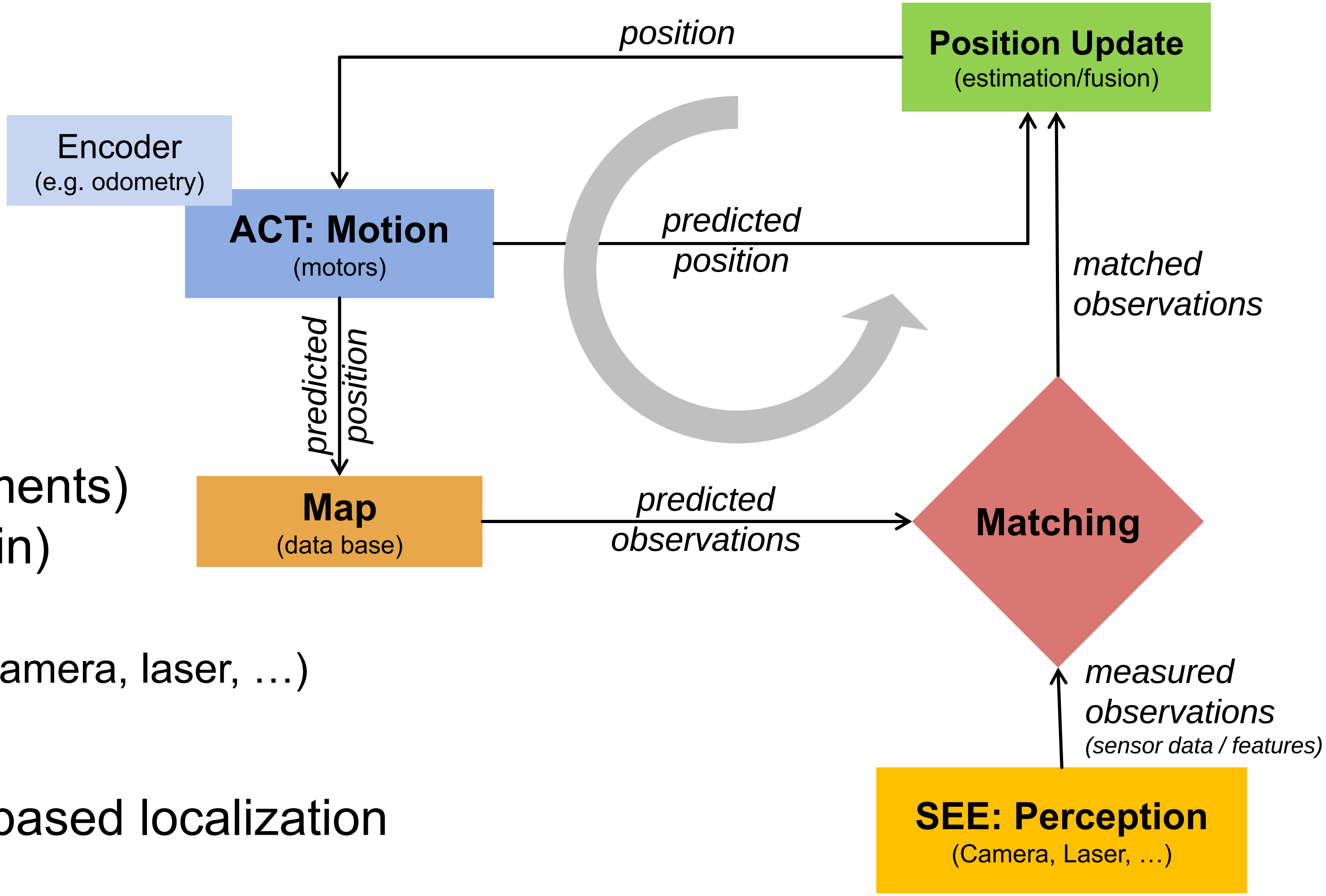
Localization | **the Markov Approach**

Autonomous Mobile Robots

Roland Siegwart

Margarita Chli, Paul Furgale, Marco Hutter, Martin Rufli, Davide Scaramuzza

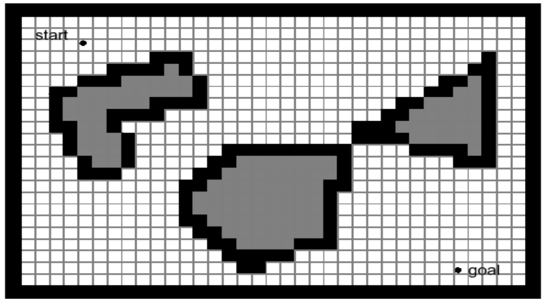
Markov localization | applying probability theory to localization



- Information (measurements) is error prone (uncertain)
  - Odometry
  - Exteroceptive sensors (camera, laser, …)
  - Map

→ Probabilistic map-based localization

Markov localization | basics and assumption



- Discretized pose representation $x_t \rightarrow$ grid map

- Markov localization tracks the robot's belief state $bel(x_t)$ using an arbitrary probability density function to represent the robot's position

- *Markov assumption*: Formally, this means that the output of the estimation process is a function x_t only of the robot's previous state x_{t-1} and its most recent actions (odometry) u_t and perception z_t.

$$p(x_t | x_0, u_t \cdots u_0, z_t \cdots z_0) = p(x_t | x_{t-1}, u_t, z_t)$$

- Markov localization addresses the *global localization problem*, the *position tracking problem*, and the *kidnapped robot problem*.

Markov localization | applying probability theory to localization

- **ACT** | probabilistic estimation of the robot's new belief state $\overline{bel}\,(x_t)$ based on the previous location $bel(x_{t-1})$ and the probabilistic motion model $p(x_t|u_t, x_{t-1})$ with action u_t (control input).

→ application of ***theorem of total probability*** / ***convolution***

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

$$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1}) bel(x_{t-1})$$ for discrete probabilities

Markov localization | applying probability theory to localization

- **SEE** | probabilistic estimation of the robot's new belief state $bel(x_t)$ as a function of its measurement data z_t and its former belief state $\overline{bel}\ (x_t)$:

→ application of ***Bayes rule***

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

where $p(z_t|x_t, M)$ is the probabilistic measurement model (SEE), that is, the probability of observing the measurement data z_t given the knowledge of the map M and the robot's position x_t. Thereby $\eta = p(y)^{-1}$ is the normalization factor so that $\sum p = 1$.

Markov localization | the basic algorithms for Markov localization

For all x_t do

$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1}) bel(x_{t-1})$ (prediction update)

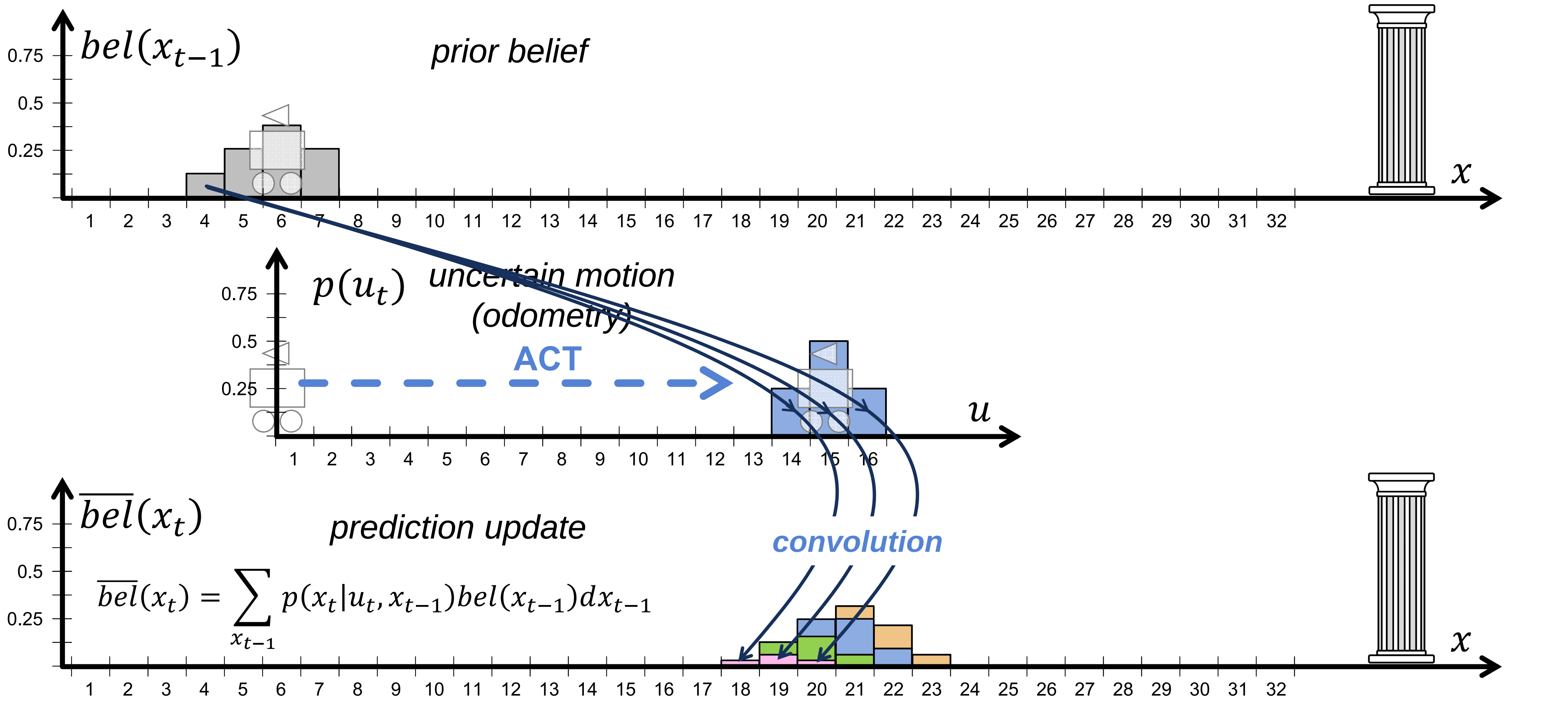
$bel(x_t) = \eta p(z_t|x_t, M)\overline{bel}(x_t)$ (measurement update)

endfor

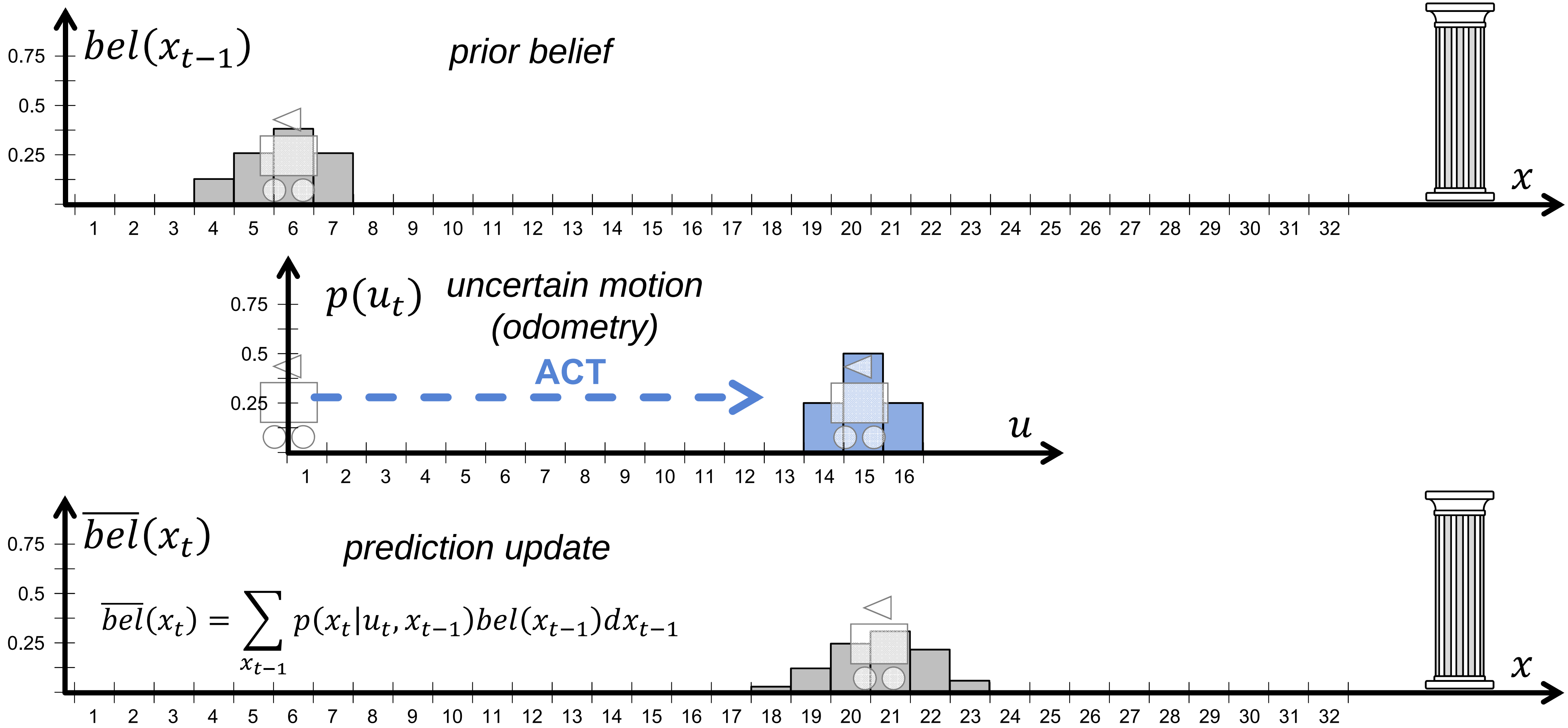
Return $bel(x_t)$

- ***Markov assumption***: Formally, this means that the output is a function x_t only of the robot's previous state x_t and its most recent actions (odometry) u_t and perception z_t.

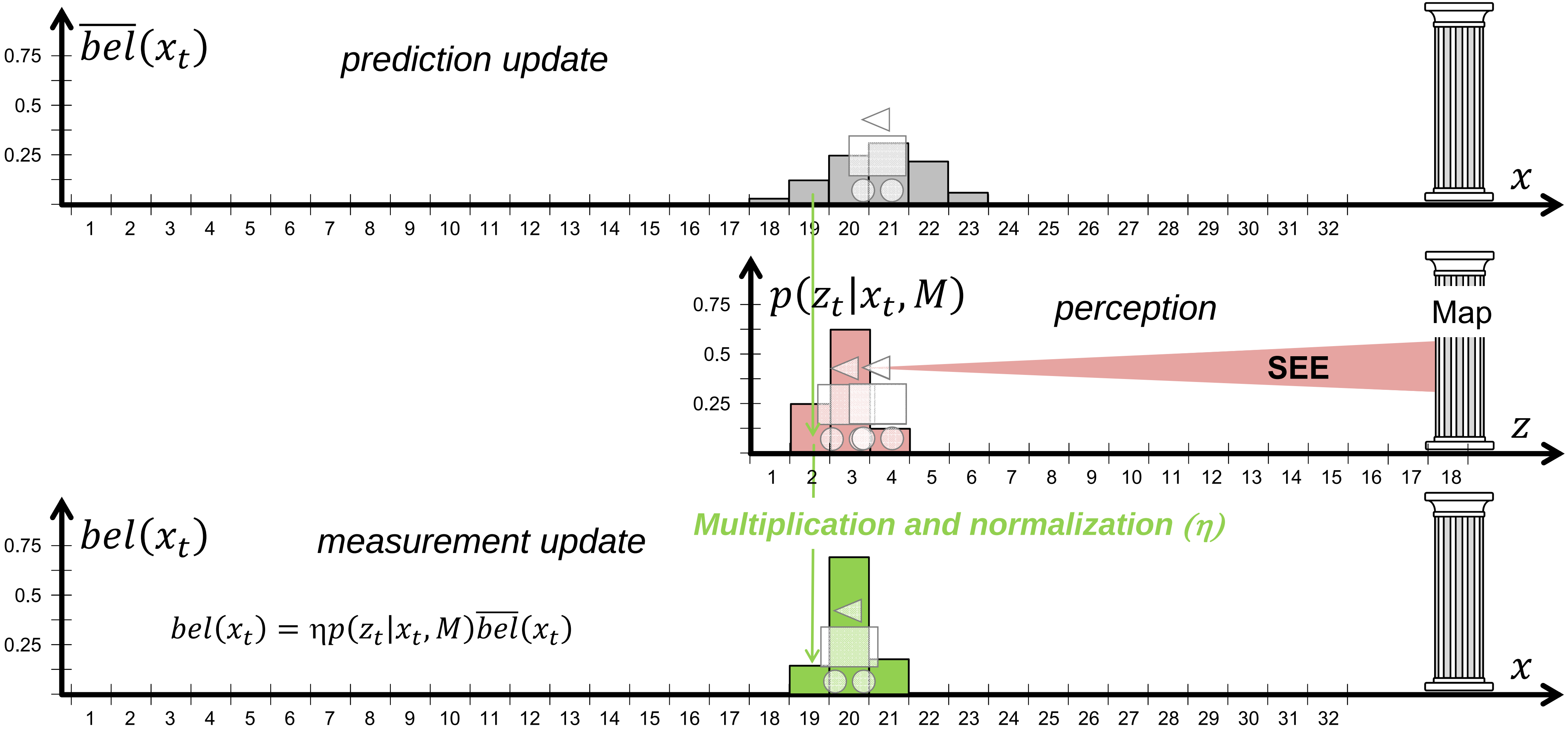
ACT | using motion model and its uncertainties



ACT | using motion model and its uncertainties

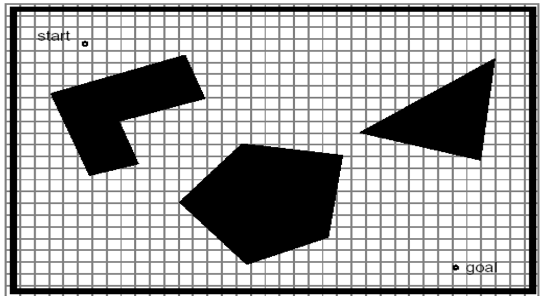


SEE | estimation of position based on perception and map

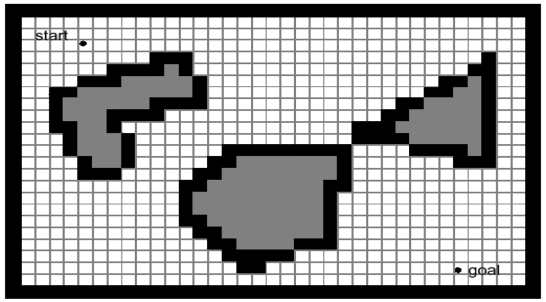


Markov localization | extension to 2D

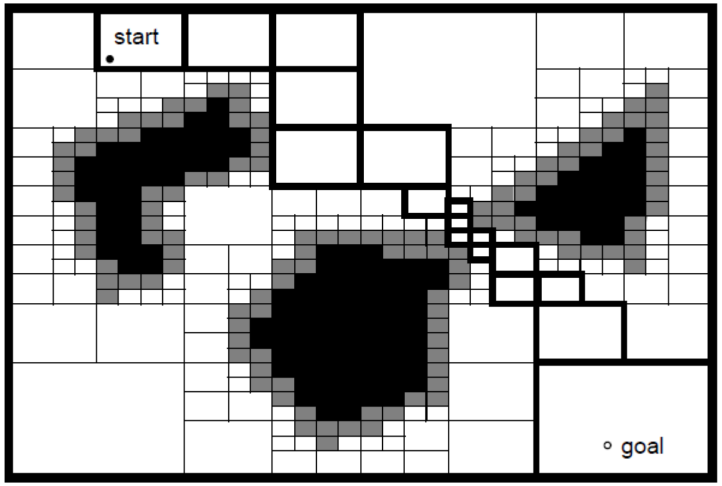
- The real world for mobile robot is at least 2D (moving in the plane)
  - → discretized pose state space (grid) consists of x, y, θ
  - → Markov Localization scales badly with the size of the environment
- Space: 10 m x 10 m with a grid size of 0.1 m and an angular resolution of 1°
  - → $100 \cdot 100 \cdot 360 = 3.6\ 10^6$ grid points (states)
  - → prediction step requires in worst case $(3.6\ 10^6)^2$ multiplications and summations



- Fine fixed decomposition grids result in a huge state space
  - Very important processing power needed
  - Large memory requirement



Markov localization | reducing computational complexity



- Adaptive cell decomposition
- Motion model (Odomety) limited to a small number of grid points
- Randomized sampling
  - Approximation of belief state by a representative subset of possible locations
  - weighting the sampling process with the probability values
  - Injection of some randomized (not weighted) samples

  - randomized sampling methods are also known as particle filter algorithms, condensation algorithms, and Monte Carlo algorithms.