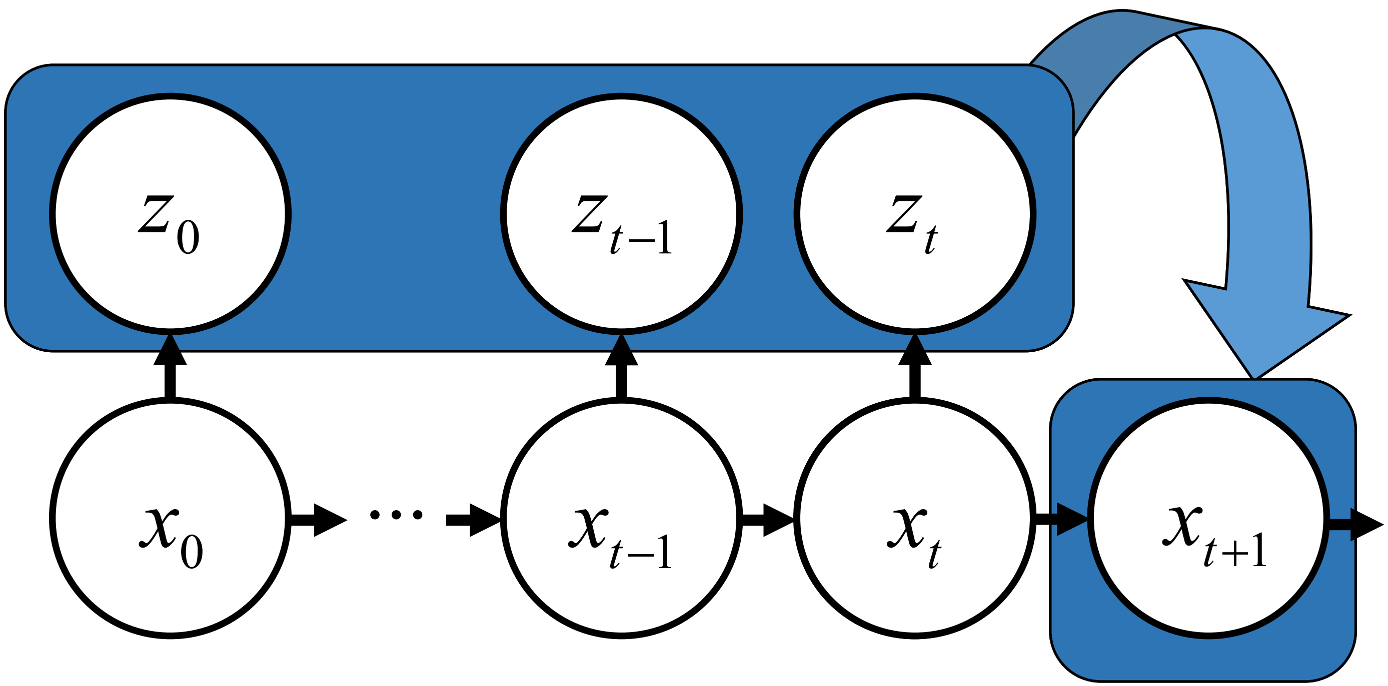


- We want to sample: $\mathrm{B}^{\mathrm{i}} \sim p(B)$
  - Don‘t know $p(B)$
- We sample first: $A^i \sim p(A)$
- And then $B^i \sim p\left(B|A=A^i\right)$



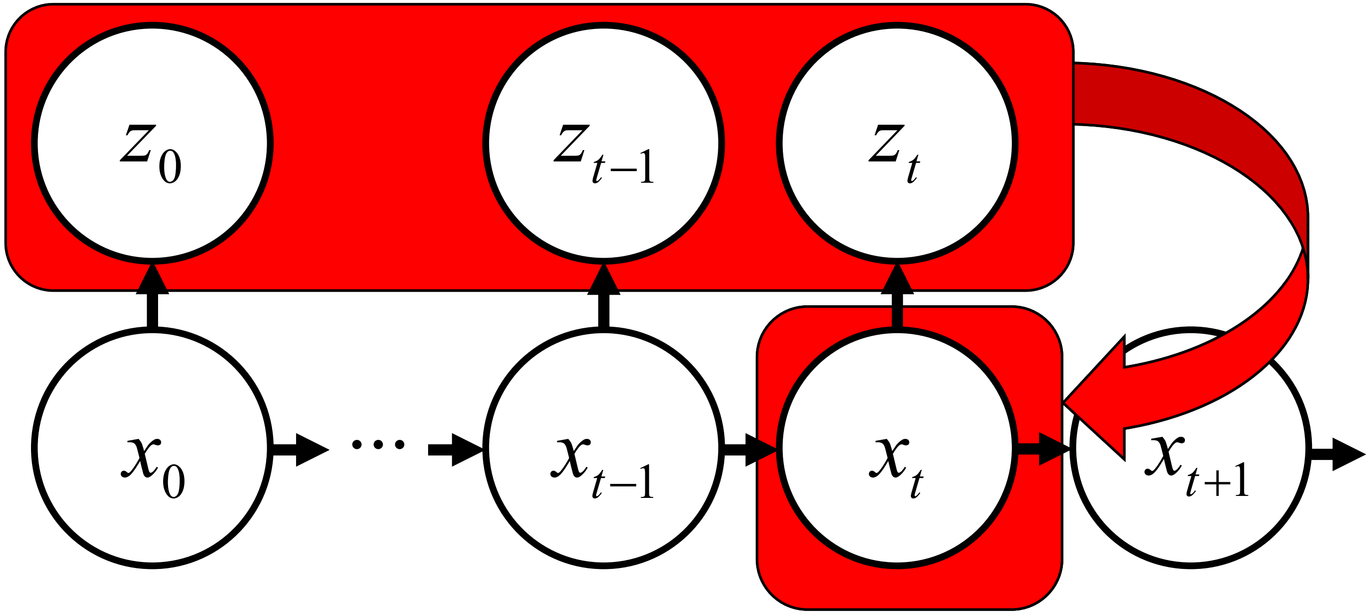
Why is this allowed?

- We want to sample $\tilde{x}_{t+1}^{i} \sim p(x_{t+1}|z_{0:t})$
  - Don‘t know $p(x_{t+1}|z_{0:t})$



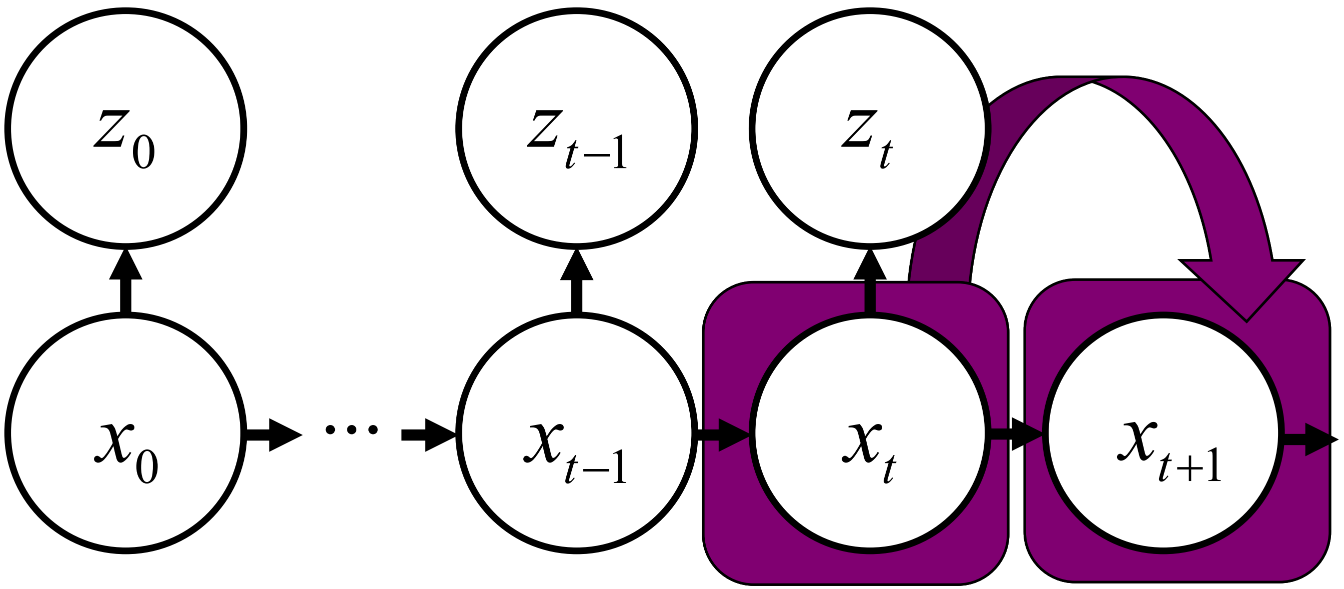
Why is this allowed?

- We want to sample $\tilde{x}_{t+1}^{i} \sim p(x_{t+1}|z_{0:t})$
    - Don‘t know $p(x_{t+1}|z_{0:t})$
- We use samples $x_t^i \sim p(x_t|z_{0:t})$



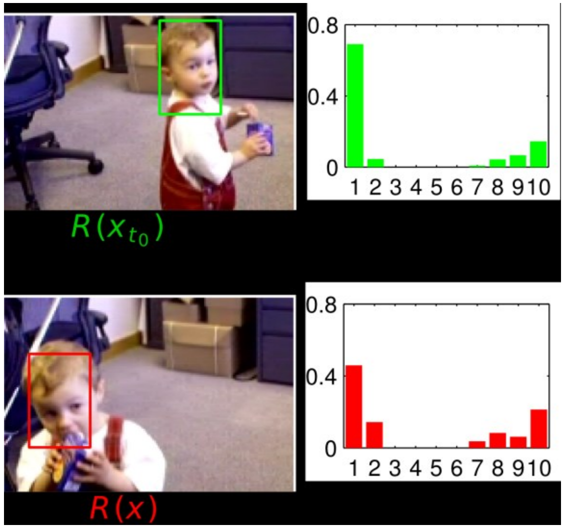
Why is this allowed?

- We want to sample $\tilde{x}_{t+1}^{i} \sim p(x_{t+1}|z_{0:t})$
  - Don't know $p(x_{t+1}|z_{0:t})$
- We use samples $x_t^i \sim p(x_t|z_{0:t})$
- We sample $x_{t+1}^i \sim p\left(x_{t+1} \middle| x_t = x_t^i\right) = p\left(x_{t+1} \middle| x_t = x_t^i, z_{0:t}\right)$



Particle Filter (Tracking application)

- Tracking an object in a video sequence [Perez at al. 2002]
- States:
  - 2D windows (location and size)
- Gaussian motion
- Observations:
  - Color histograms

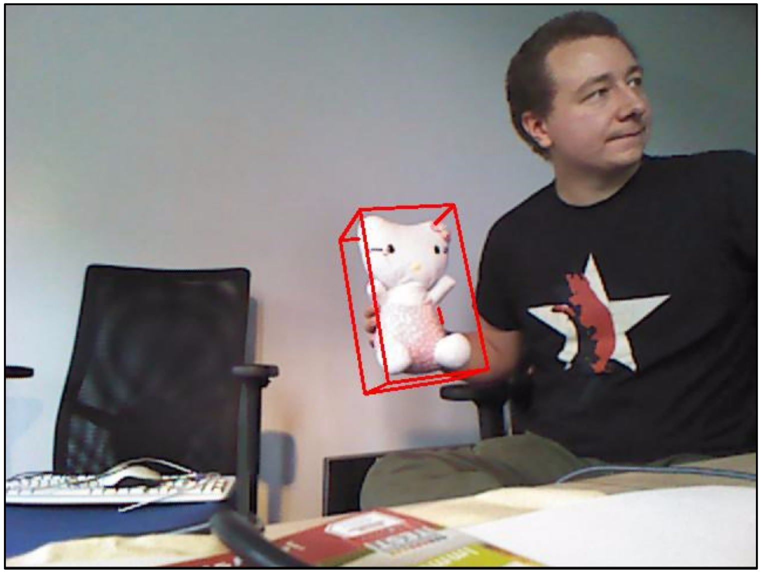


Particle Filter (Tracking application)

- Pose tracking of an object in a kinect sequence [Krull at al. 2014]

- States:
  - 6D Pose
    - 3D position
    - 3D rotation

- Observations:
  - Depth images
  - Predicted Object coordinates



Particle Filter (Summary)

- The Particle Filter implements the Bayes Filter

- **Prior** and **posterior** are represented as sample sets (particle)
  - **Likelihood** is only evaluated at particles
  - Multiplication -> weighted resampling
  - Convolution -> random movement according to **motion model**

Overview Continuous Approaches

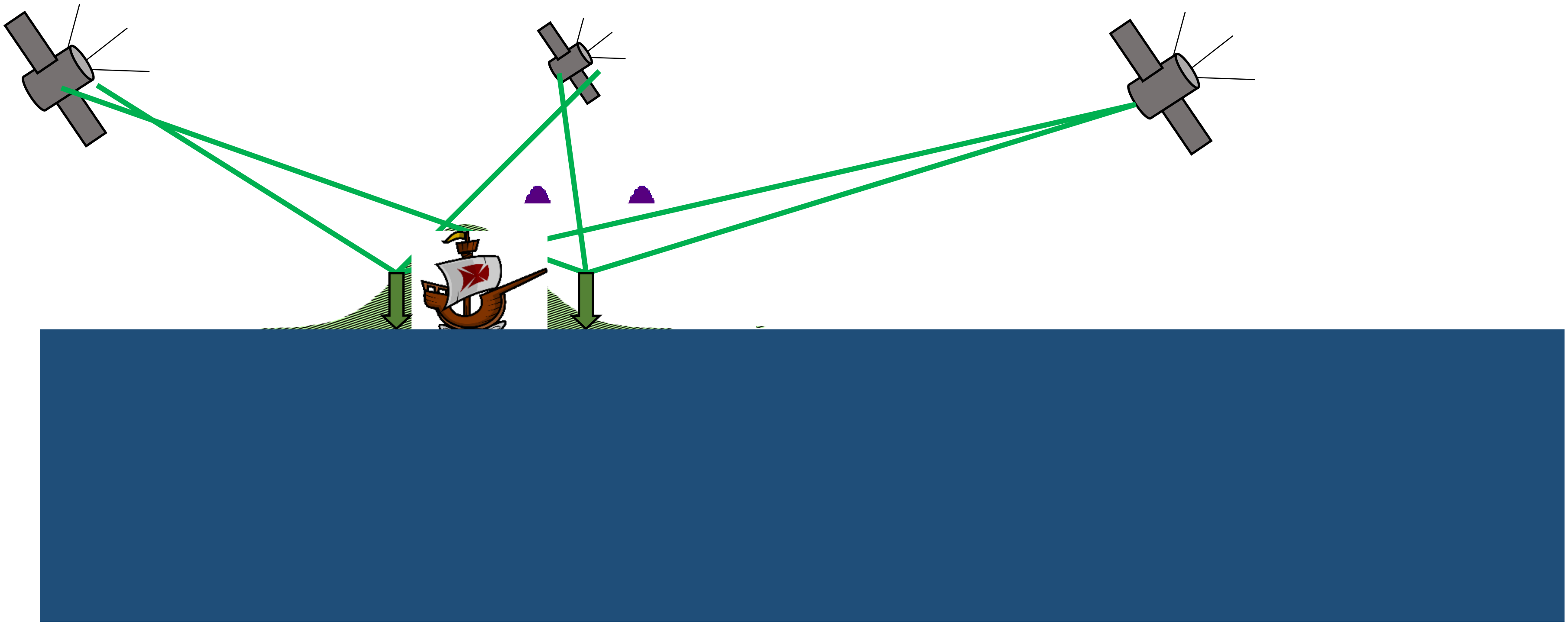
- How to apply it in continuous space?

- Two popular alternatives:

  - Particle filter
    - Represent **prior** and **posterior** with samples

  - Kalman Filter
    - Represent **prior** and **posterior** distributions as Gaussians

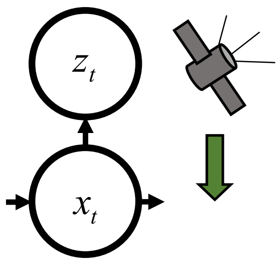
Kalman Filter



Important Distributions (Kalman Filter)

Observation Model:

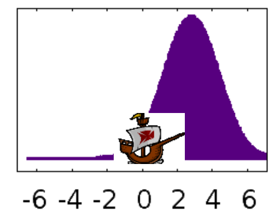
$p(z_t|x_t)$



- Likelihood of observation given state
- Continuous Gaussian **around real position**

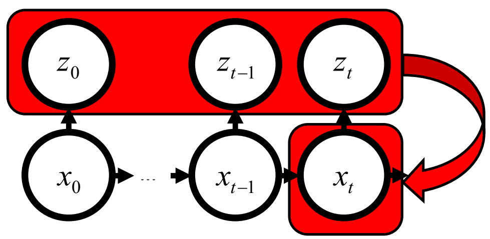
Motion Model:

$p(x_{t+1}|x_t)$



- Probability of new state given old one
- Continuous Gaussian

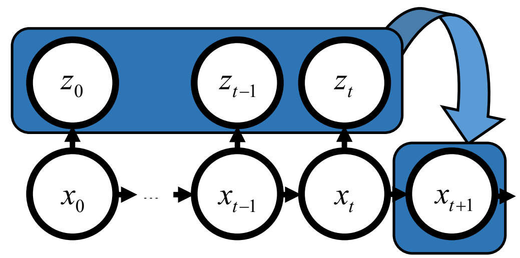
Posterior:



$p(x_t|z_{0:t})$

- Probability of state given previous and current observations
- **Continuous, represented as Gaussiasn**

Prior:



$p(x_{t+1}|z_{0:t})$

- Probability of state given only previous observations
- **Continuous, represented as Gaussiasn**

Discrete Bayes Filter vs. Kalman Filter

- Discrete Bayes Filter:

1. Make observation

2. Calculate likelihood for every position

3. **Multiply** with last prior and normalize

4. **Convolution** with motion model

5. Go to 1.

- Kalman Filter:

1. Make observation

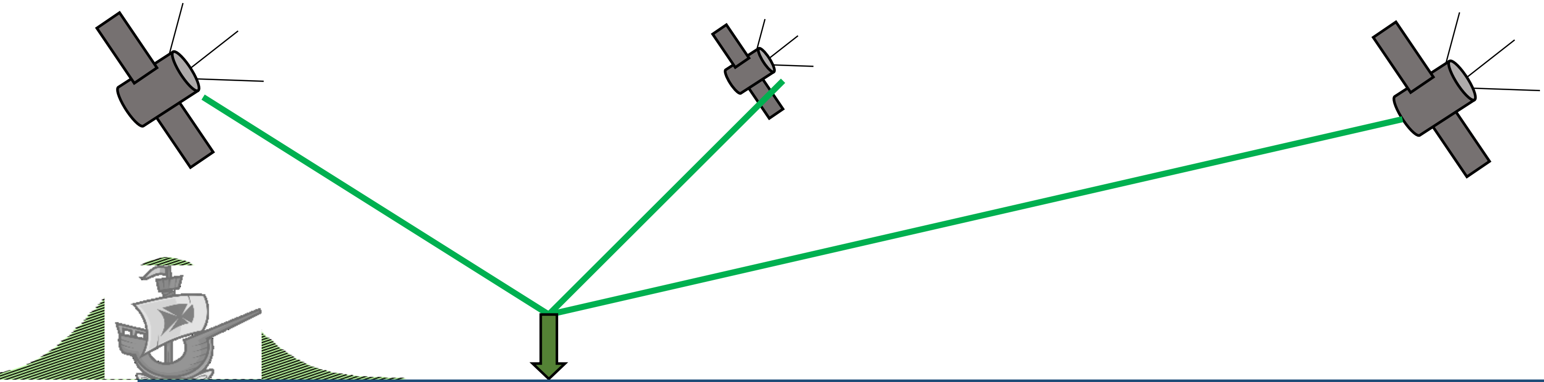
2. Calculate likelihood for every position in **closed form**

3. **Multiply** with last prior and normalize in **closed form**

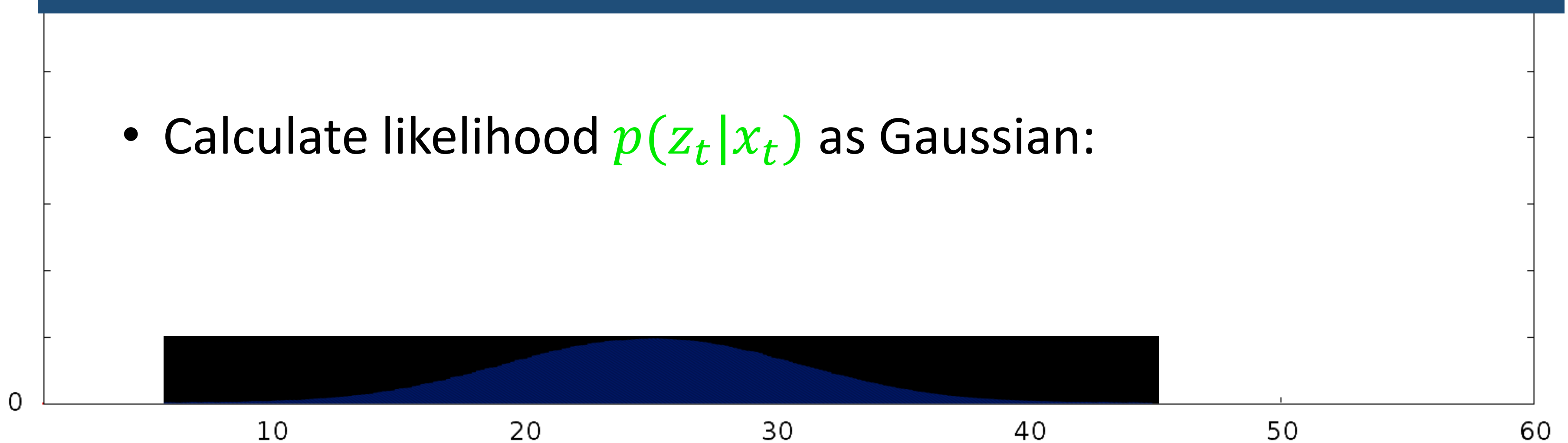
4. **Convolution** with motion model in **closed form**

5. Go to 1.

Kalman Filter



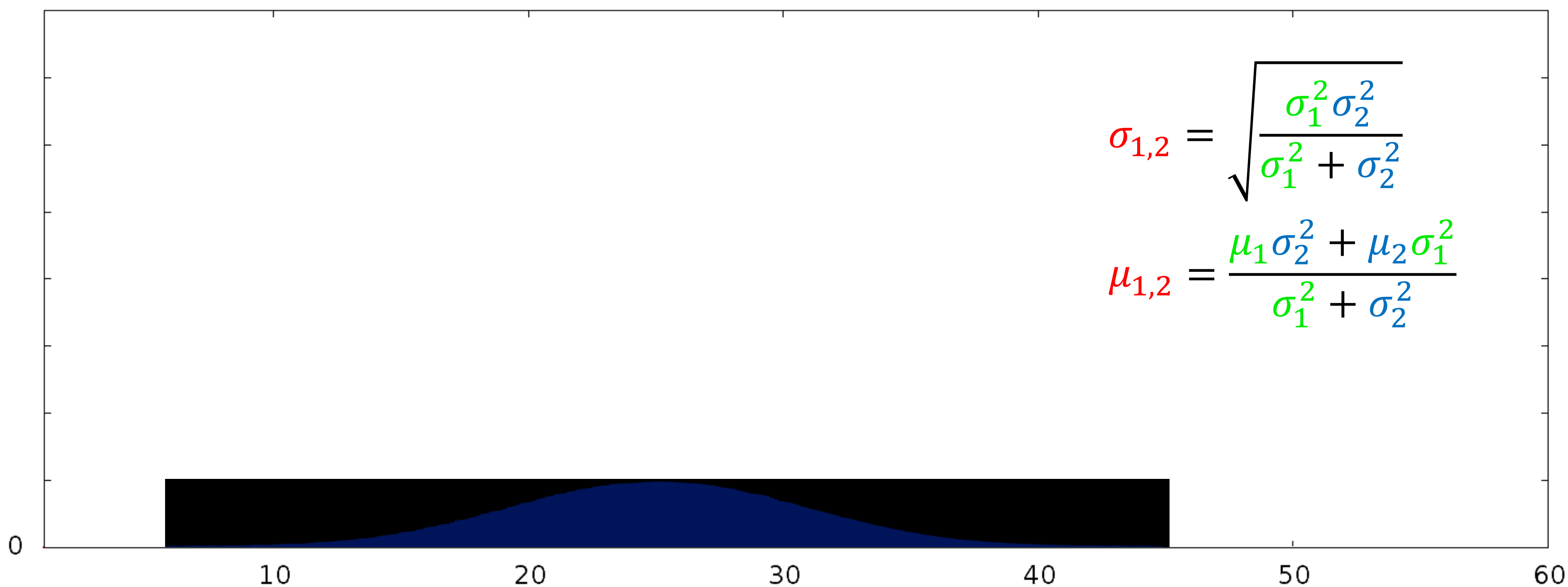
- Calculate likelihood $p(z_t|x_t)$ as Gaussian:



- Calculate the **posterior** by
  - **Multiplying** with prior
  - Normalizing
- Reducing uncertainty

$$\boxed{p(x_t|z_{0:t}) = \frac{p(x_t|z_{0:t-1})p(z_t|x_t)}{\int p(x_t=\hat{x}|z_{0:t-1})p(z_t|x_t=\hat{x})\,d\hat{x}}}$$

- Closed form solution:
  - Another Gaussian



$$\sigma_{1,2} = \sqrt{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}$$

$$\mu_{1,2} = \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}$$

Discrete Bayes Filter vs. Kalman Filter

- Discrete Bayes Filter:

1. Make observation
2. Calculate likelihood for every position
3. **Multiply** with last prior and normalize
4. **Convolution** with motion model
5. Go to 1.

- Kalman Filter:

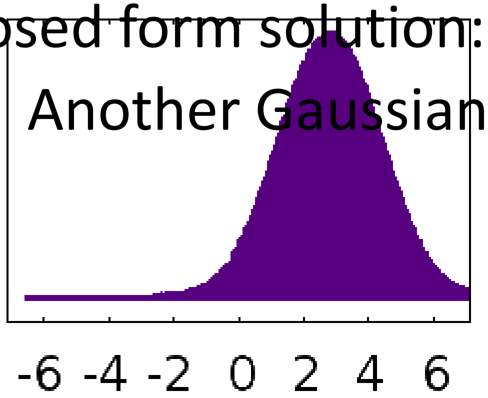
1. Make observation
2. Calculate likelihood for every position in **closed form**
3. **Multiply** with last prior and normalize in **closed form**
4. **Convolution** with motion model in **closed form**
5. Go to 1.

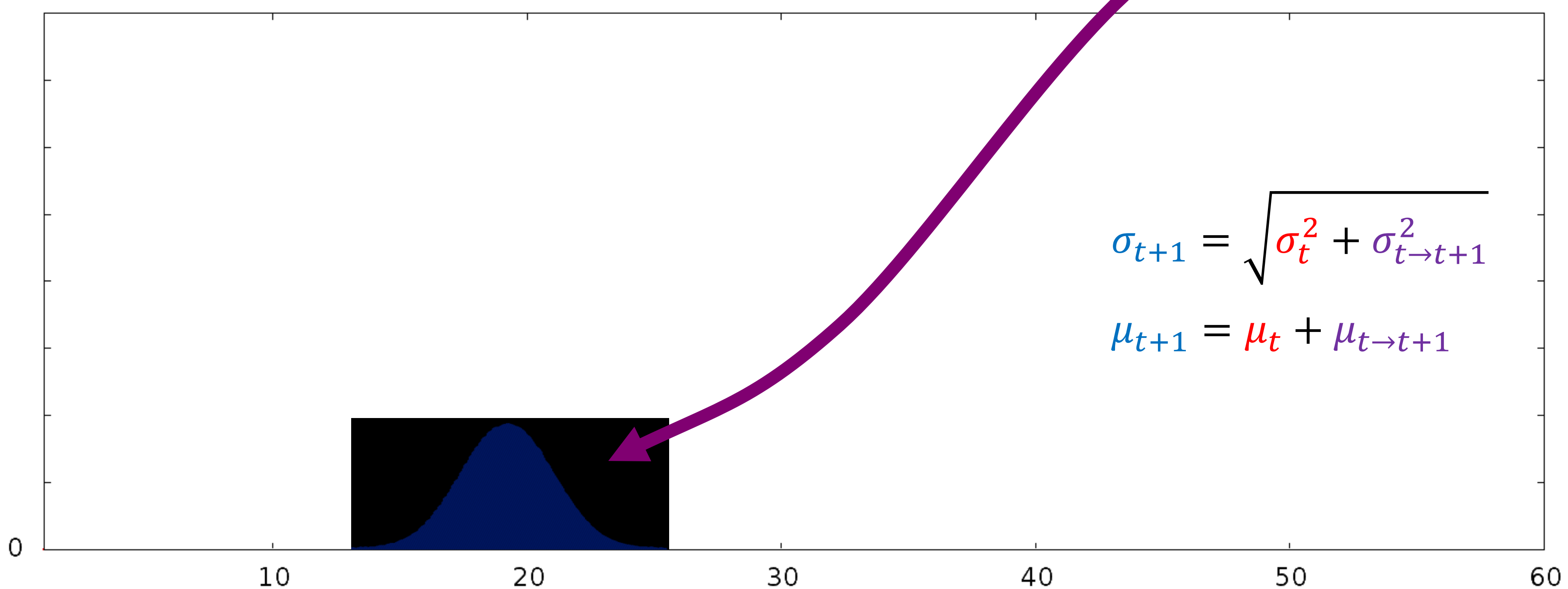
Kalman Filter

- Calculate the **prior** by
  Convolution with **motion model**
- Adding uncertainty

$$p(x_{t+1}|z_{0:t}) = \int p(x_t = \hat{x}|z_{0:t})p(x_{t+1}|x_t = \hat{x})d\hat{x}$$

- Closed form solution:
  - Another Gaussian





$$\sigma_{t+1} = \sqrt{\sigma_t^2 + \sigma_{t\to t+1}^2}$$

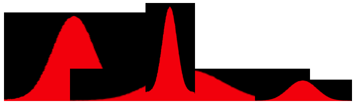
$$\mu_{t+1} = \mu_t + \mu_{t\to t+1}$$

Pros and Cons

Particle Filter:

Observation model can be anything

Motion model can be anything

Multimodal 

Easy to implement and parallelize

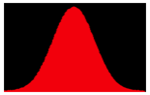
Likelihood calculation per particle can be expensive

Problematic in high dimensional state space (many particles required)

Kalman Filter:

Observation model: linear transformation of state plus Gaussian noise

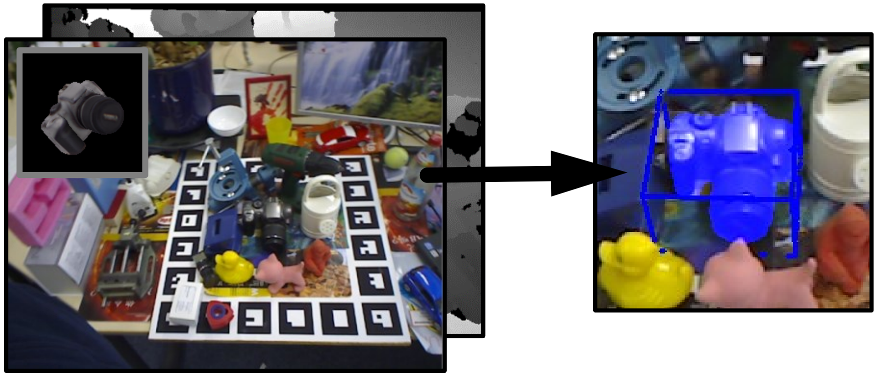
Motion model: linear transformation of last state plus Gaussian noise

Unimodal 

Closed form solution in every step

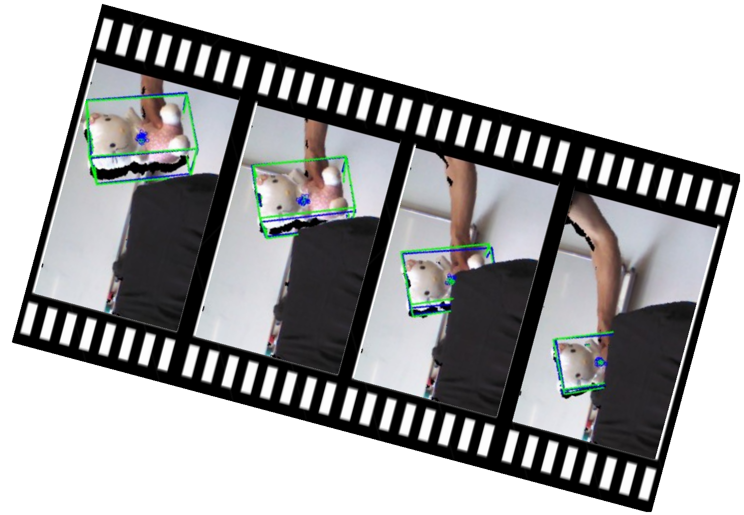
Pose Estimation vs. Pose Tracking

One Shot Pose Estimation [1]



- Estimate 6D Pose from **single** RGB-D image
- Use Object Coordinate Regression

Pose Tracking



- **Stream** of RGB-D images
- Use information from previous frames:
  - Realtime
  - Increase robustness, accuracy

[1] Brachmann, E., Krull, A., Michel, F., Shotton, J., Gumhold, S., Rother, C.: Learning 6d object pose estimation using 3d object coordinates, ECCV (2014)