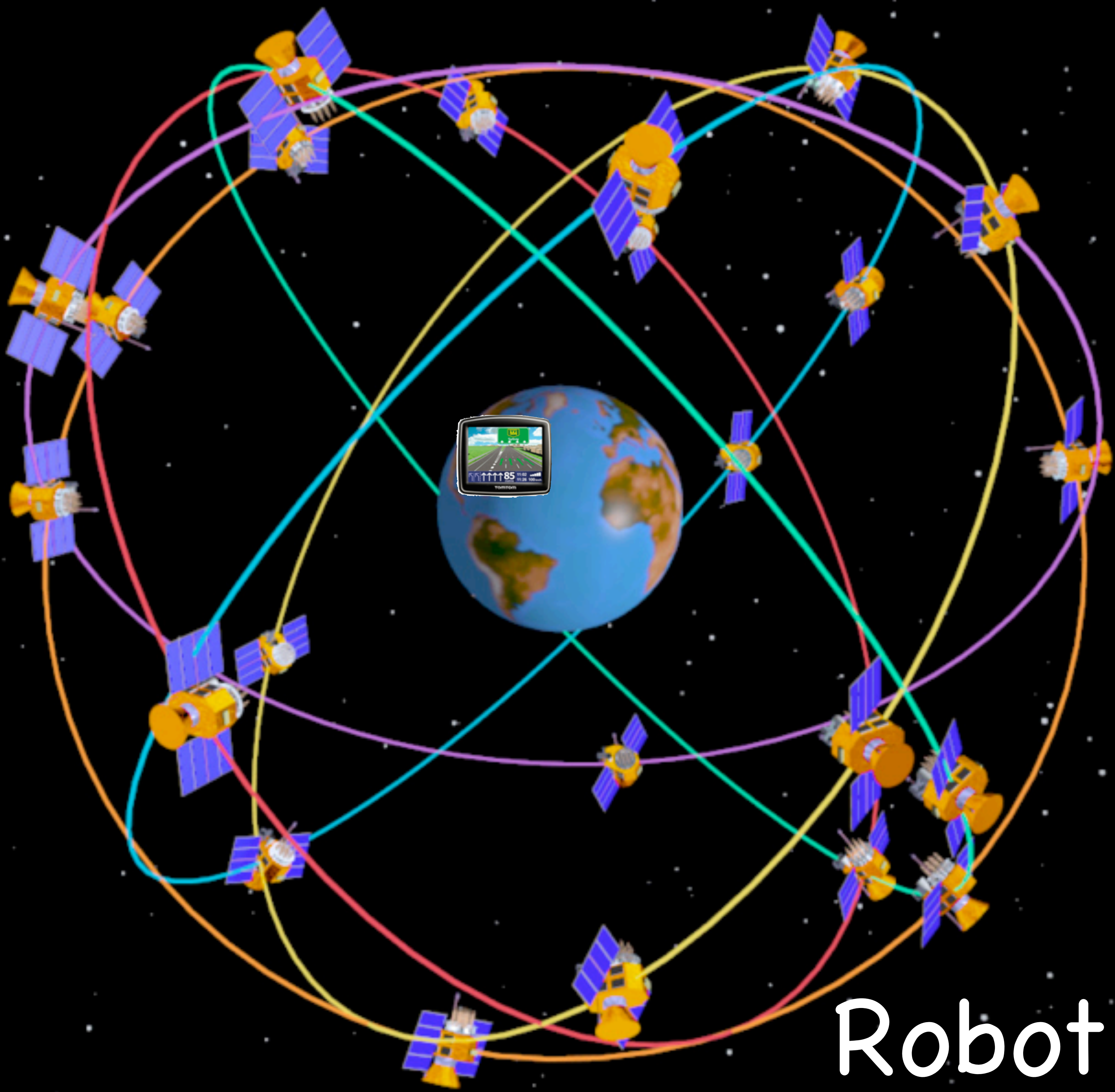


Topic 11

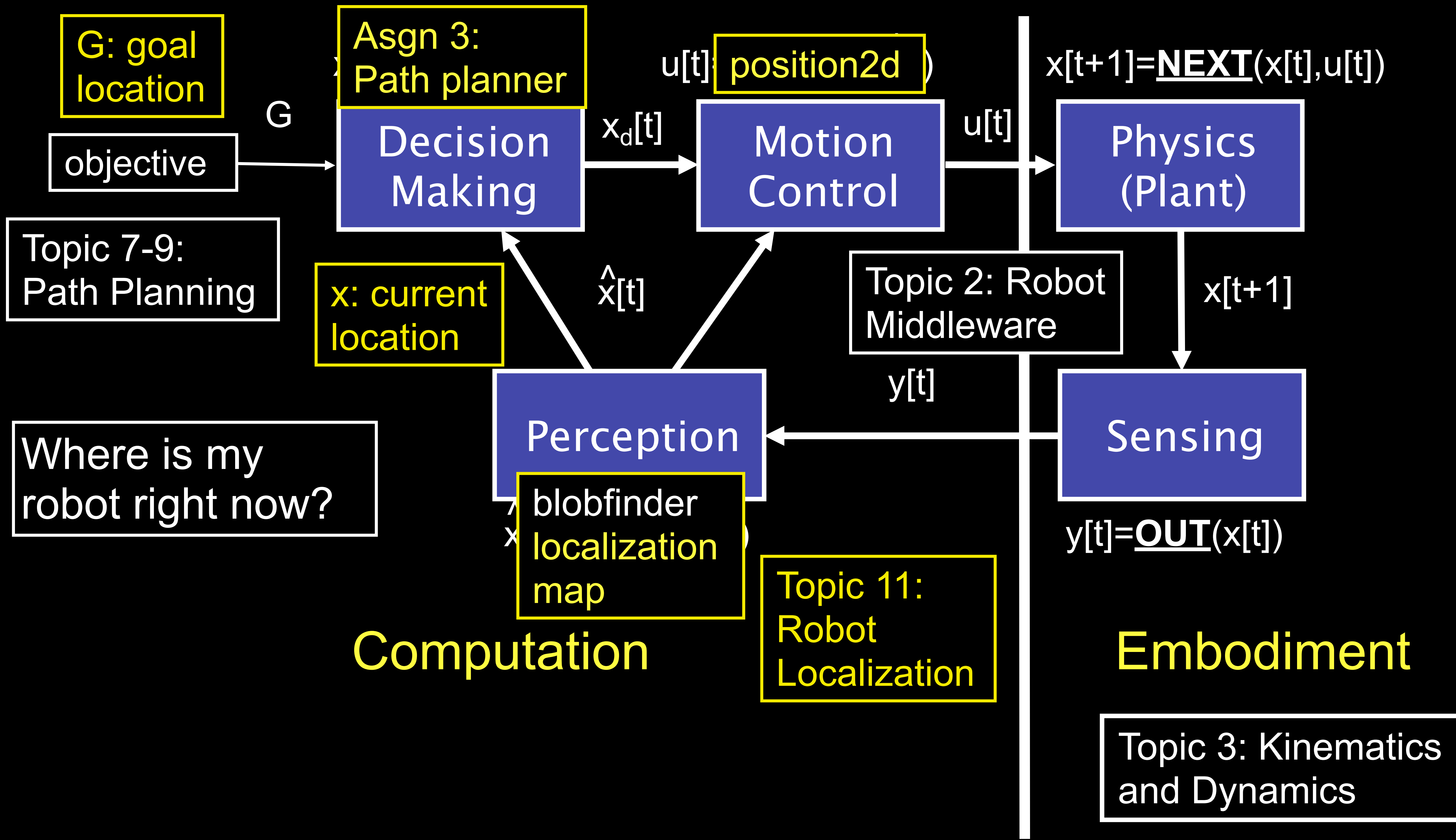
Robot Localization:

Where am I?

robot control loop

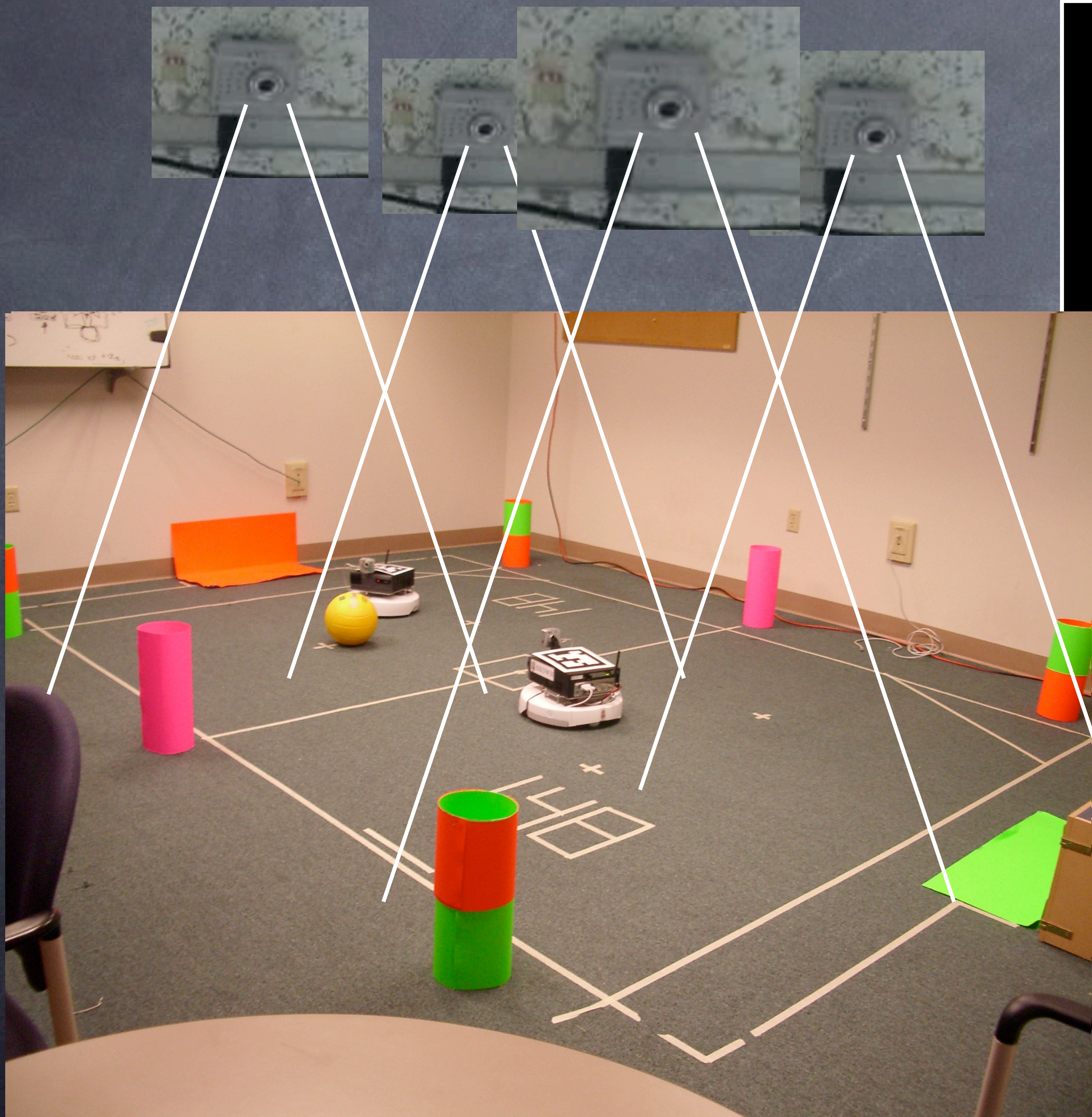
- someone please sketch on the board

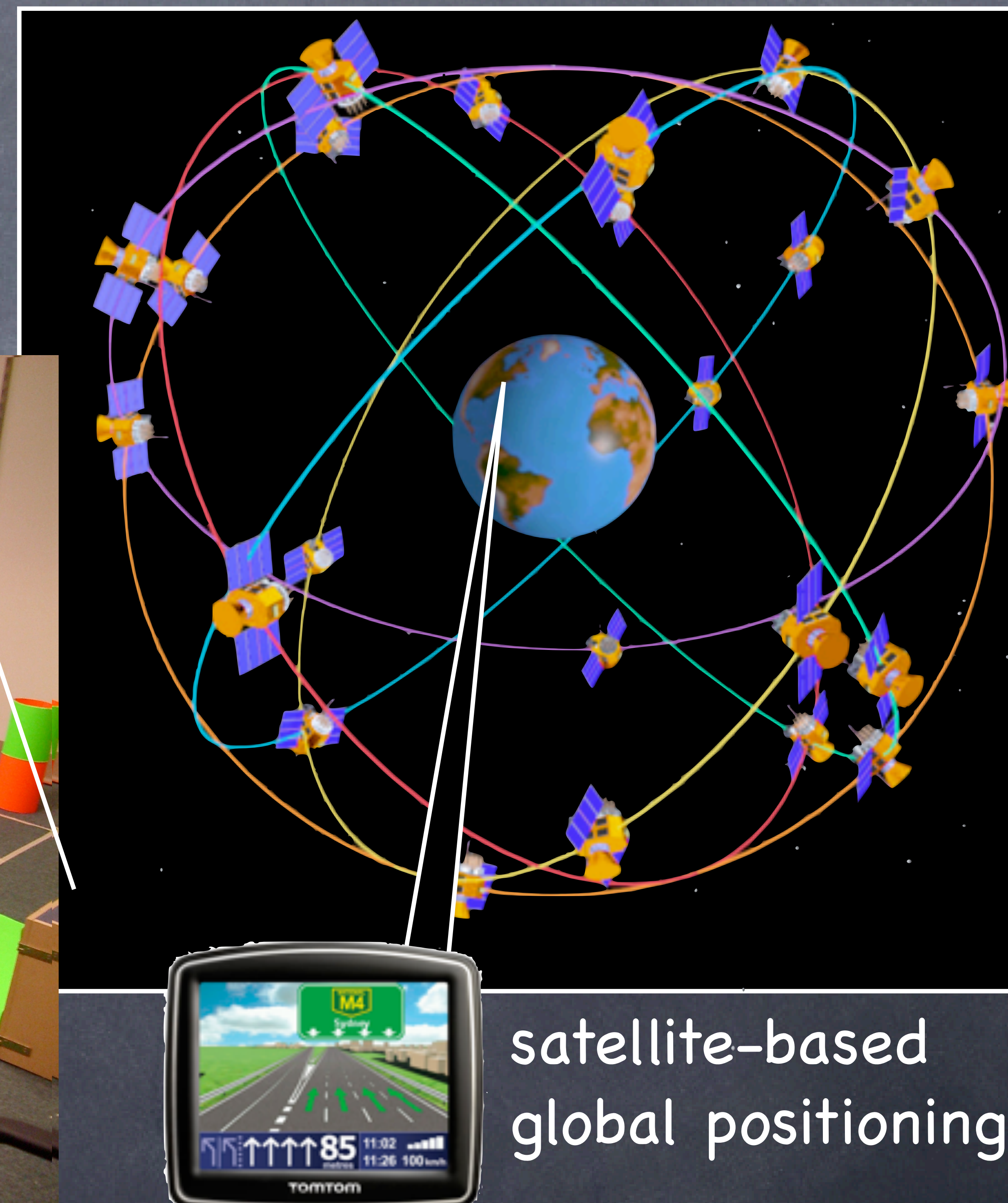
The Robot Control Loop



Global Localization

cs148 overhead system is like a mini GPS

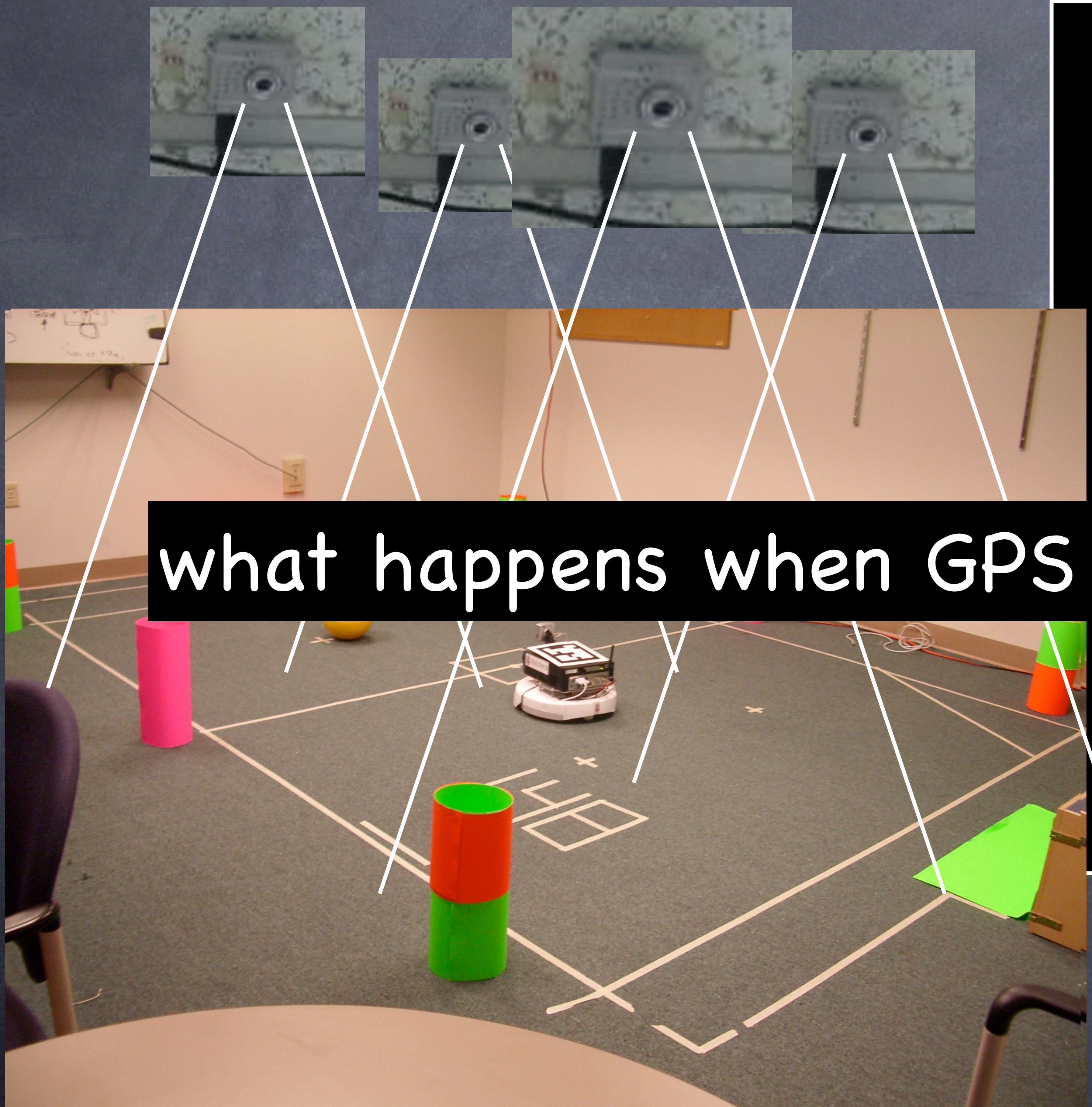


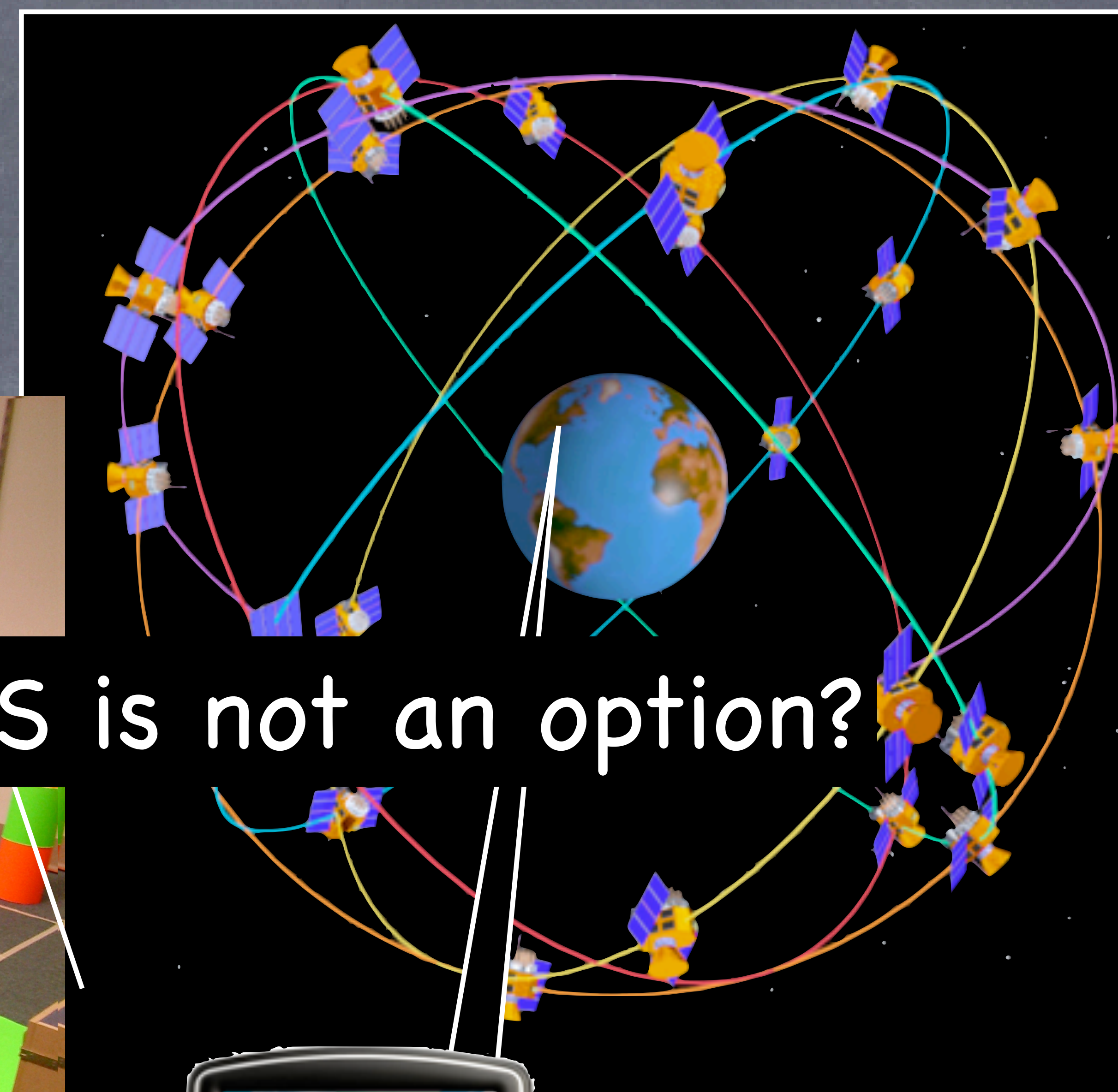


satellite-based global positioning

Global Localization

cs148 overhead system is like a mini GPS

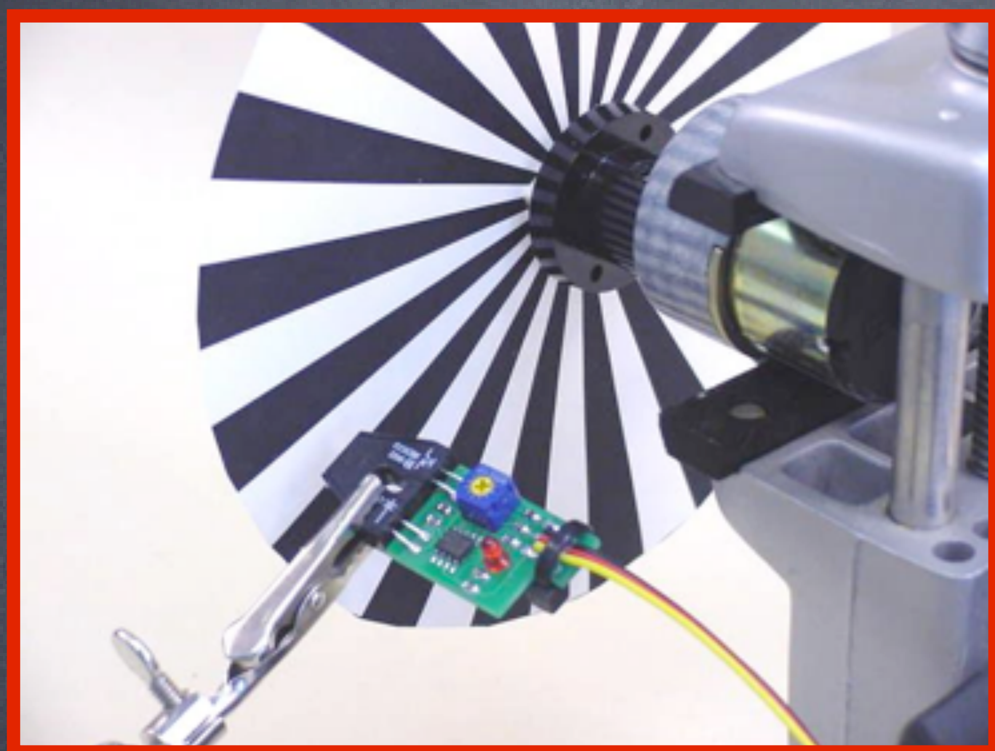


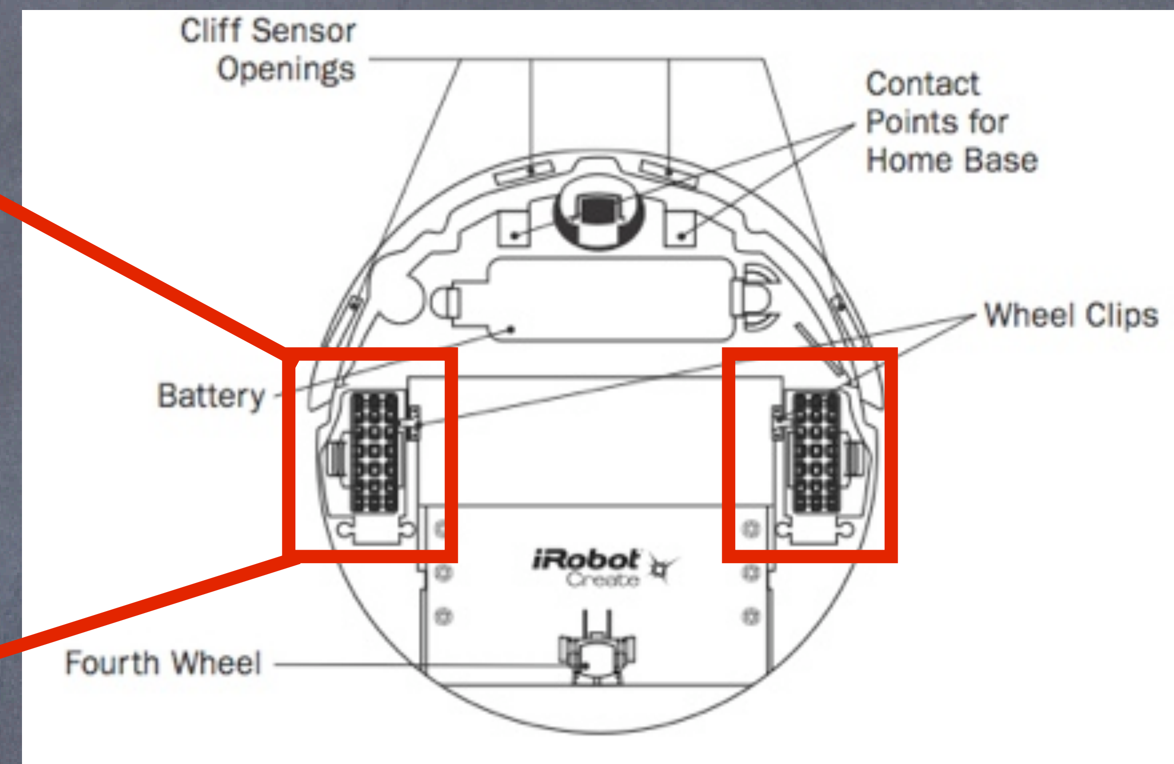


what happens when GPS is not an option?

satellite-based global positioning

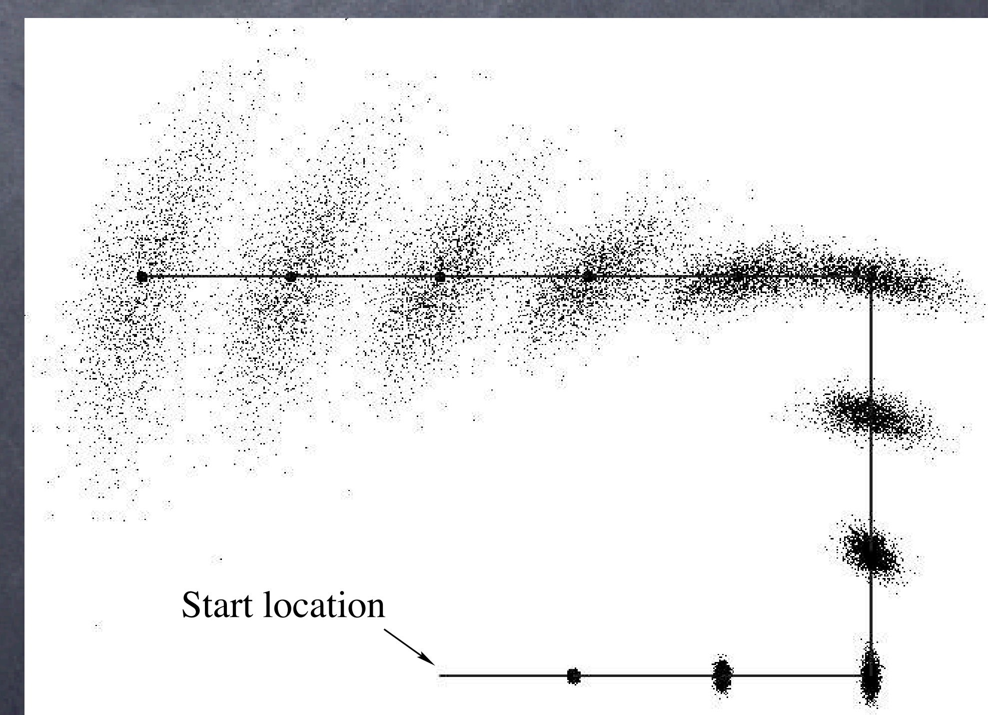
One option: Odometry





Transform wheel rotation (measured by encoders) to motion of robot

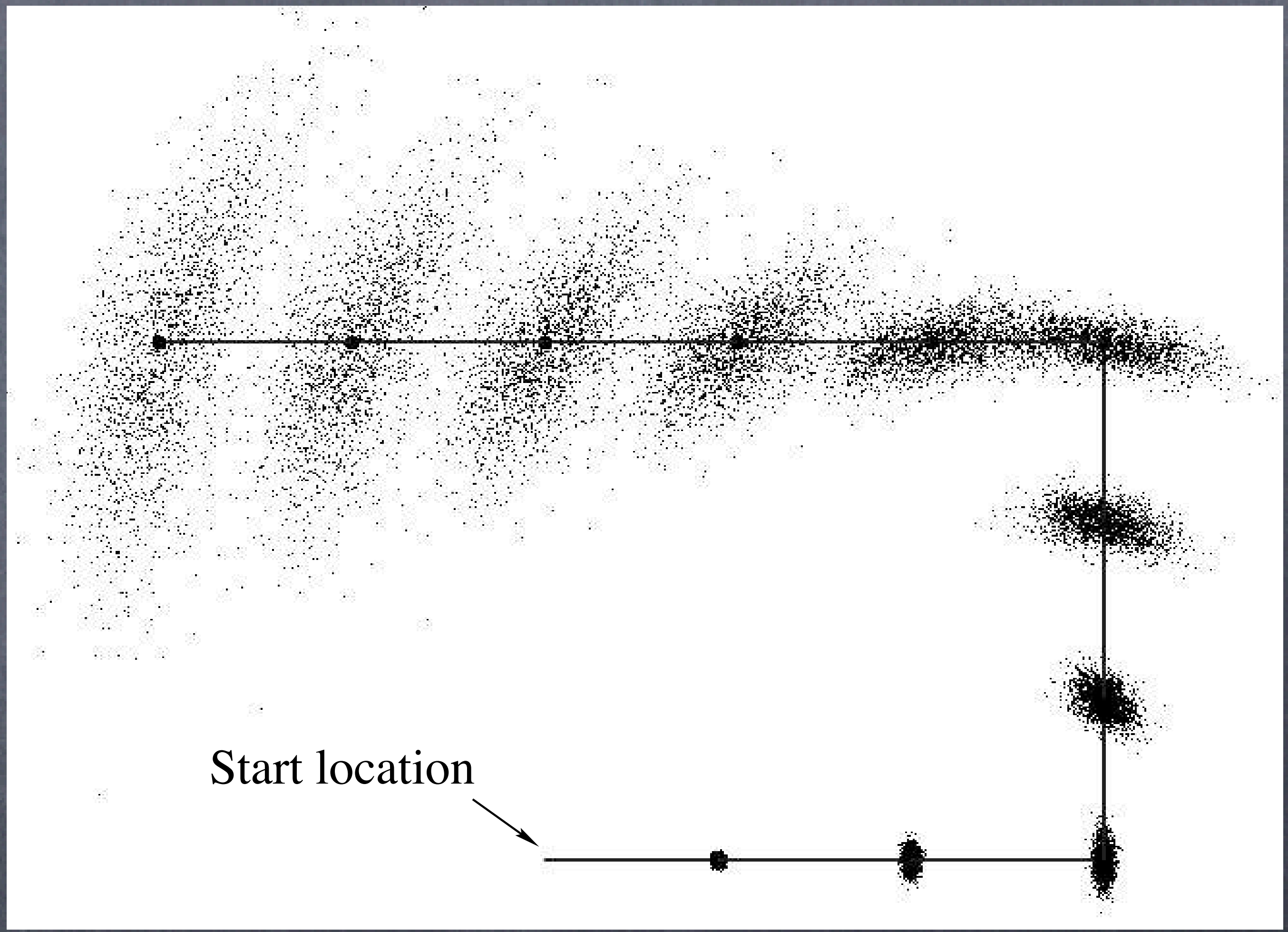
Problems: slip, drift



Problems with Odometry

(for example)

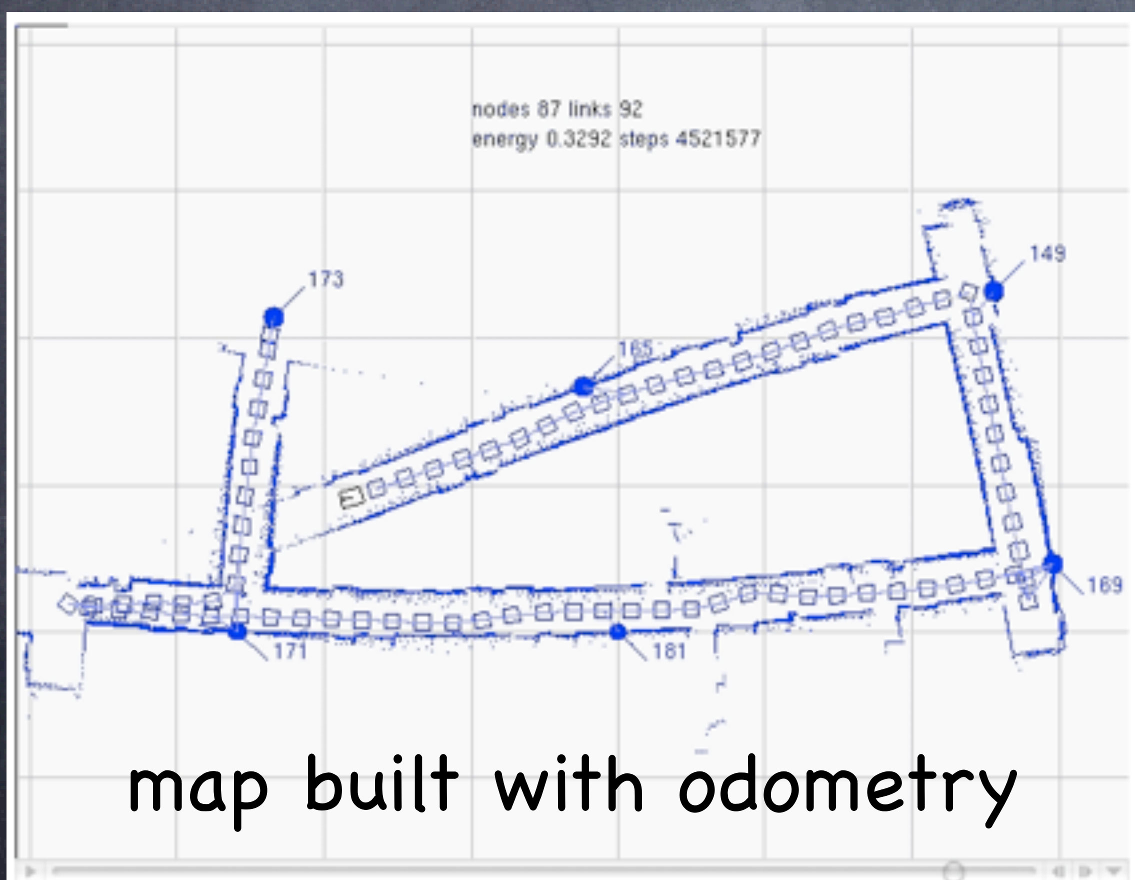
- Slipping: carpet, slippery surfaces, collisions
- Drift: small odometry errors accumulate
- Calibration: finding specific values encoder-to-movement transforms



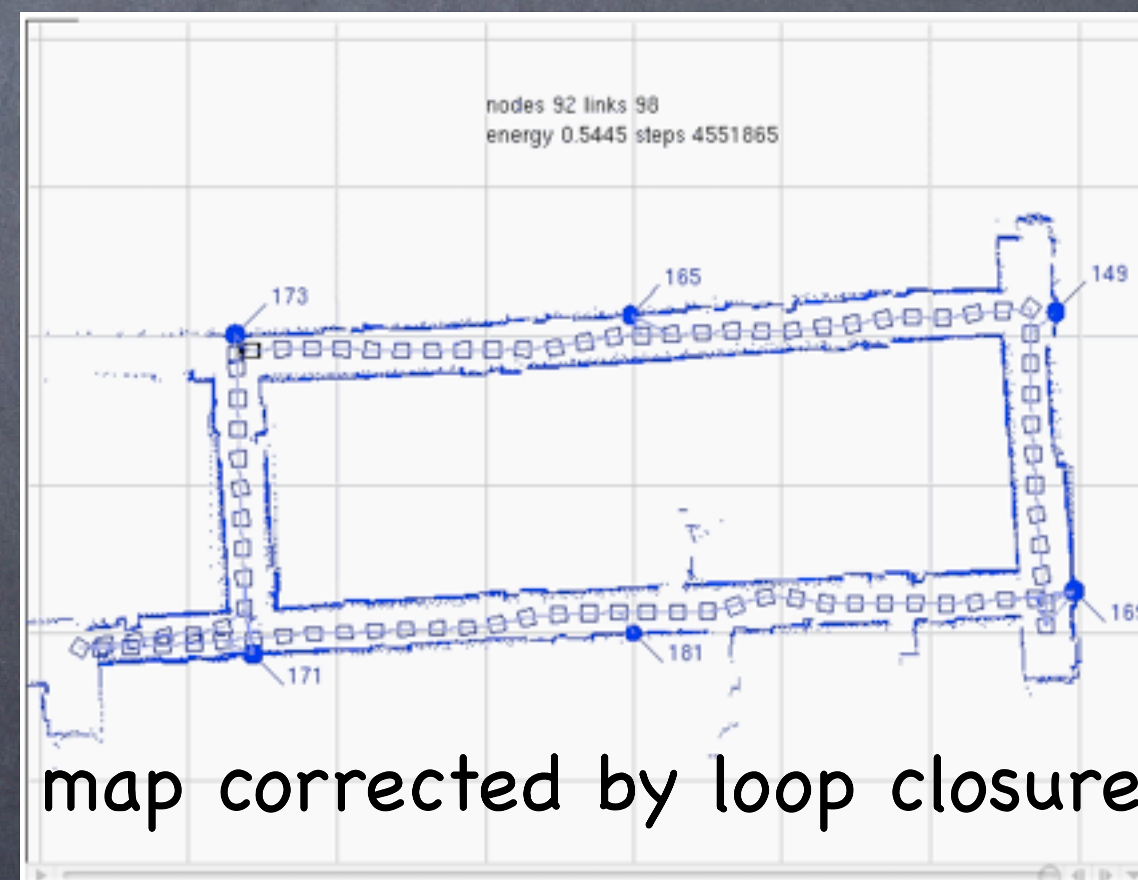
Example: SLAM

(Simultaneous Localization and Mapping)

- Kidnapped robot problem
  - Build map and determine position in map
  - Loop closures (remember previous locations)



map built with odometry



map corrected by loop closure

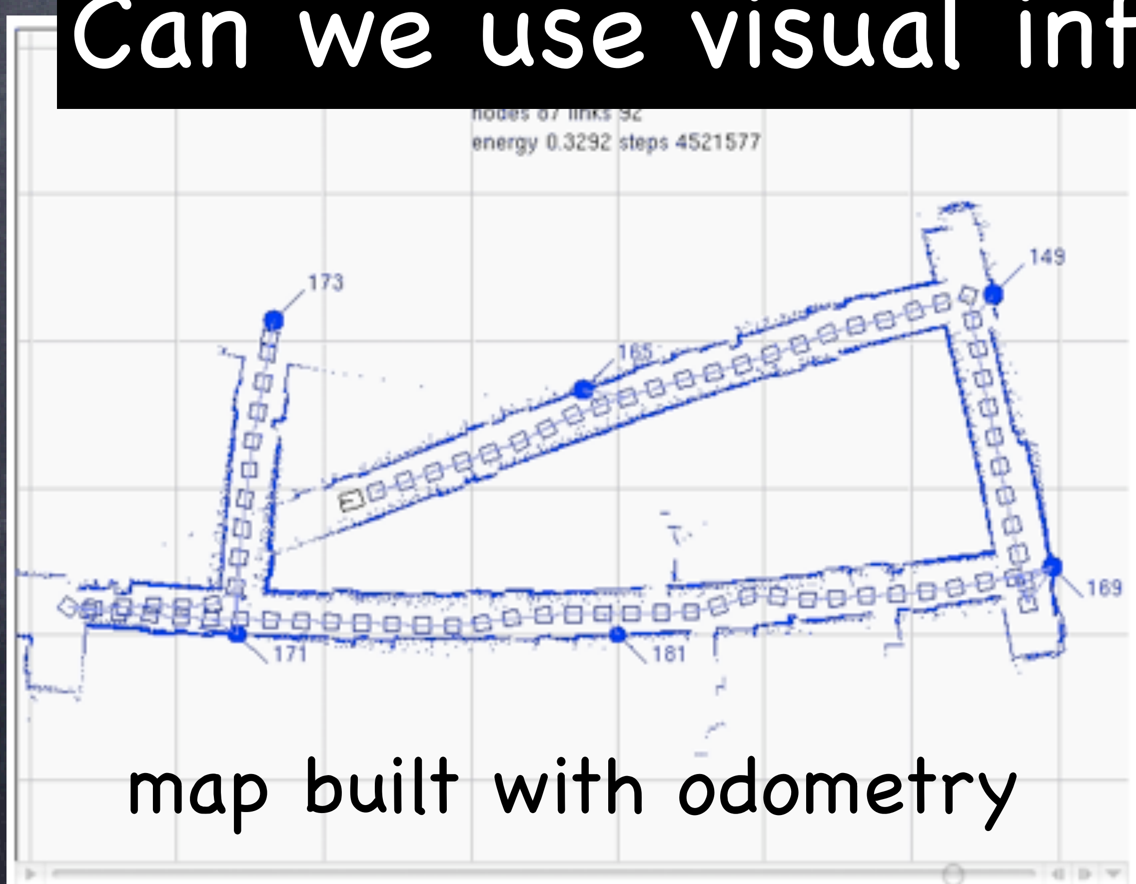
- http://robotics.usc.edu/~ahoward/movies/iros2001a_slam.mov

Example: SLAM

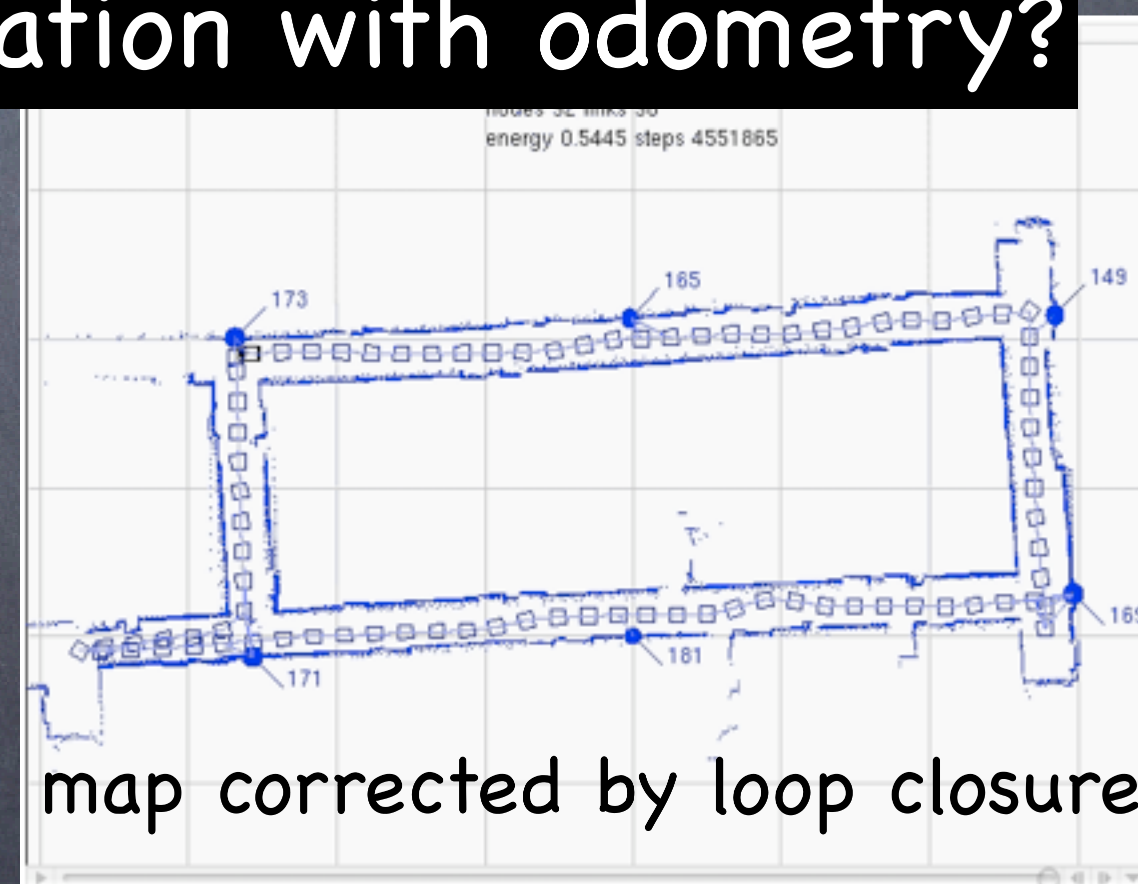
(Simultaneous Localization and Mapping)

- Kidnapped robot problem
  - Build map and determine position in map
  - Loop closures (remember previous locations)

Can we use visual information with odometry?



map built with odometry



map corrected by loop closure

- http://robotics.usc.edu/~ahoward/movies/iros2001a_slam.mov

Onboard Localization

use onboard visual evidence to determine location

assuming a given map

where is this location in the world?

Onboard Localization

use onboard visual evidence
to determine location

where is this location in the world?

Onboard Localizatio

use onboard visual evidence
to determine location

where is this location in the world?

Onboard Localization

use onboard visual evidence
to determine location

where is this location in the world?

Onboard Localizati

use onboard visual evidence
to determine location

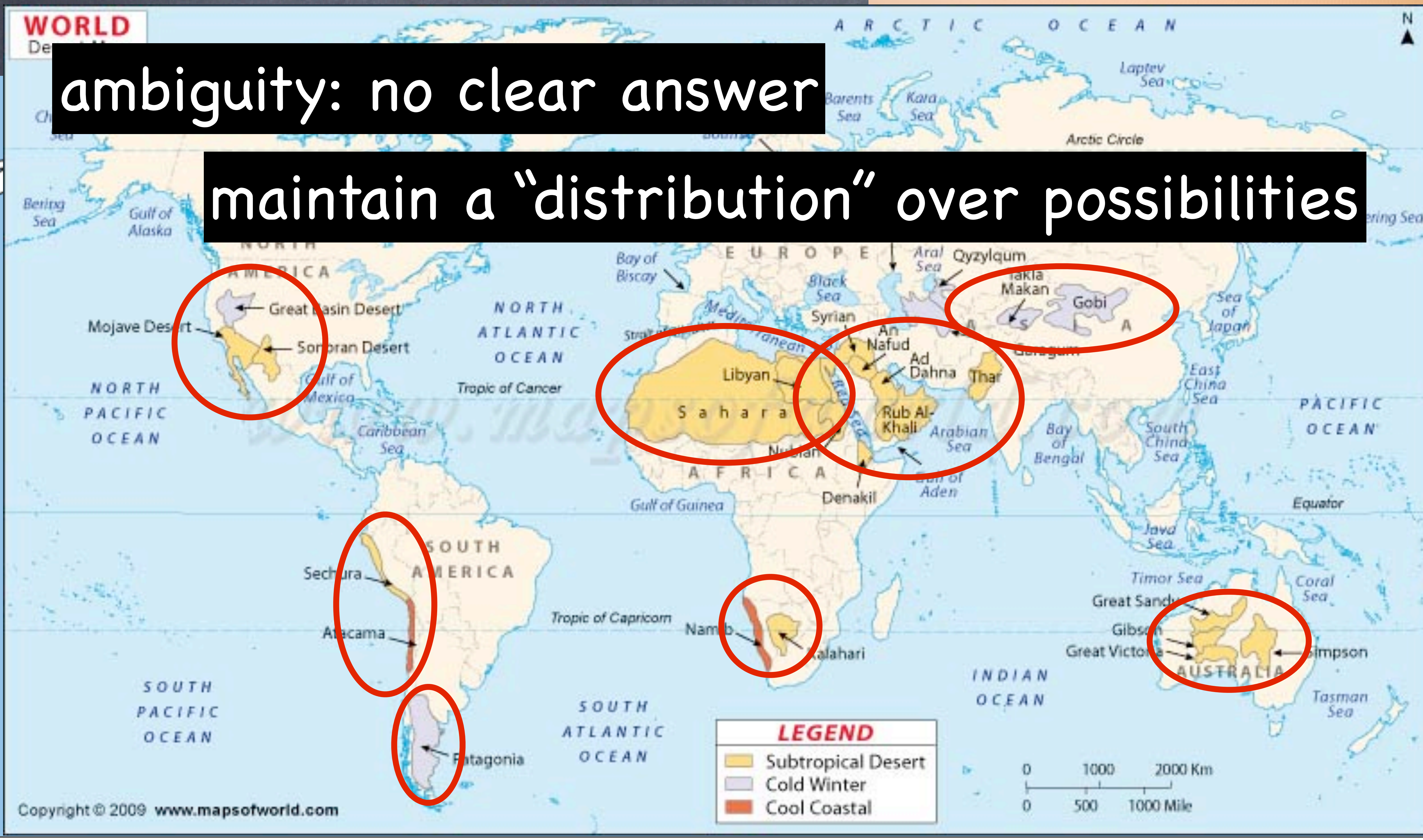
where is this location in the world?

Onboard Localizati

use onboard visual evidence to determine location

Onboard Localizati

use onboard visual evidence
to determine location



ambiguity: no clear answer

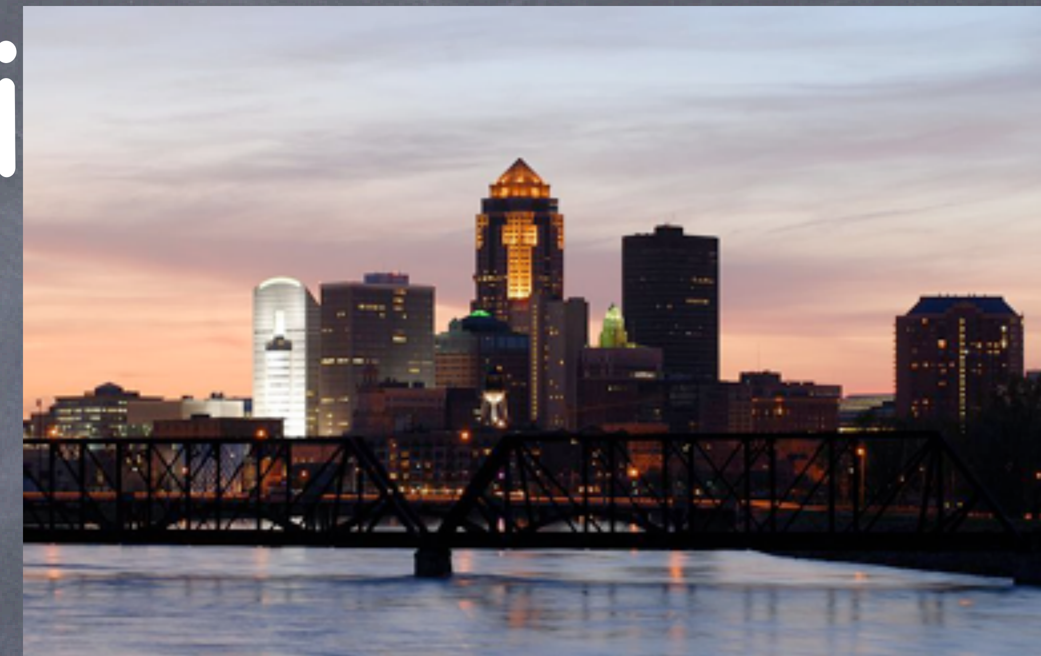
maintain a "distribution" over possibilities

Onboard Localizati

use onboard visual evidence
to determine location

where is this location in the world?

Onboard Localizati



use onboard visual evidence
to determine location

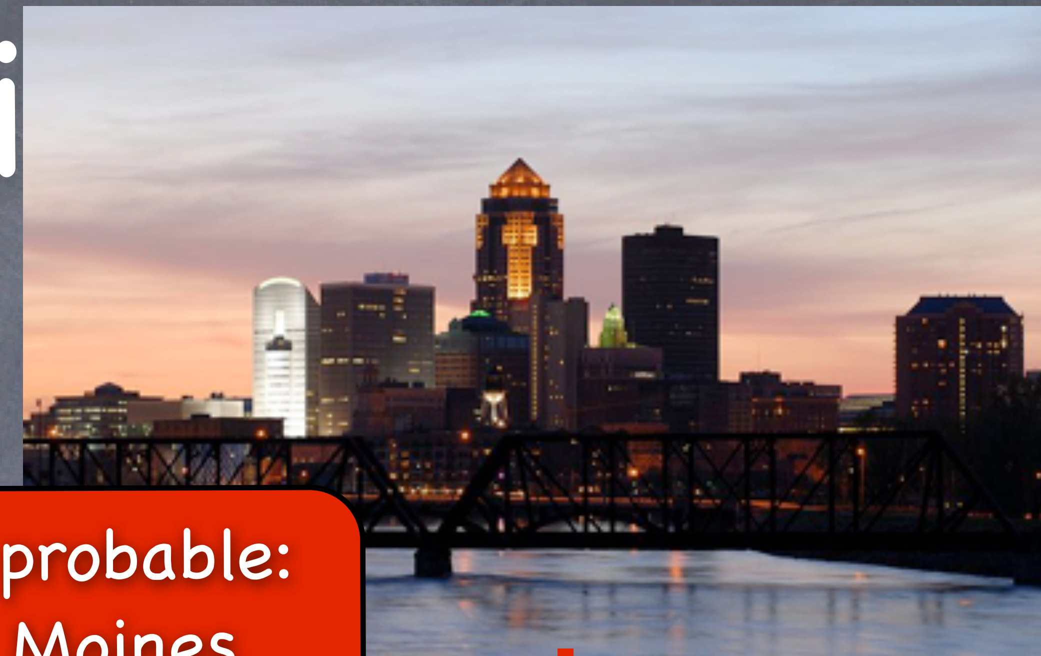
Onboard Localizati

use onboard visual evidence and odometry to determine location

where is this location in the world?

Onboard Localizati

use onboard visual evidence and odometry to determine location

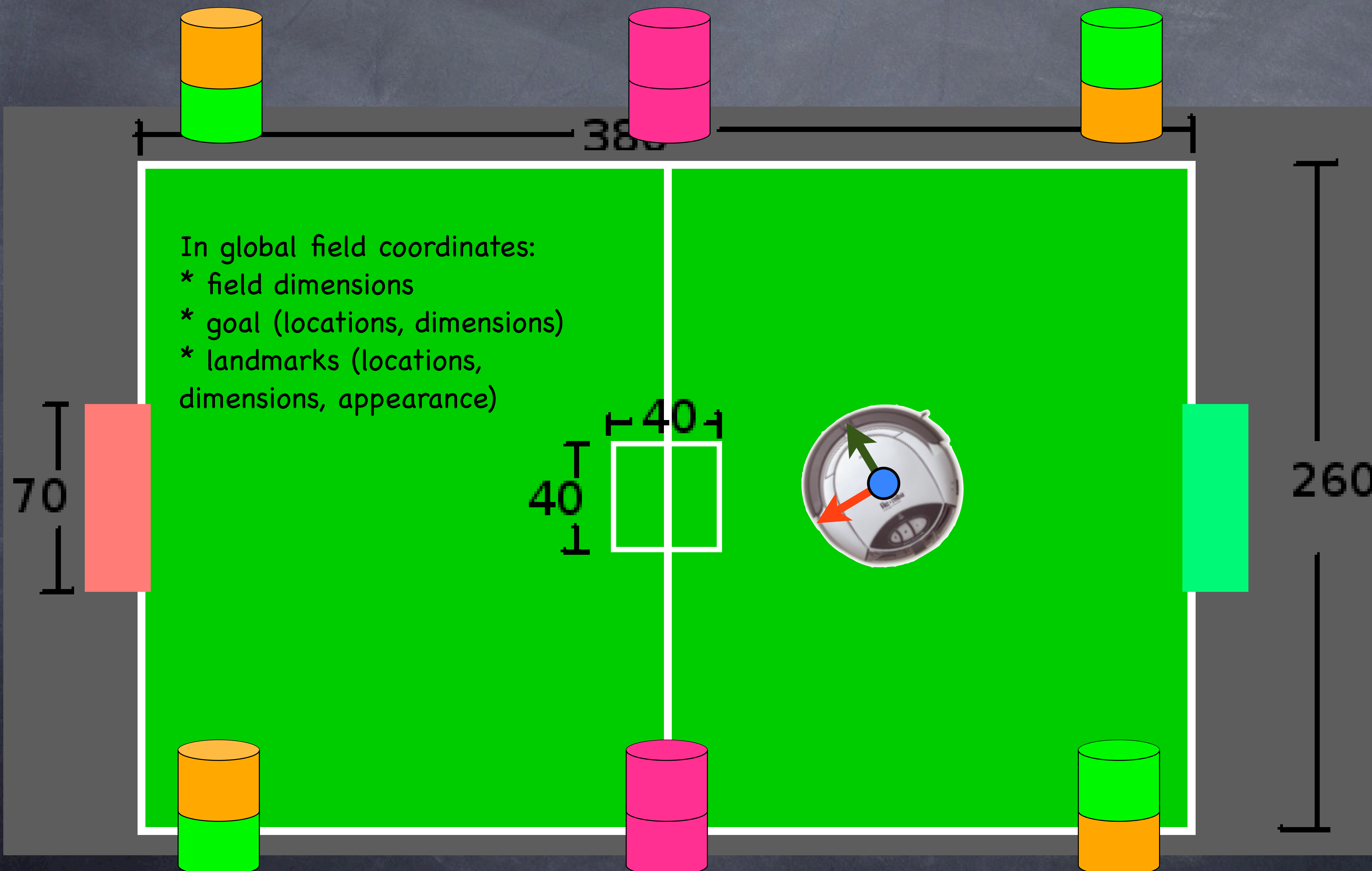


Most probable:
Des Moines

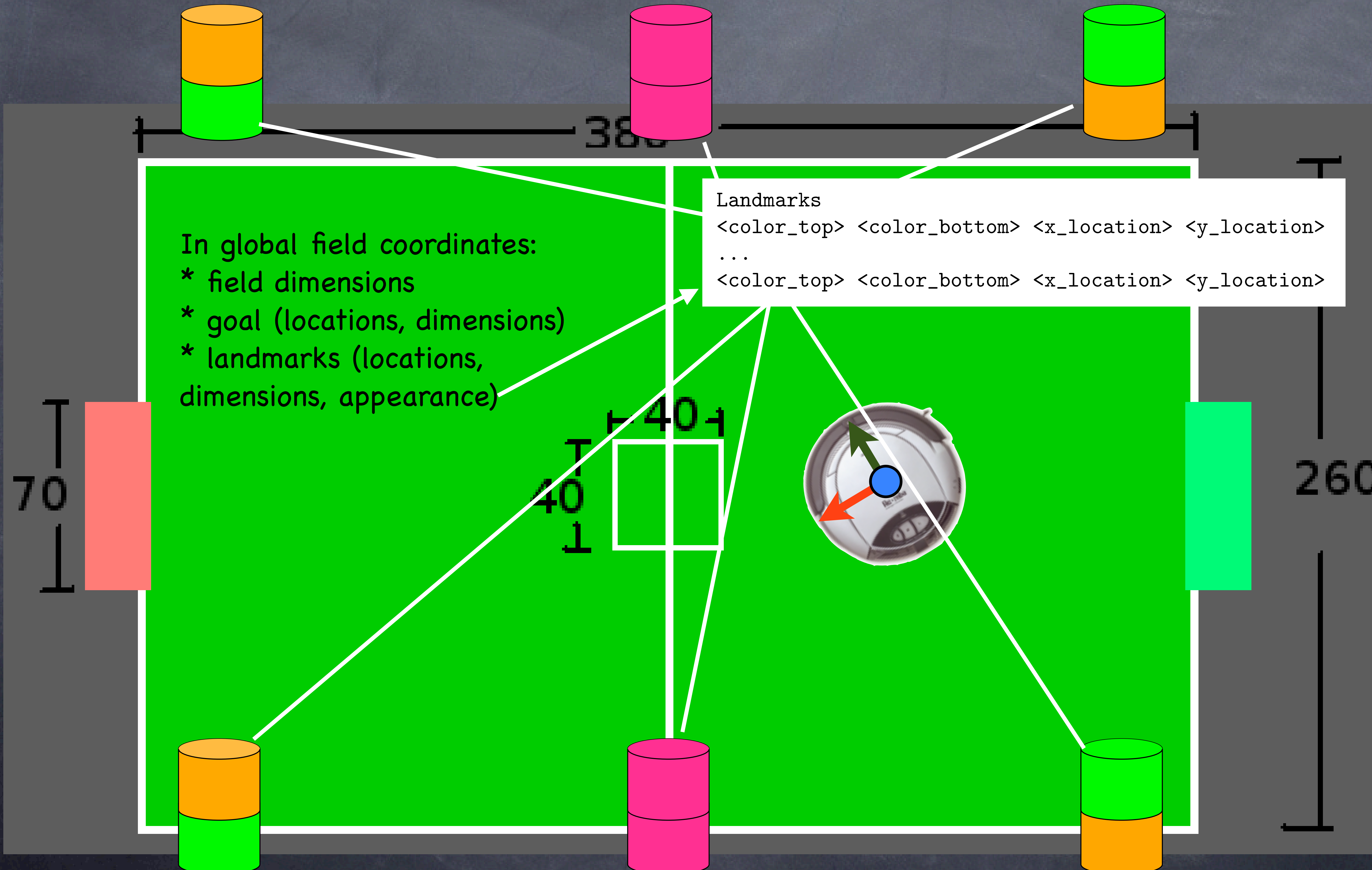
travel 20 km east

where is this location in the world?

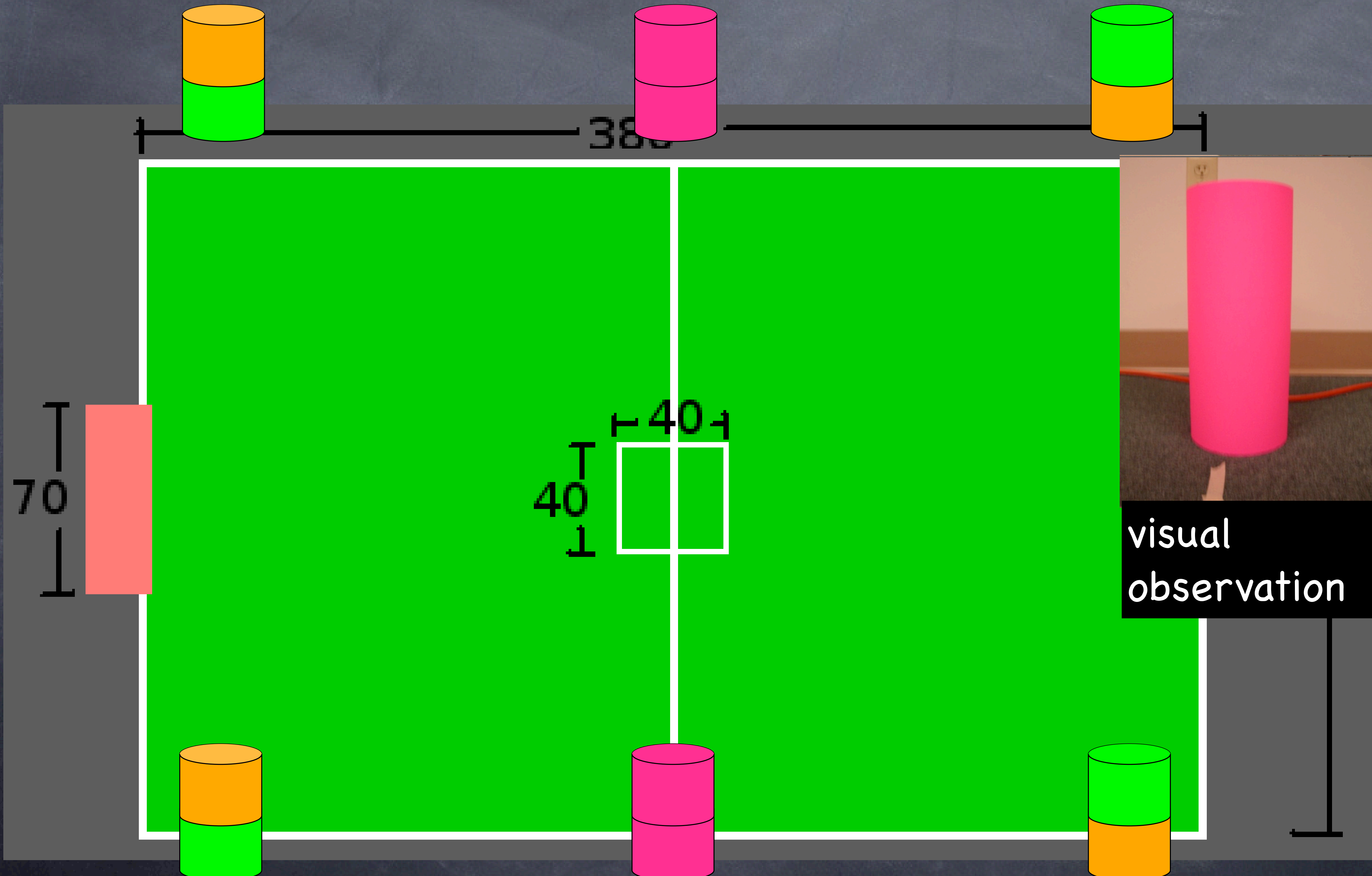
for our robots, assume a map of the soccer field



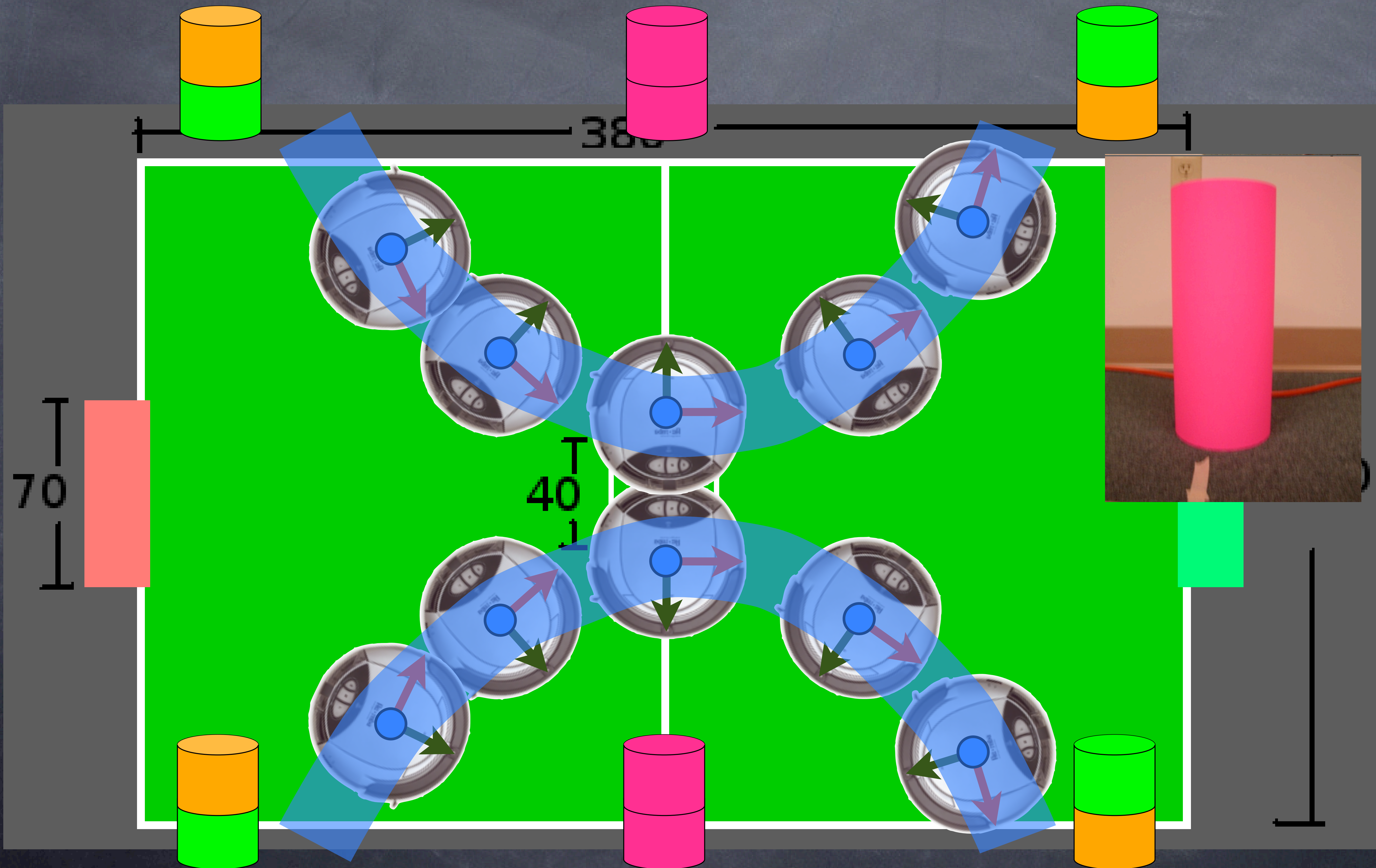
for our robots, assume a map of the soccer field



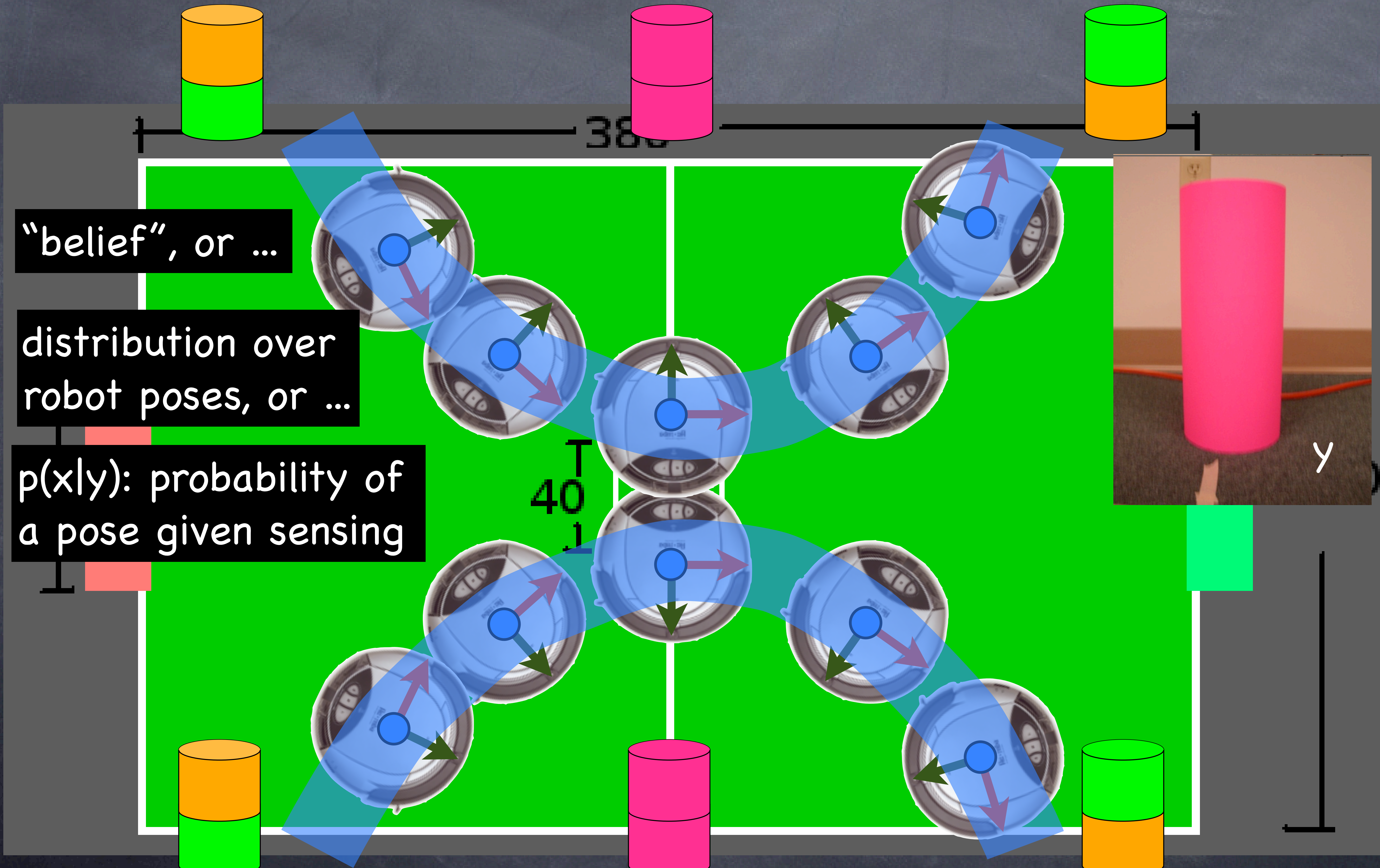
if the robot sees this landmark, where could it be?



if the robot sees this landmark, where could it be?



if the robot sees this landmark, where could it be?

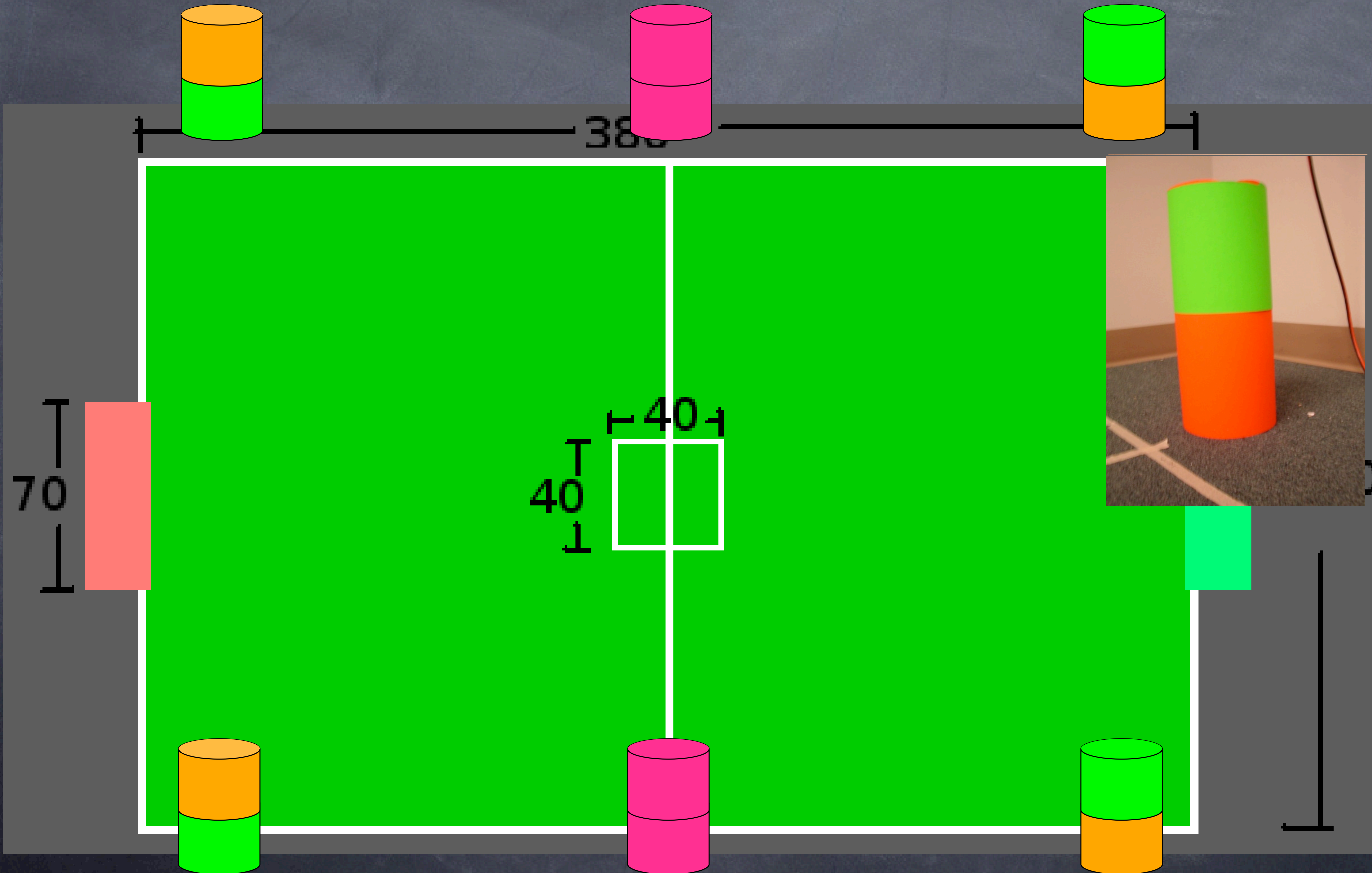


"belief", or ...

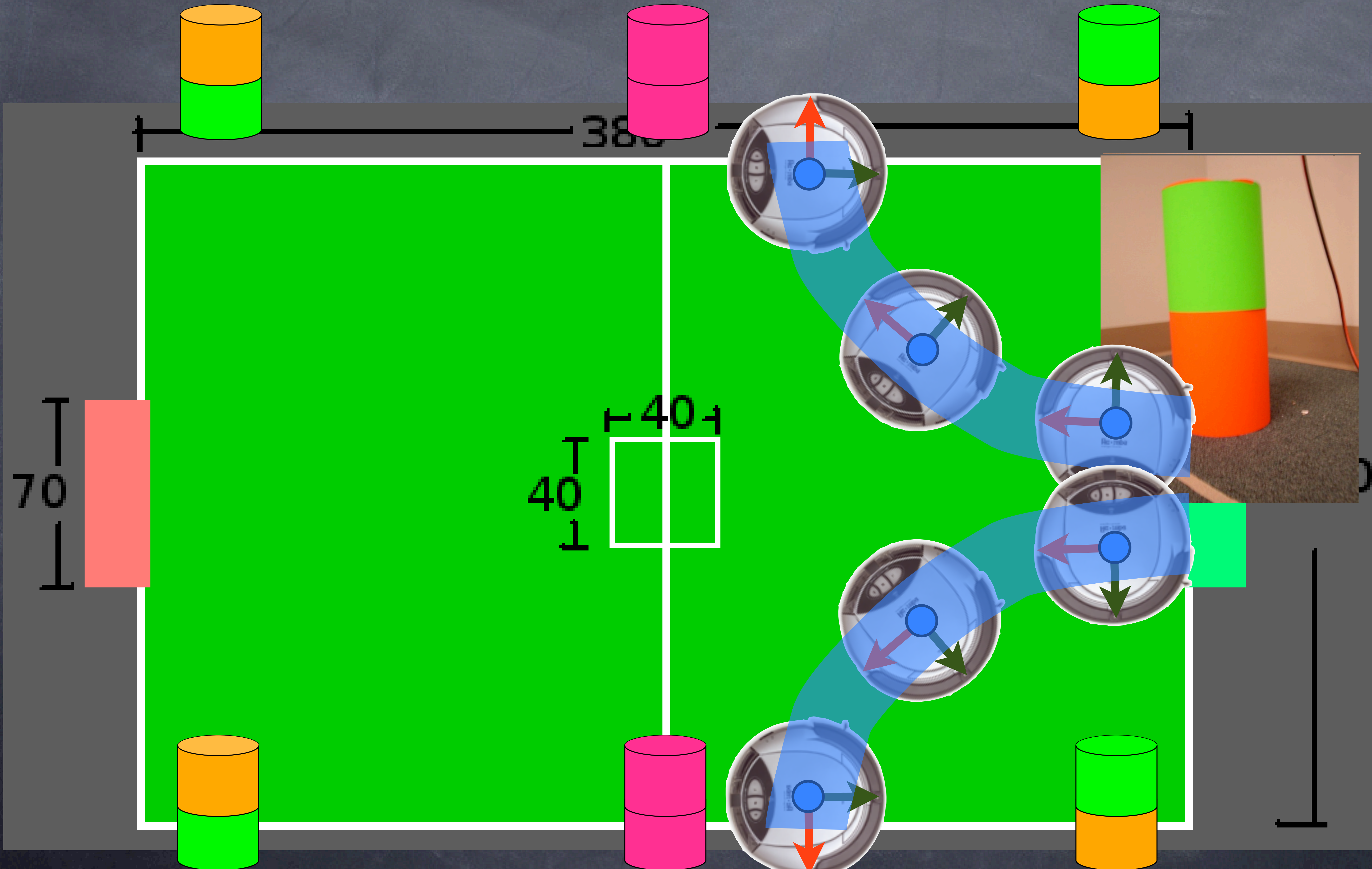
distribution over robot poses, or ...

p(x|y): probability of a pose given sensing

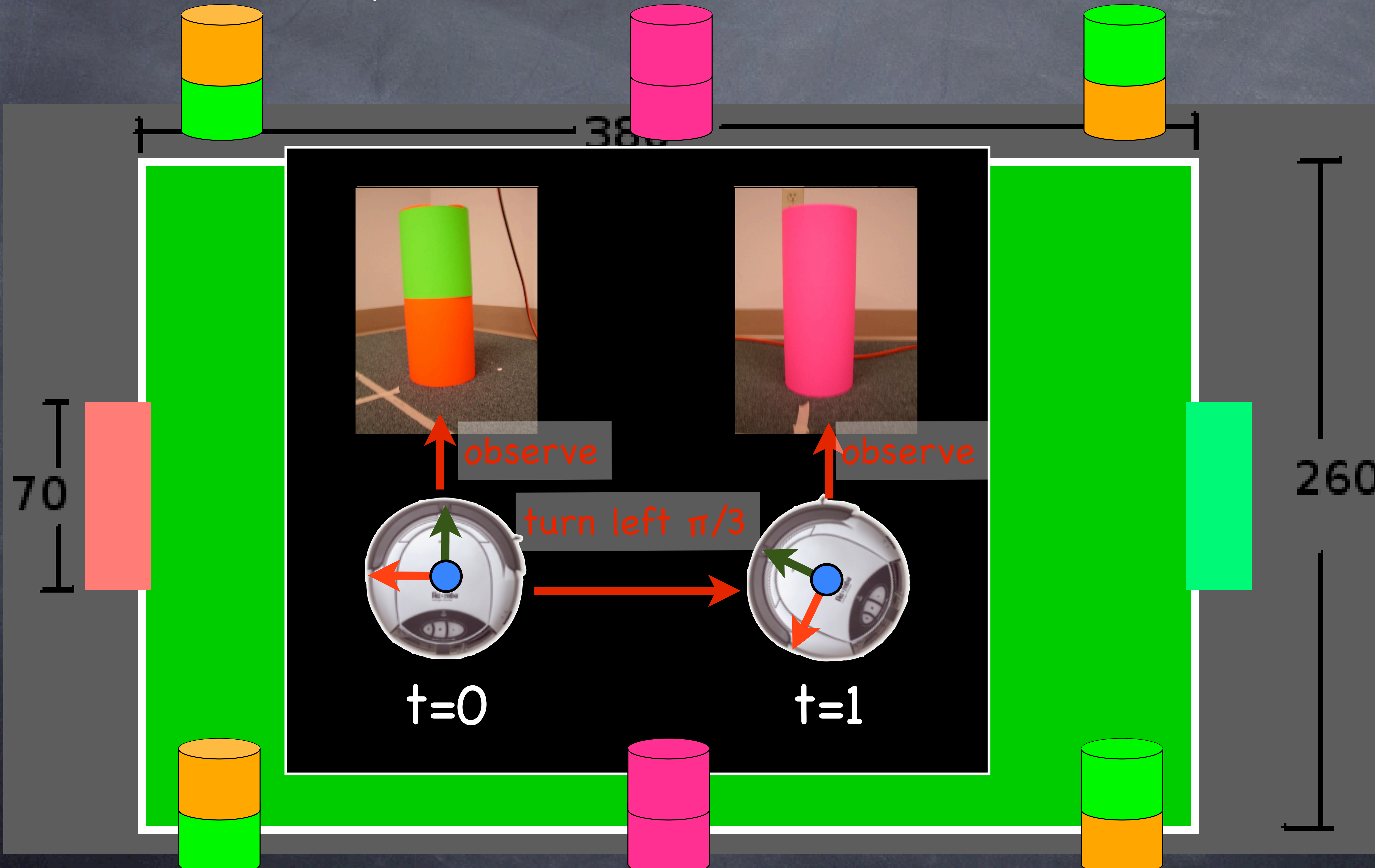
if the robot sees this landmark, where could it be?



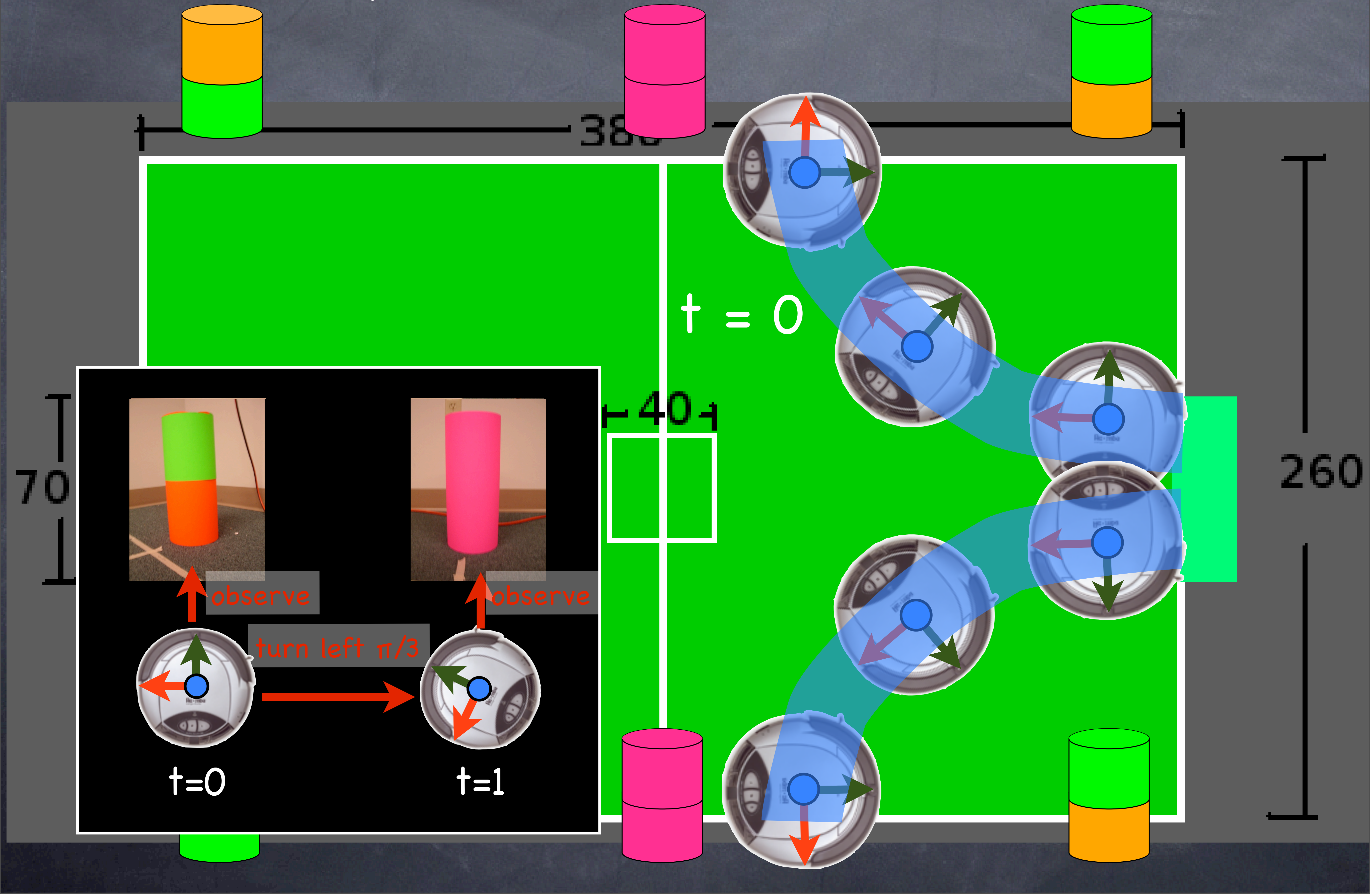
if the robot sees this landmark, where could it be?



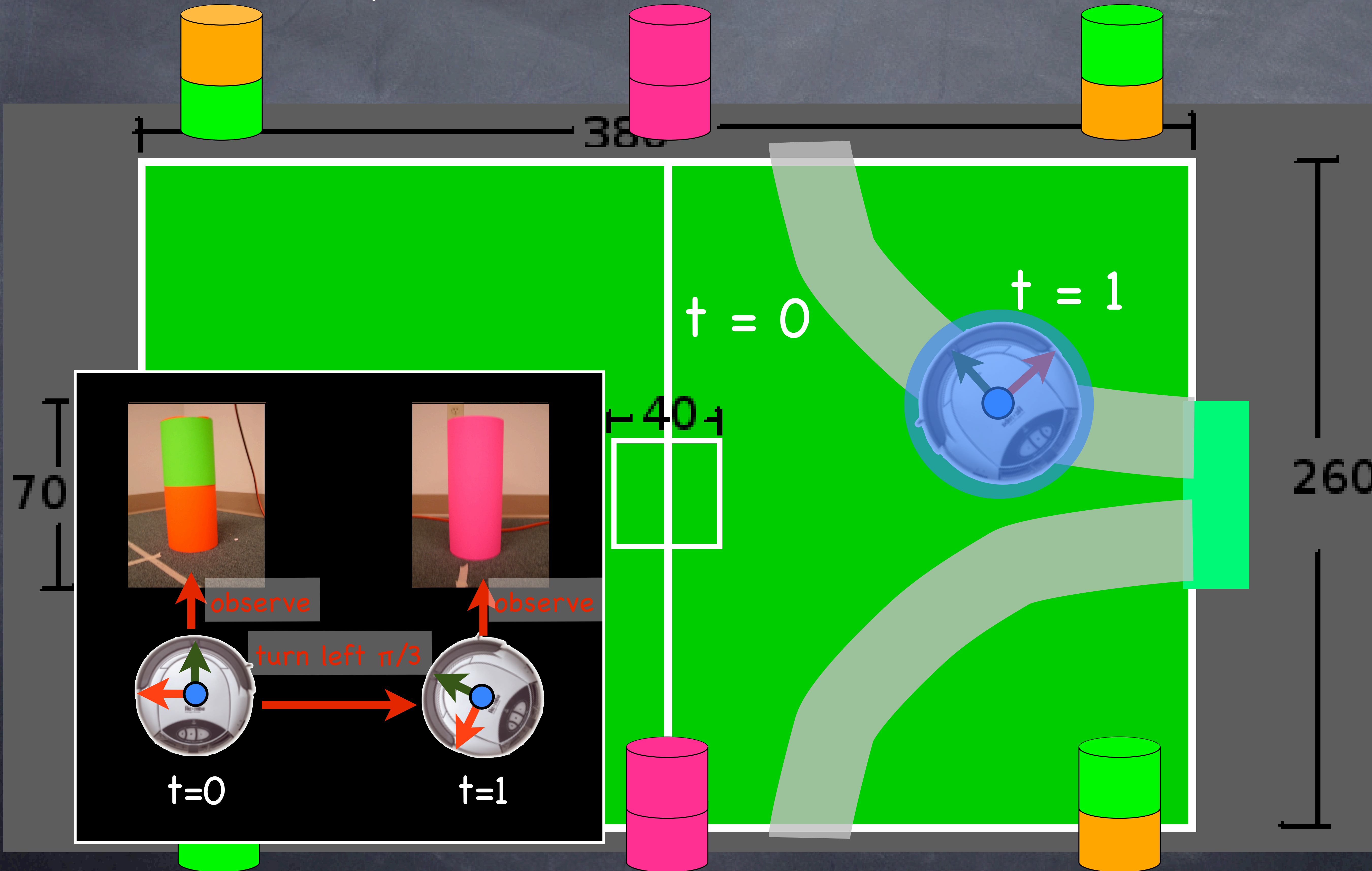
suppose the robot starts by seeing this landmark, then turns left, and then sees another landmark



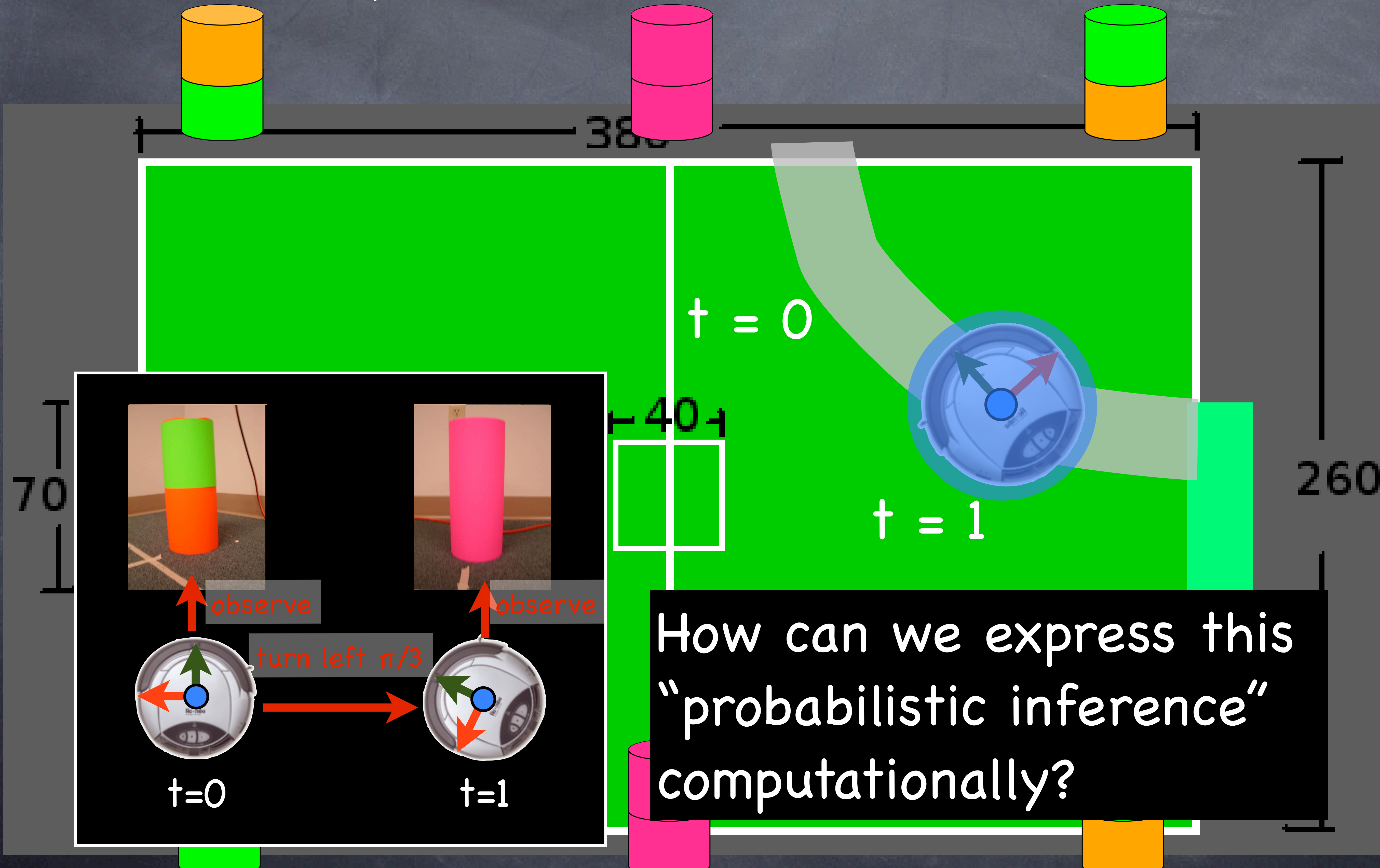
suppose the robot starts by seeing this landmark, then turns left, and then sees another landmark



suppose the robot starts by seeing this landmark, then turns left, and then sees another landmark



suppose the robot starts by seeing this landmark, then turns left, and then sees another landmark



How can we express this "probabilistic inference" computationally?

Bayes Rule

- relates one conditional probability to its inverse
  - posterior is proportional to likelihood * prior
  - A: possible robot pose, B: given robot observation

likelihood: consistency of an assumed pose with given observations

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

prior: previous knowledge/bias about robot poses

posterior: resultant distribution of poses considering observations

normalization: distributions must sum to 1 note: Z is y in control loop

Bayesian Filtering

posterior

likelihood

"update"

prior: predict new belief from previous belief

"predict"

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{Z}_k|\mathbf{Z}_{k-1})}$$

assume constant denominator, becomes constant α

$$= \alpha\, p(\mathbf{Z}_k|\mathbf{x}_k)\underbrace{p(\mathbf{x}_k|\mathbf{Z}_{k-1})}$$

restate prior into dynamics and recursive prior

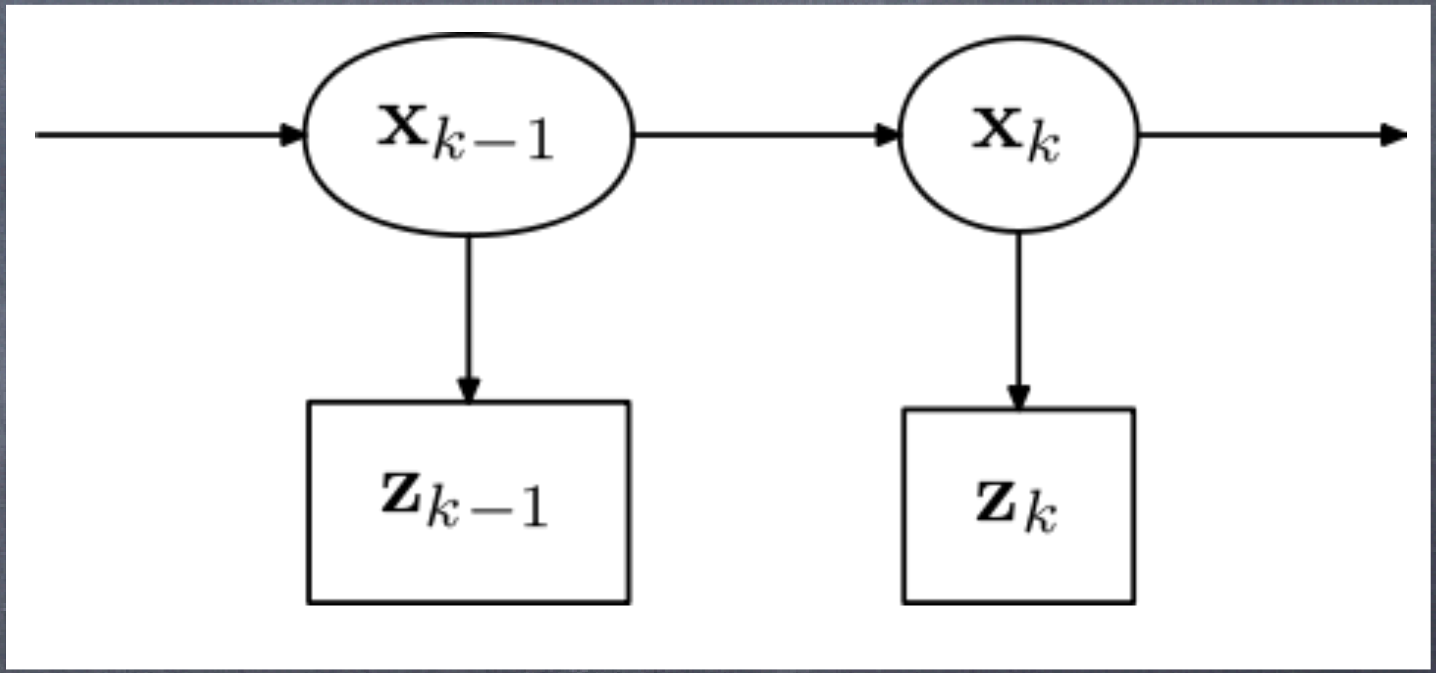
$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}$$

dynamics: predict new belief from previous belief

prior: belief from previous time step

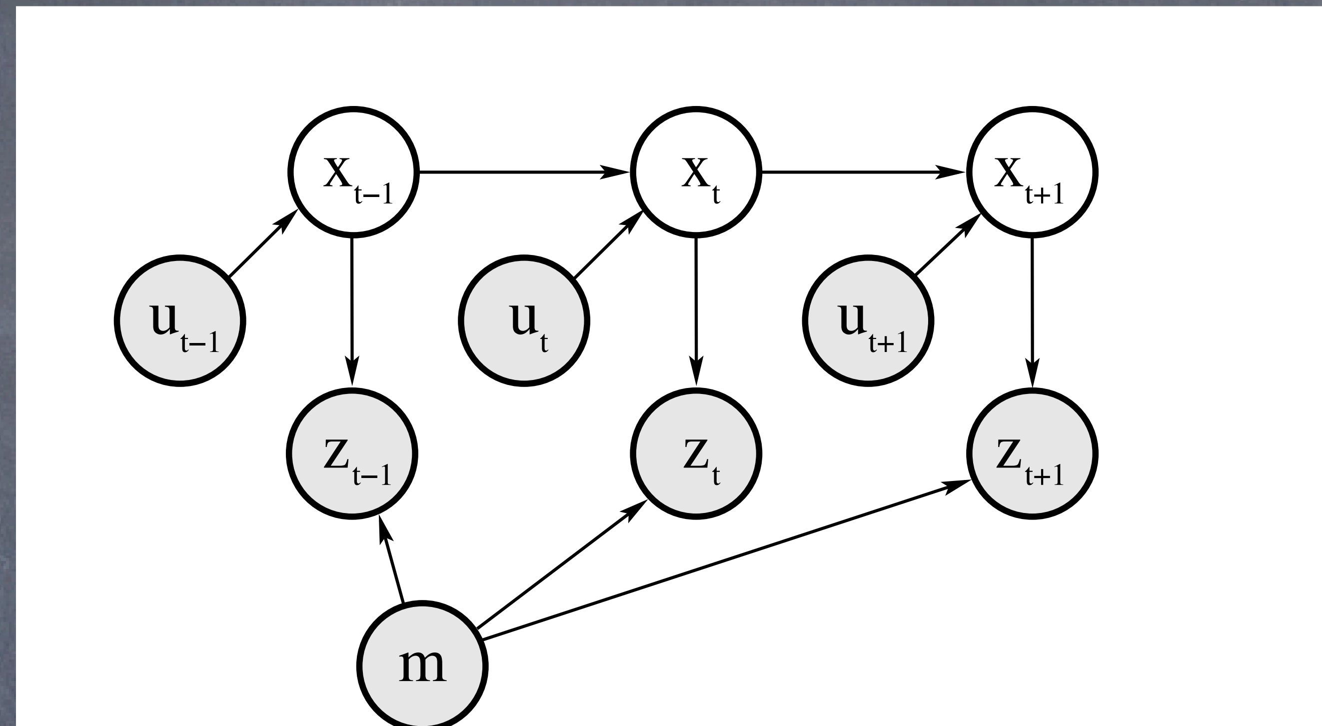
Bayesian Filtering is...

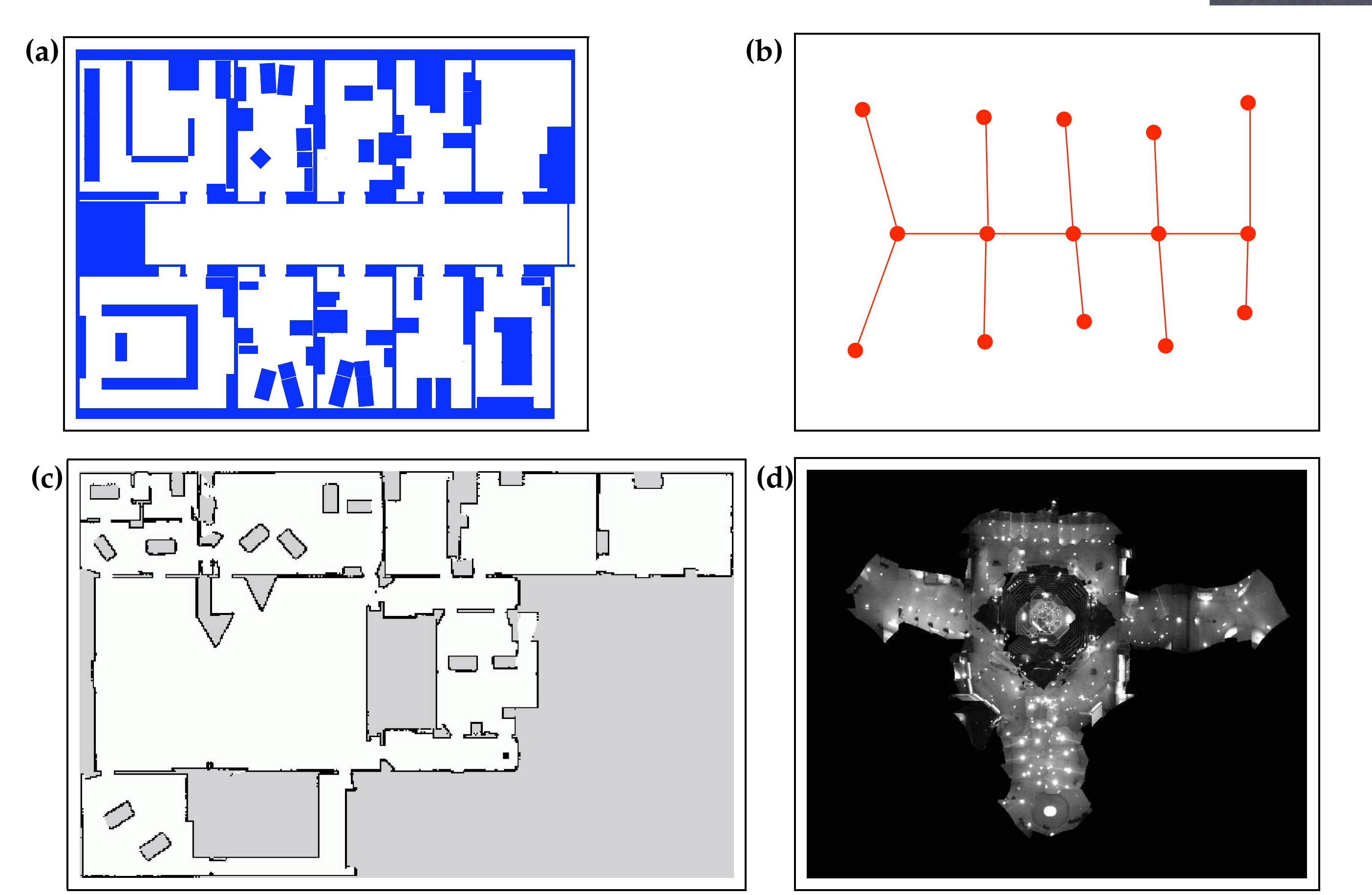
- **Markovian**: future depends on only current state
- recursive inference in time
- expressible as a "graphical model"



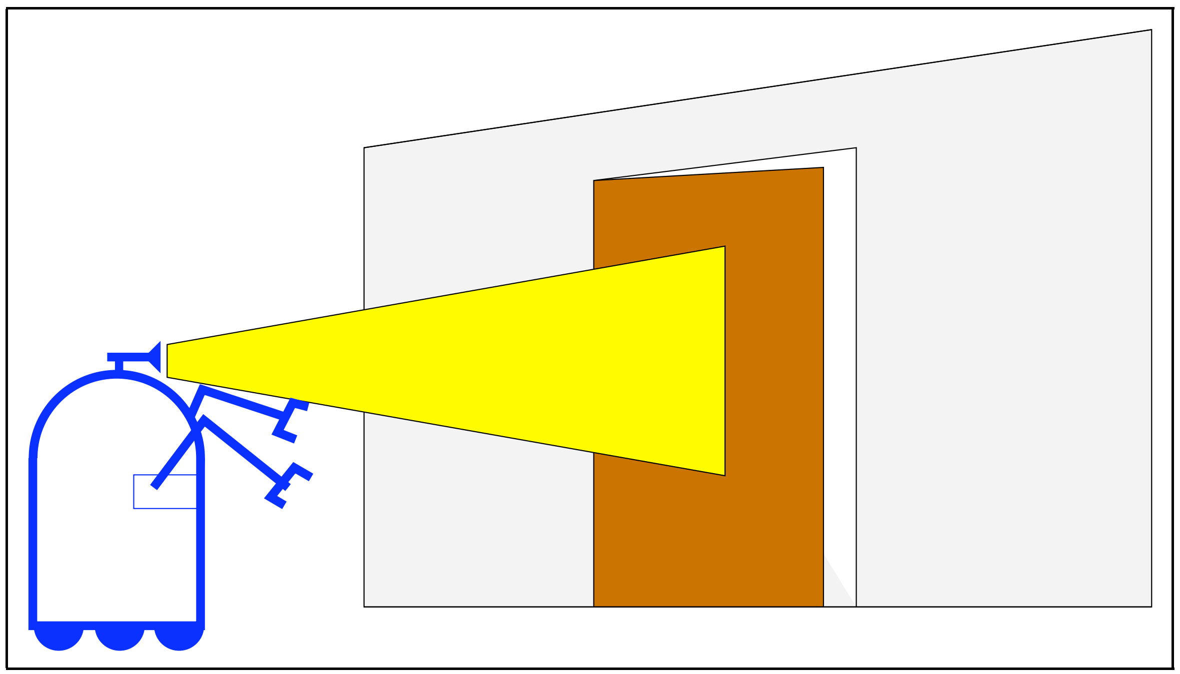
Bayesian Filtering for Robot Localization

- Assumes a map **m**
  - metric
  - topological
- Choice of belief distribution
  - Grid
  - Gaussian
  - Multi-hypothesis

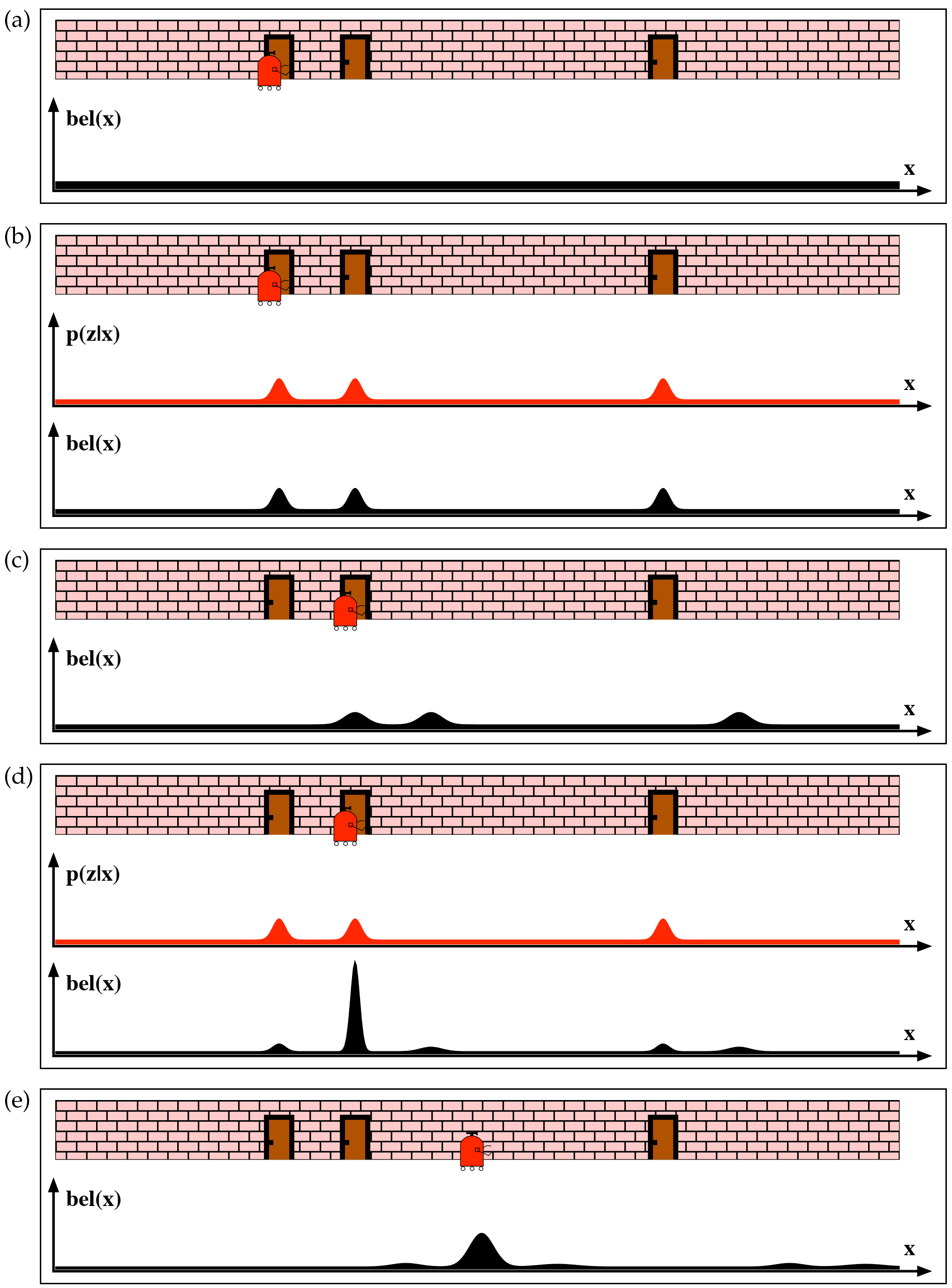




1D Hallway Robot Example [Thrun et al 2005]



Robot can sense "door" or "wall" at location

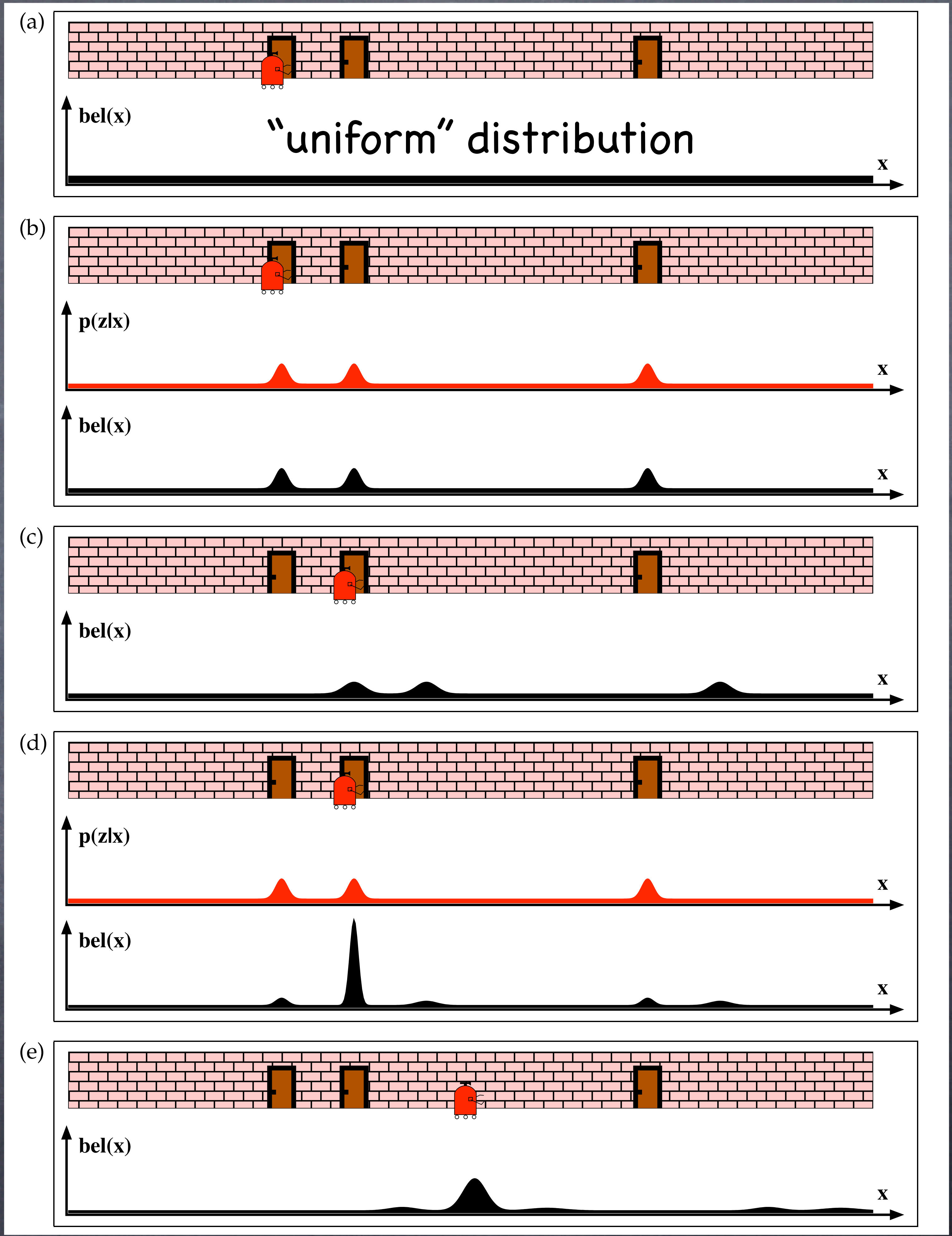


Evolution of continuous Bayes Filter

Can we infer this distribution on our robots?

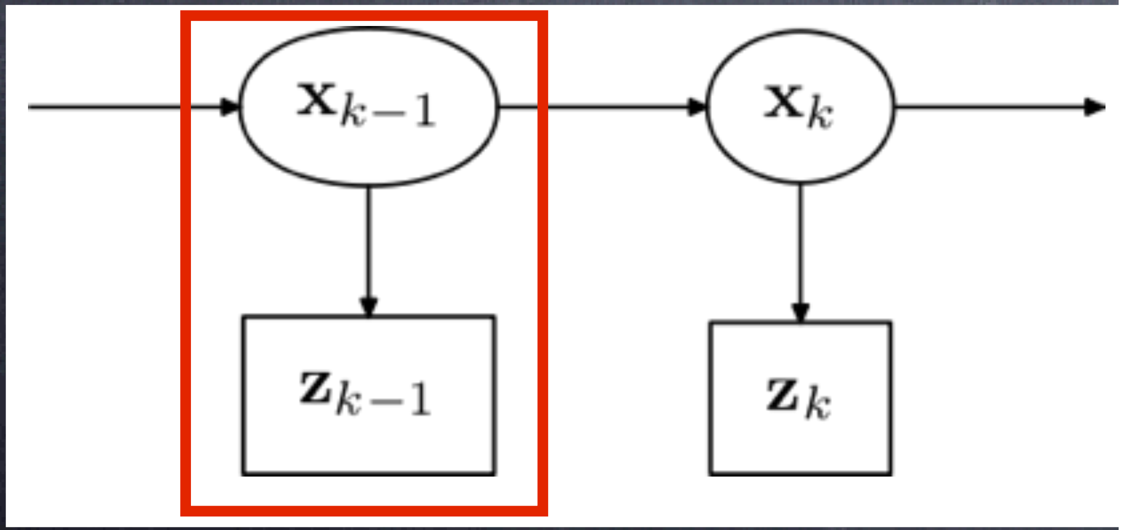
t=0, start

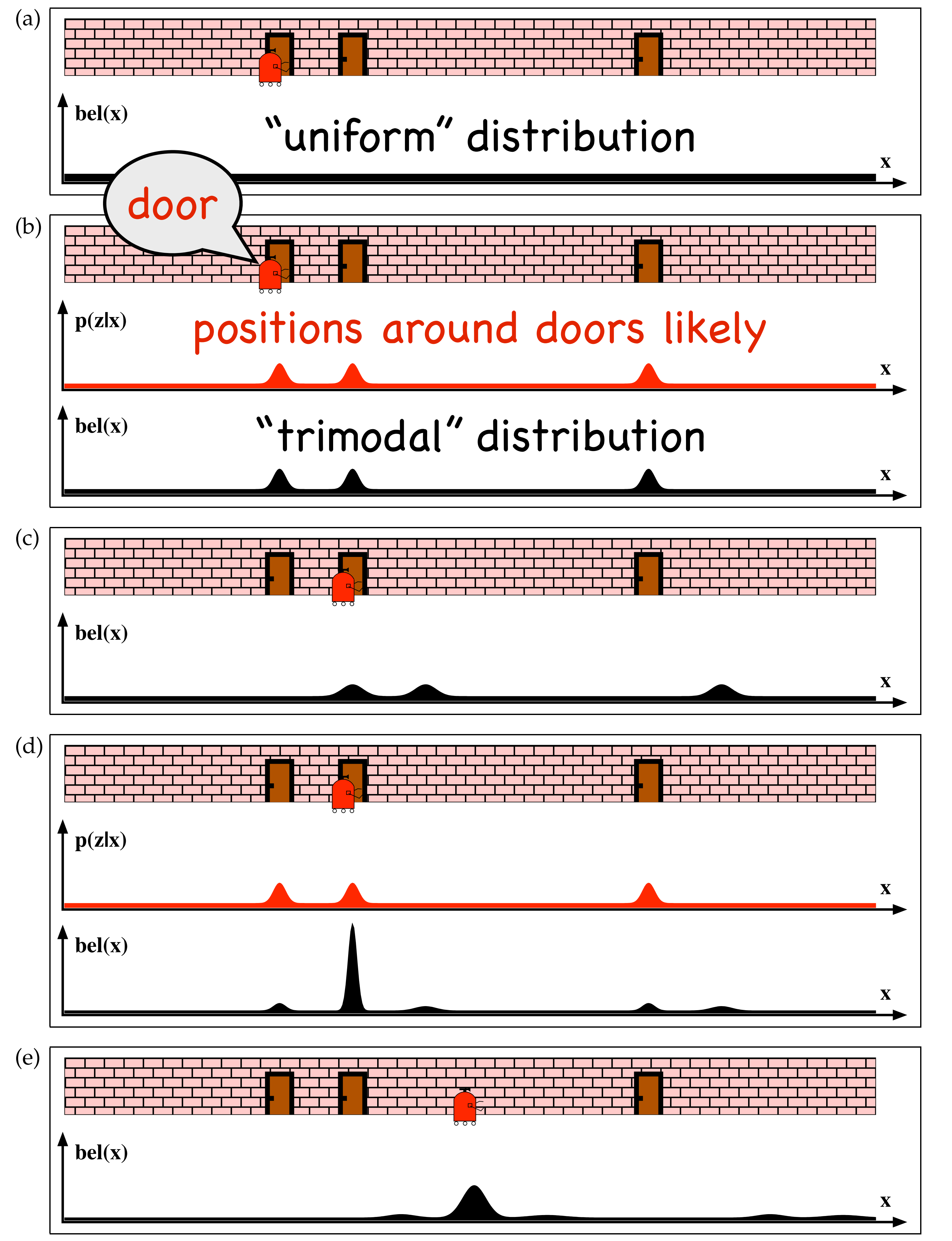
(all poses equally probable)



t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

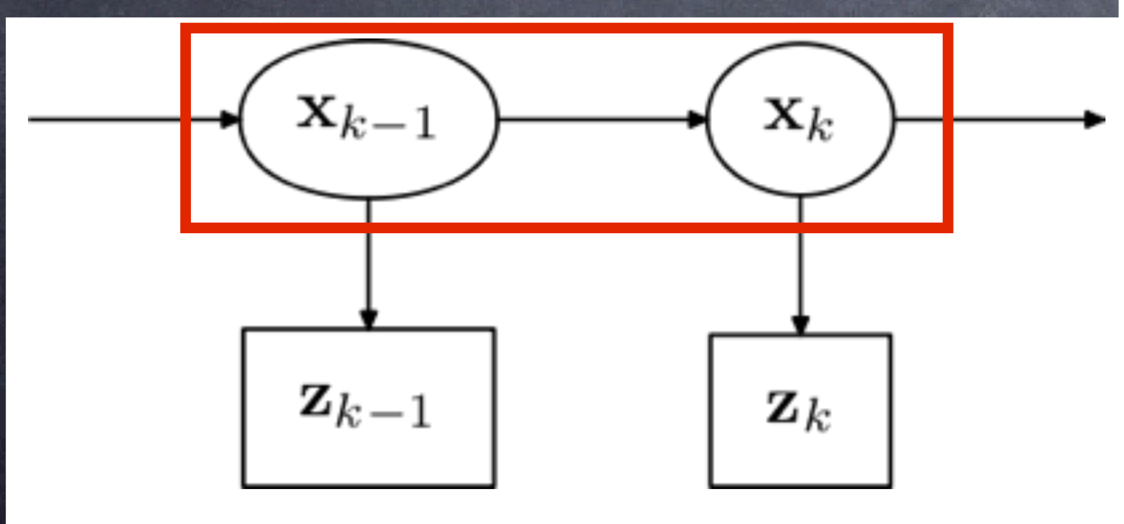


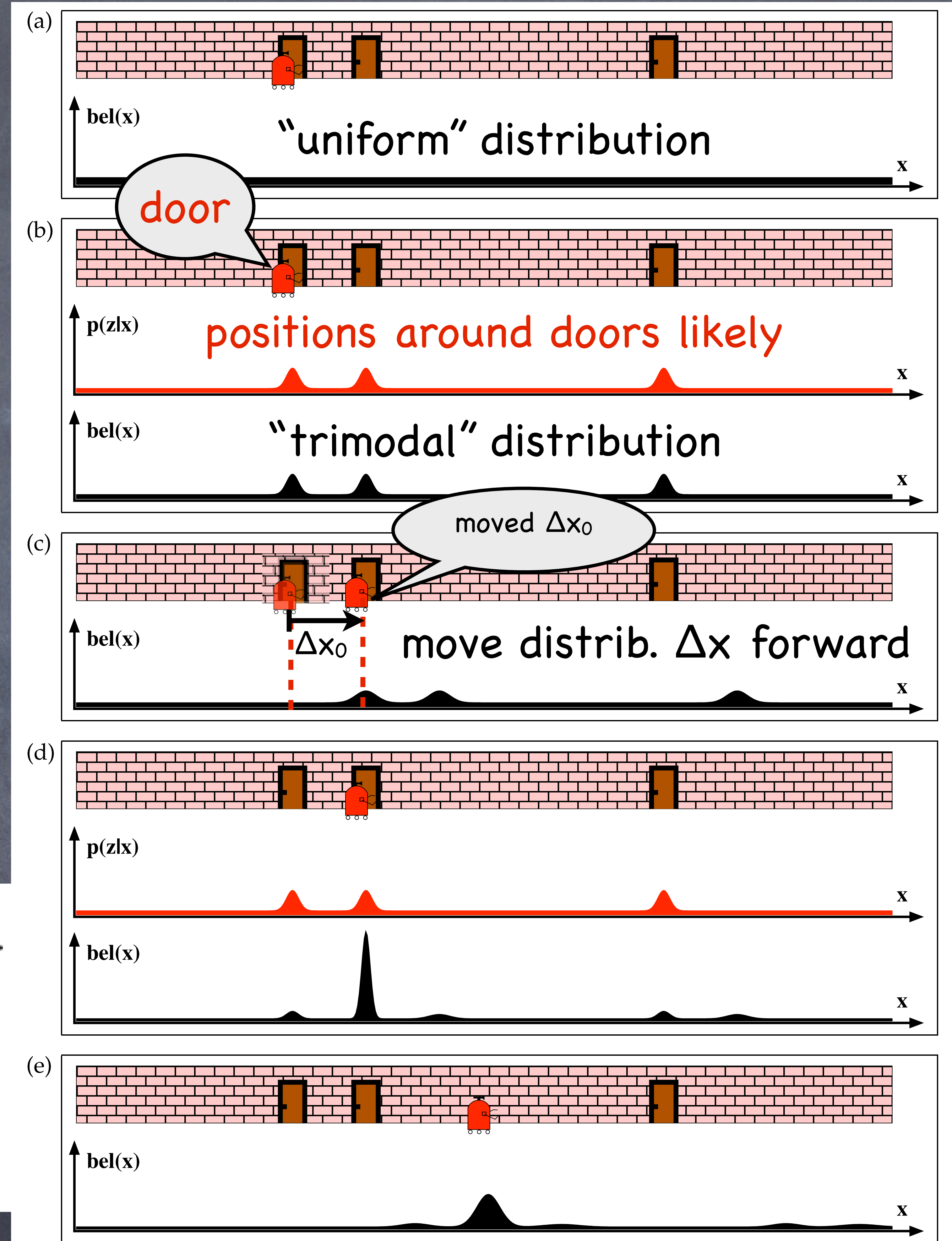


t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)



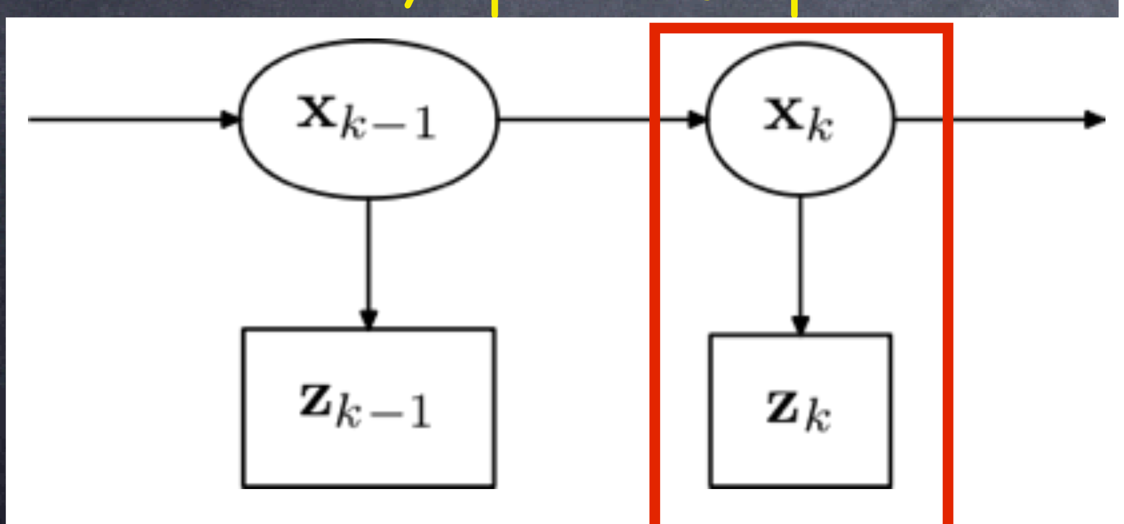


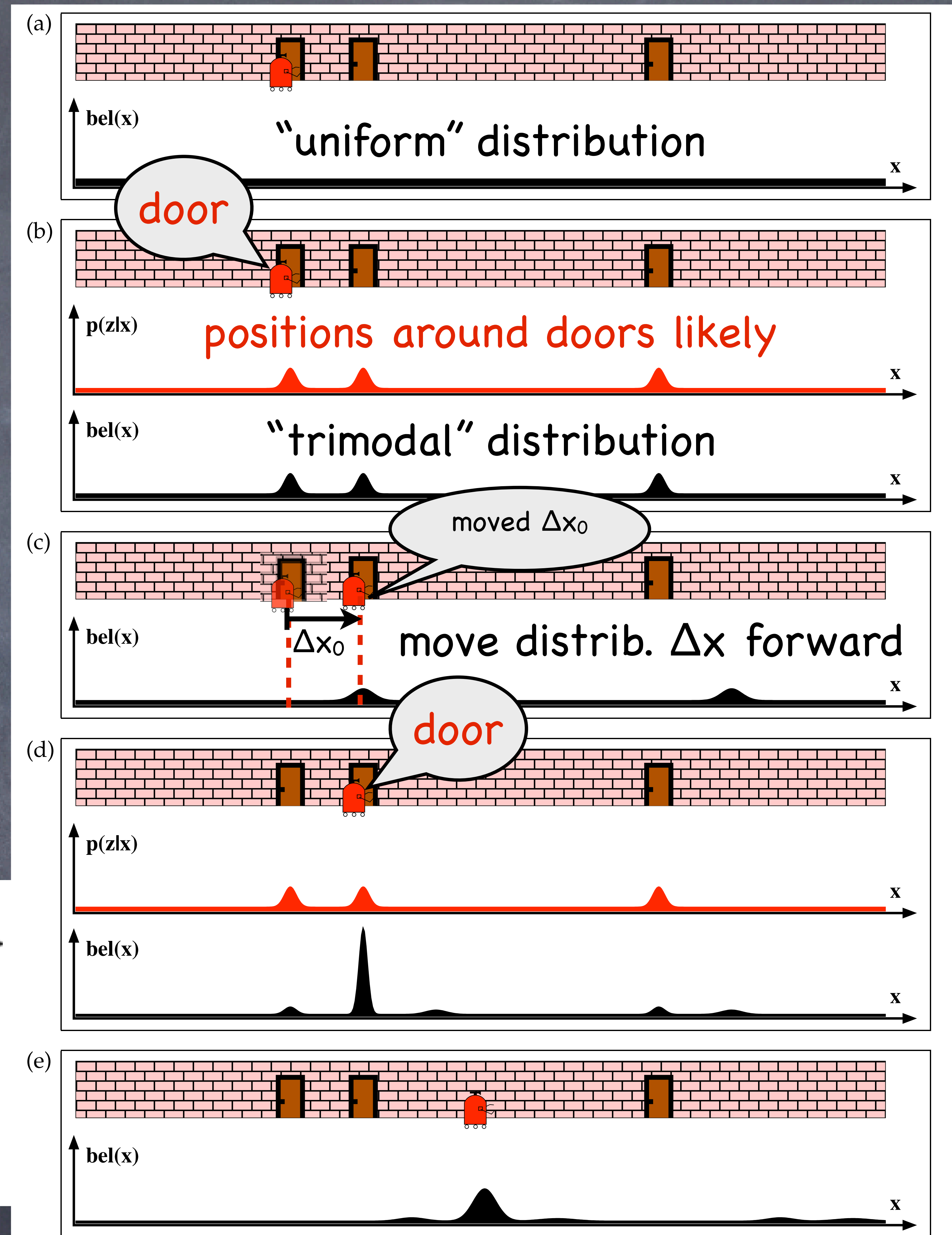
t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step



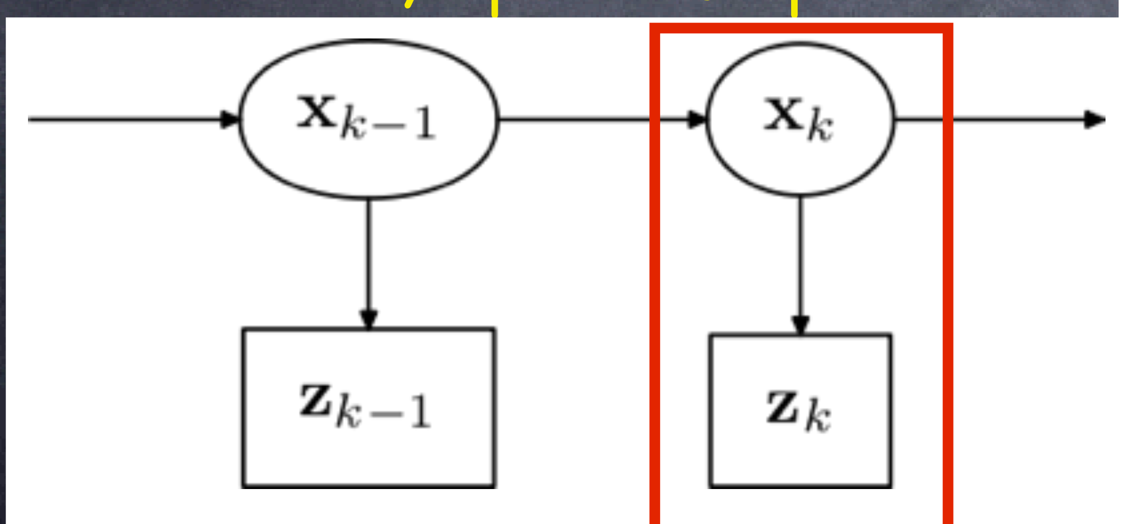


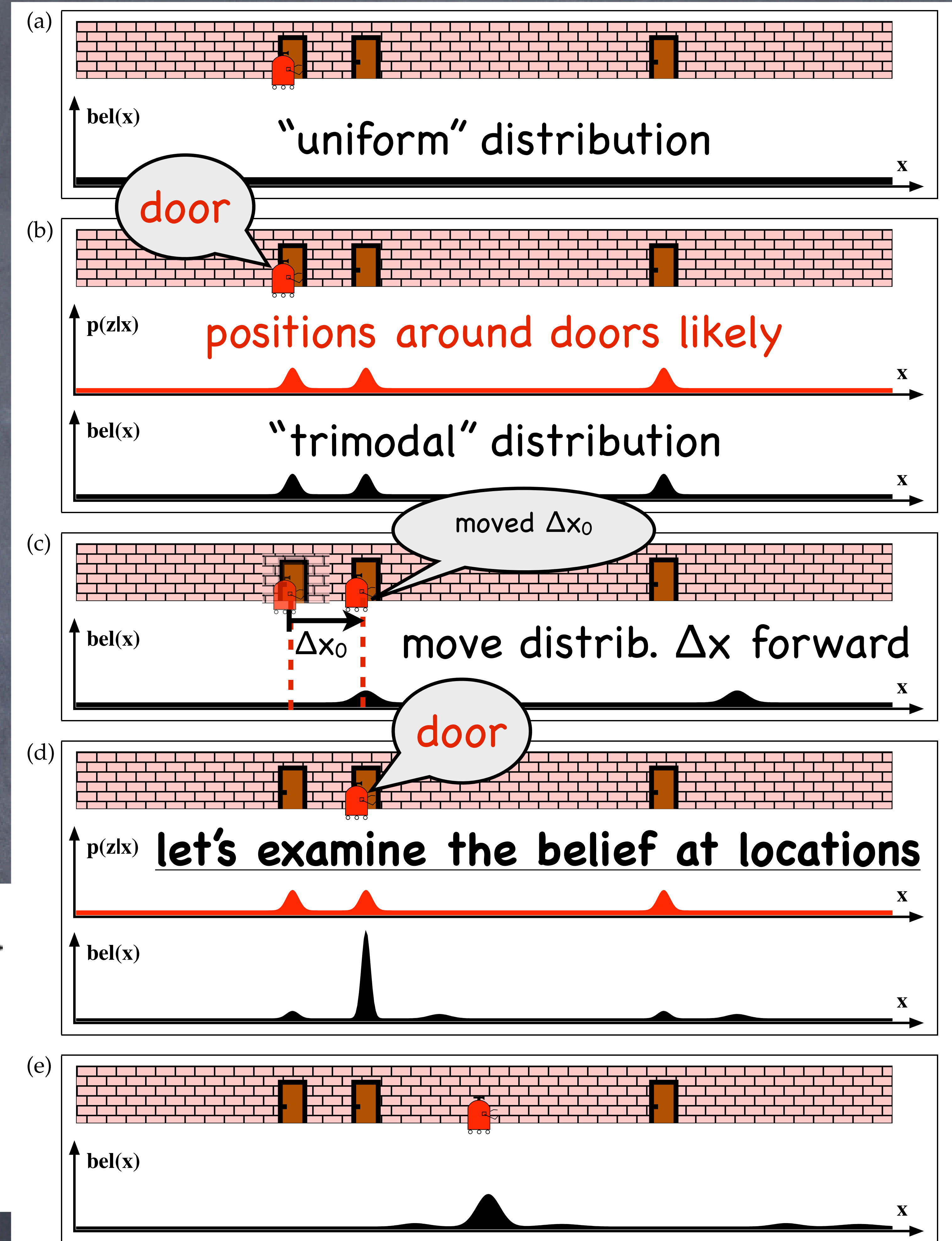
t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step



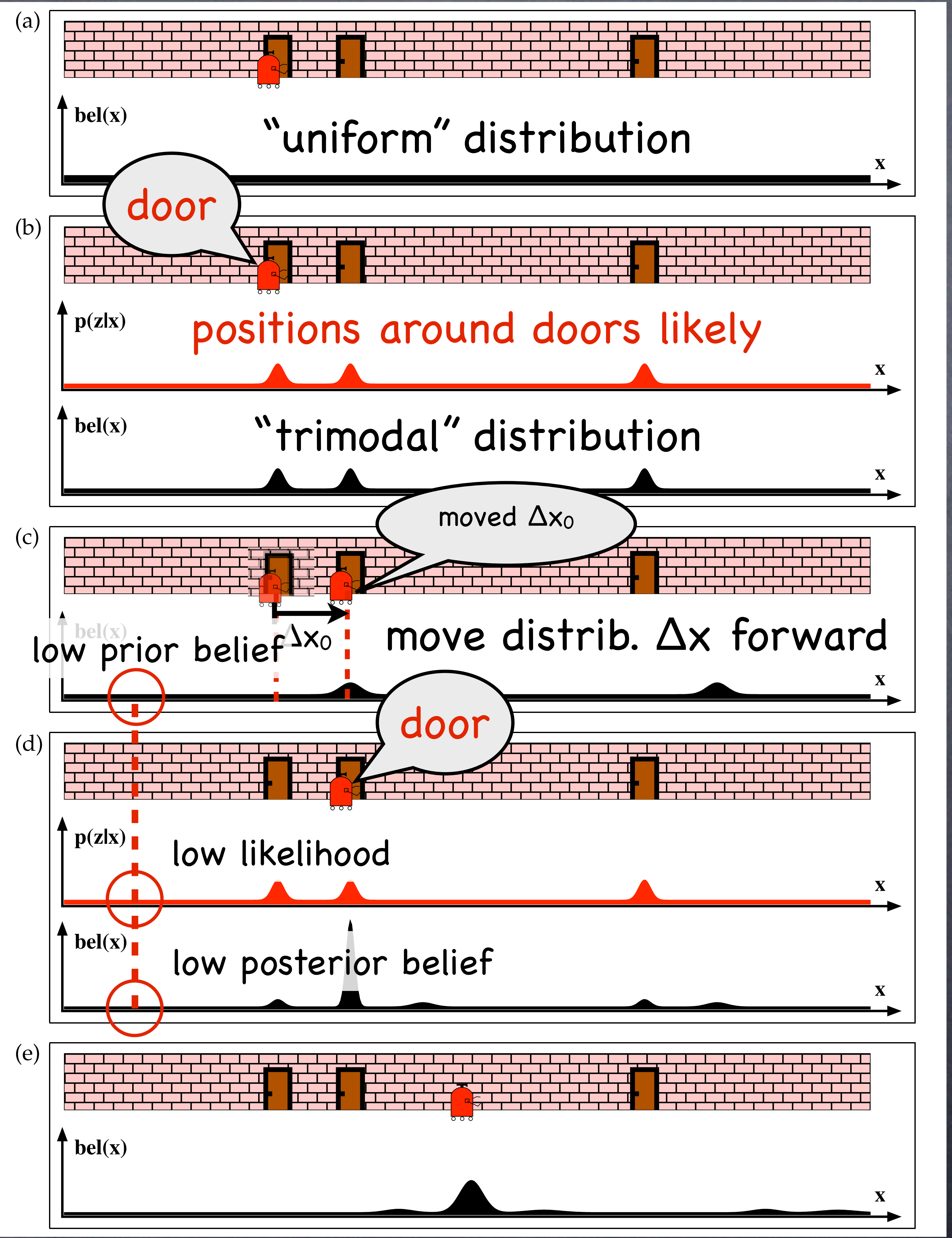


t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step
(belief peaks about true position)

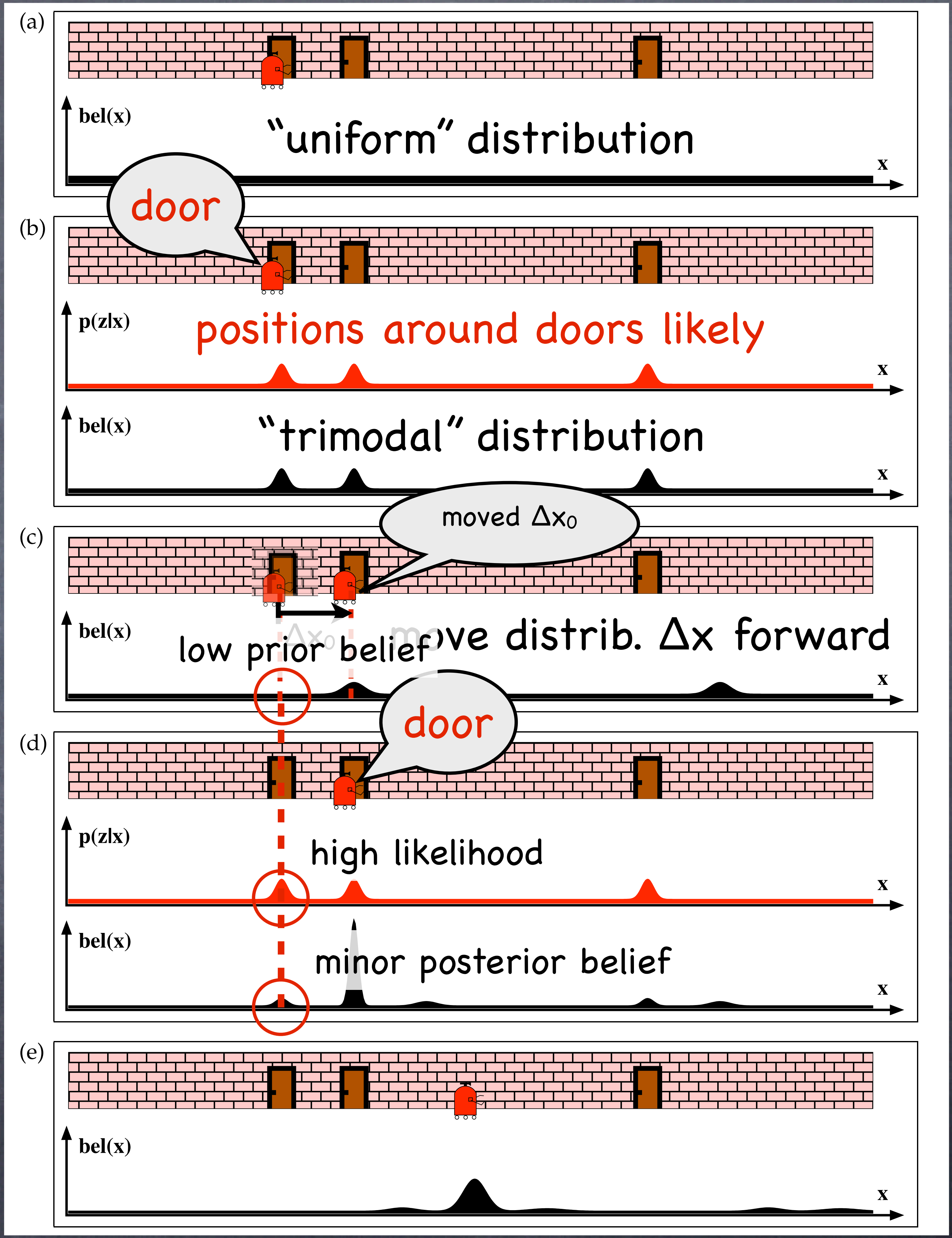


t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step
(belief peaks about true position)

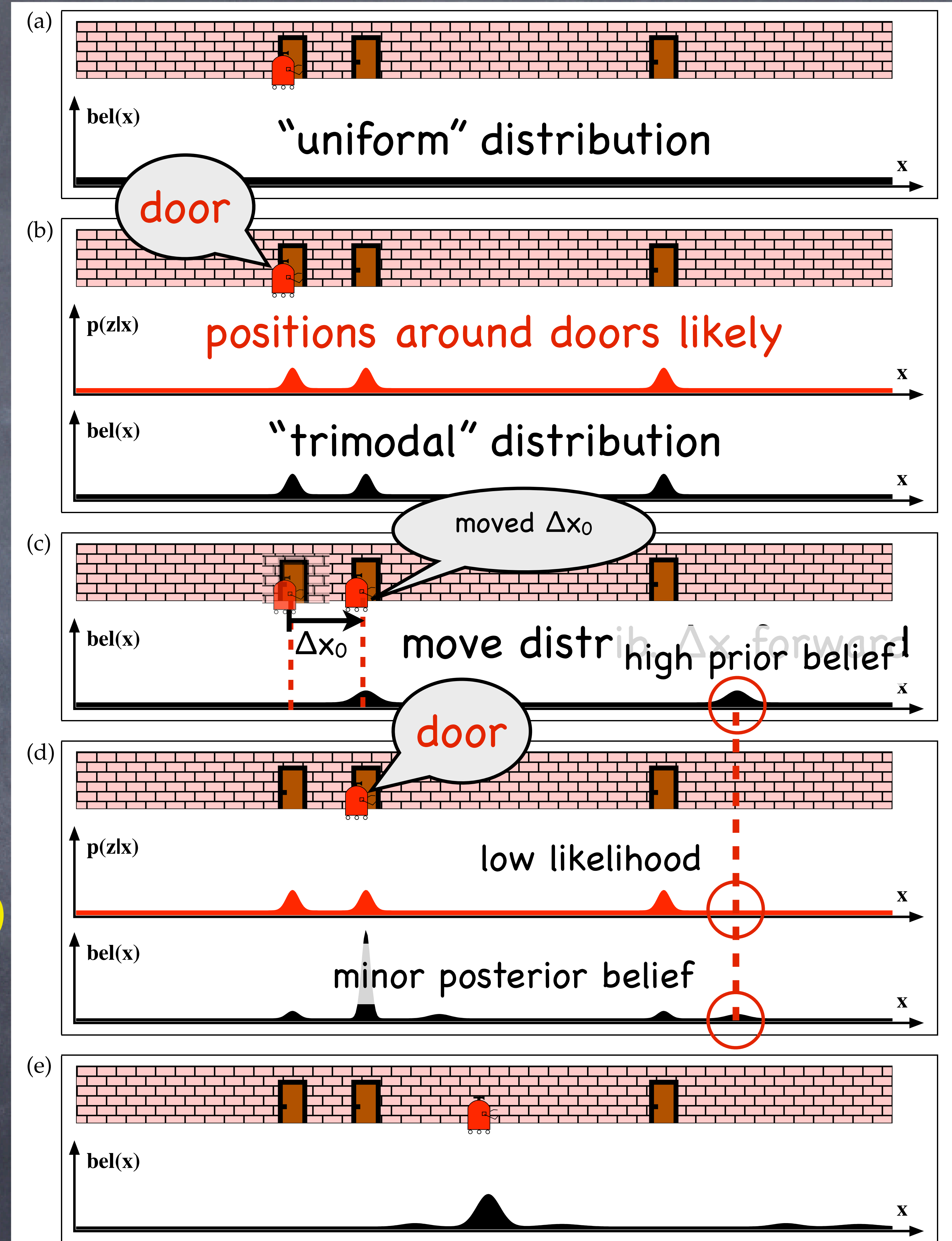


t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step
(belief peaks about true position)

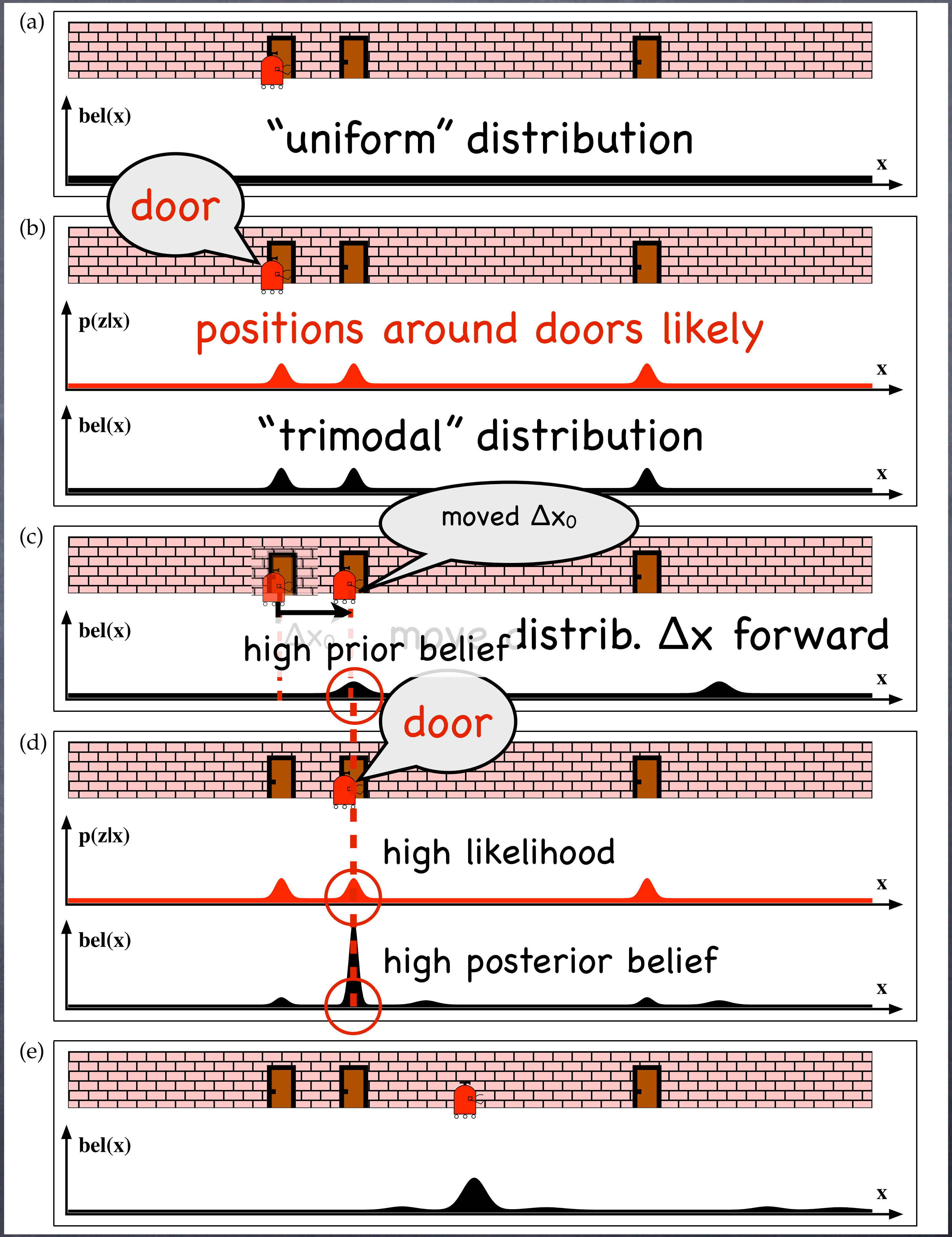


t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step
(belief peaks about true position)



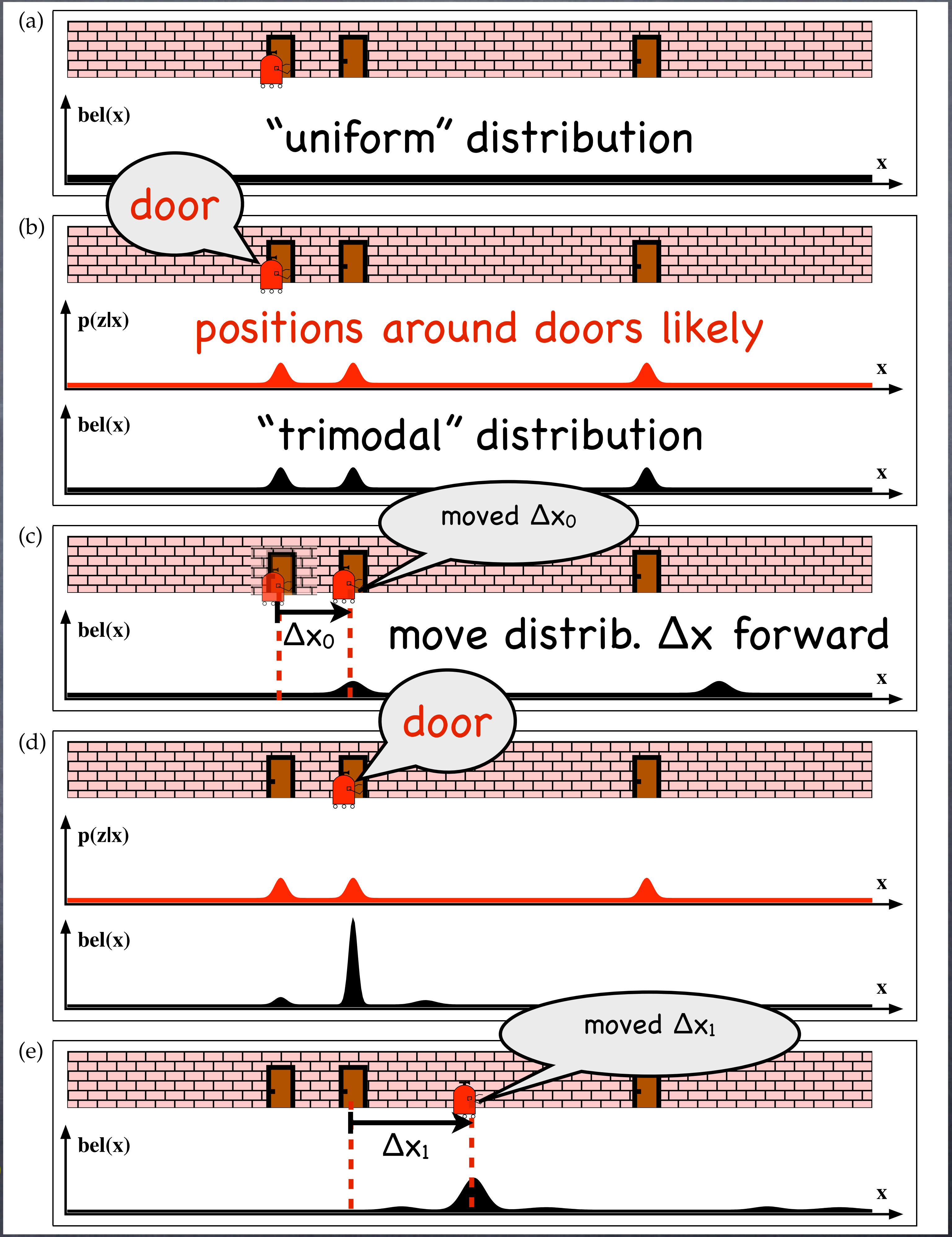
t=0, start
(all poses equally probable)

t=0, update step
(belief amasses around doors)

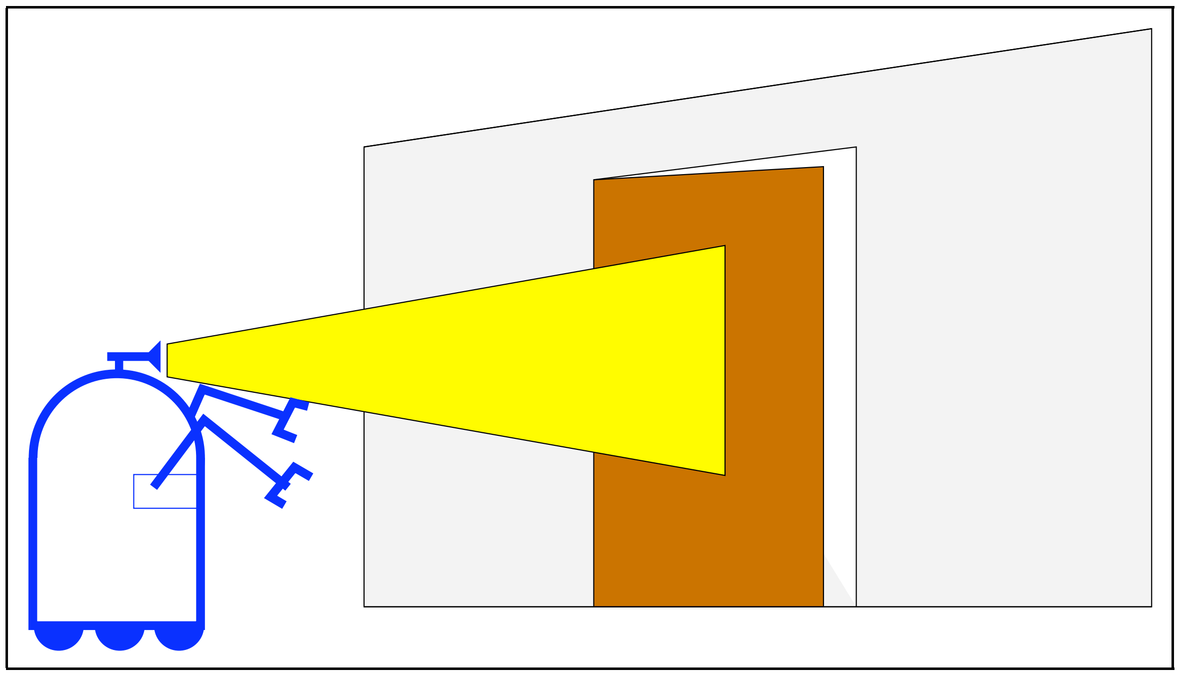
t=1, predict step
(belief moves with odometry;
diffuses due to odometry noise)

t=1, update step
(belief peaks about true position)

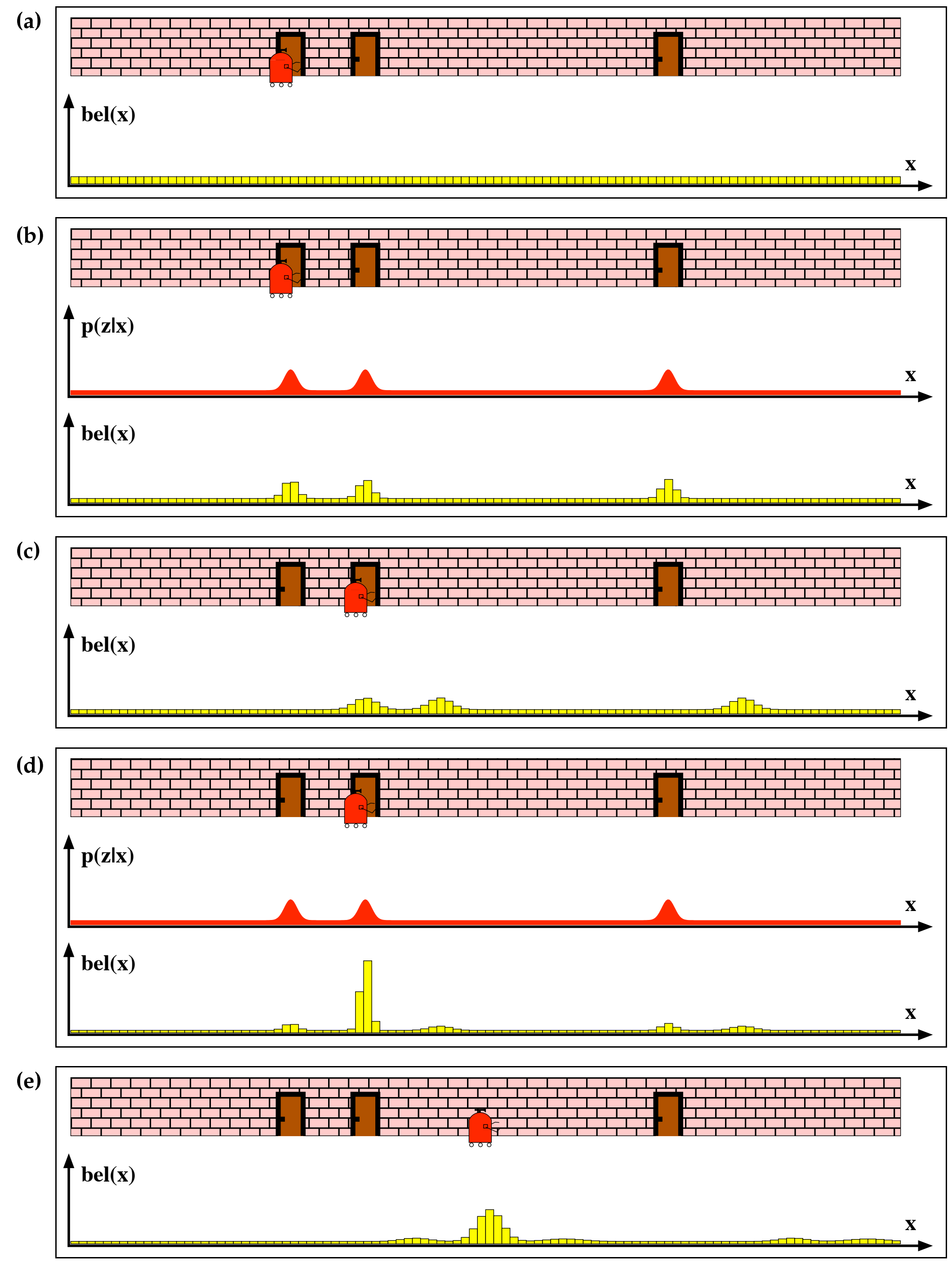
t=2, predict step
(belief moves with robot, diffuses)



Grid-based Filter

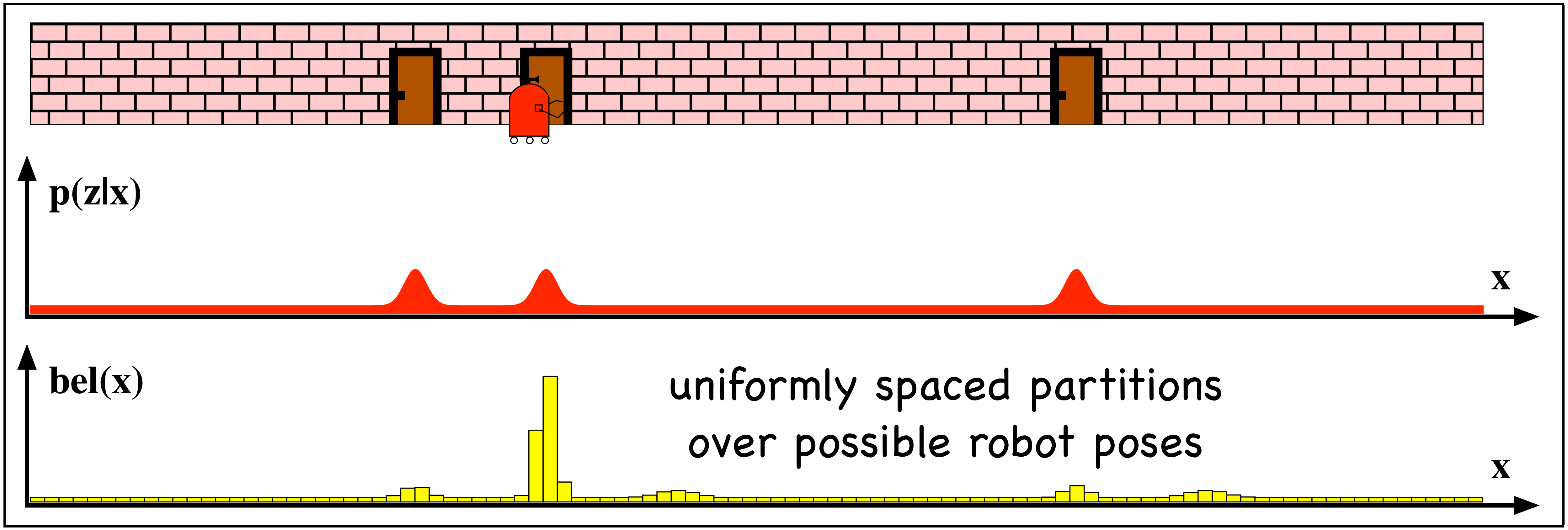


Robot can sense “door” or “wall” at location

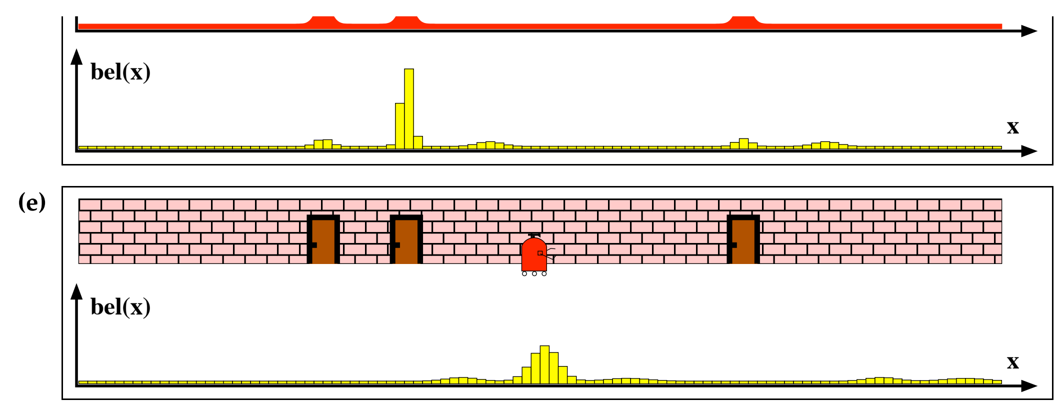


Evolution of discrete
grid-based Bayes Filter

Grid-based Filter

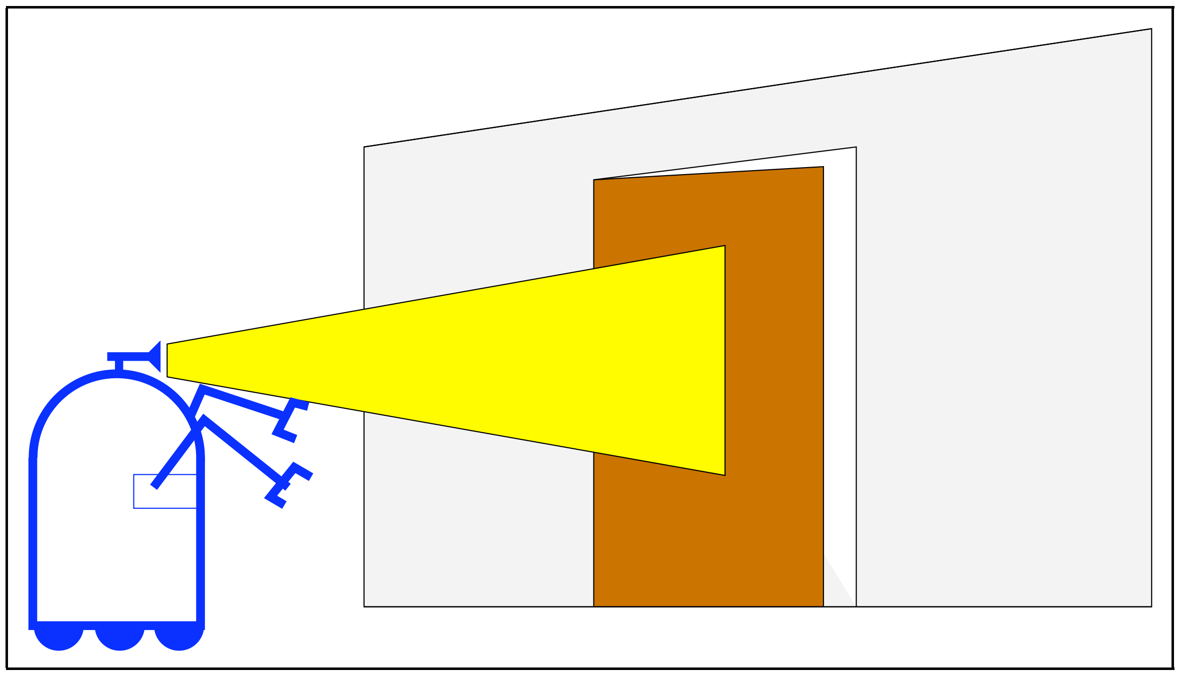


uniformly spaced partitions
over possible robot poses

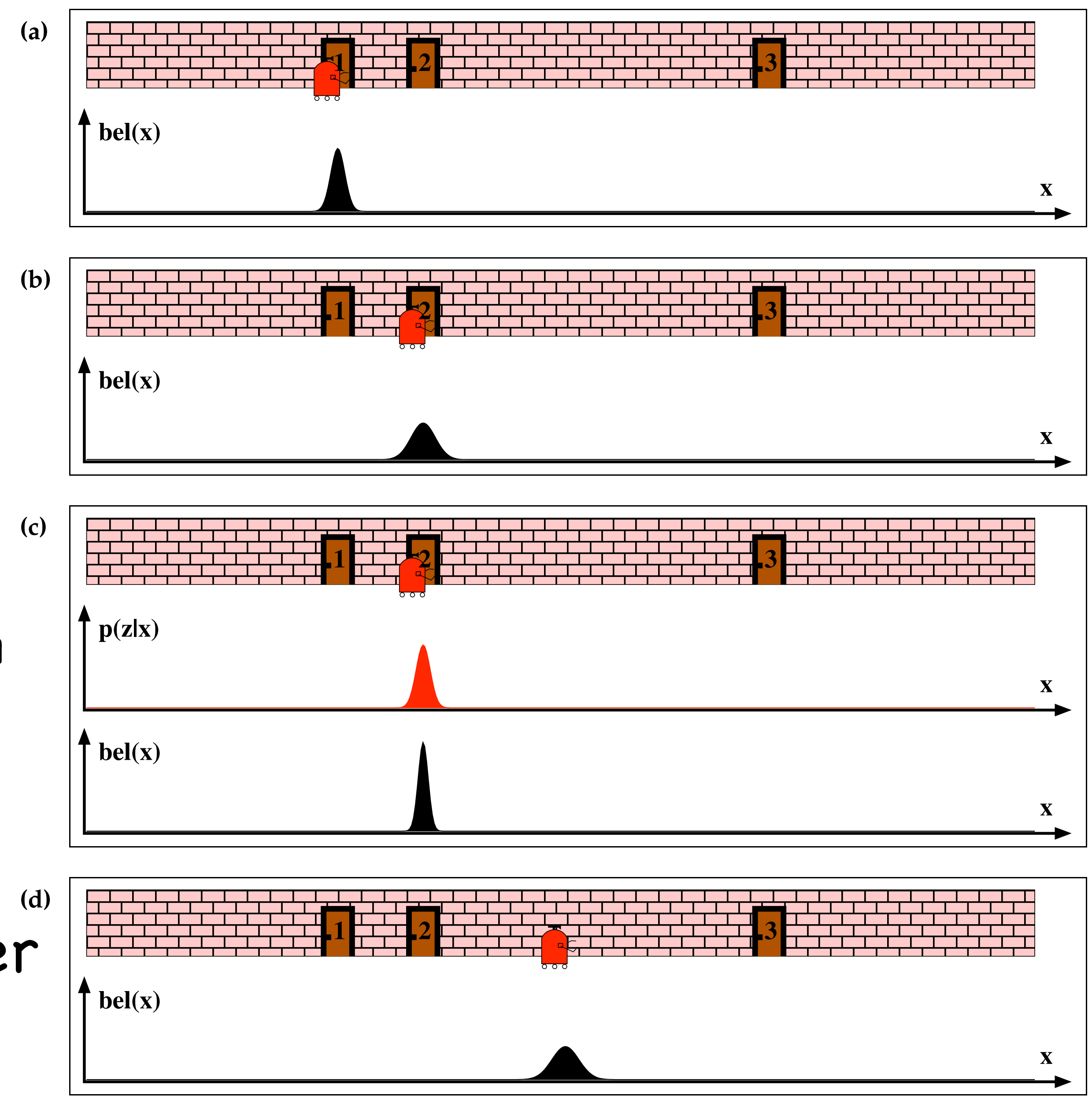


Evolution of discrete
grid-based Bayes Filter

Kalman Filter



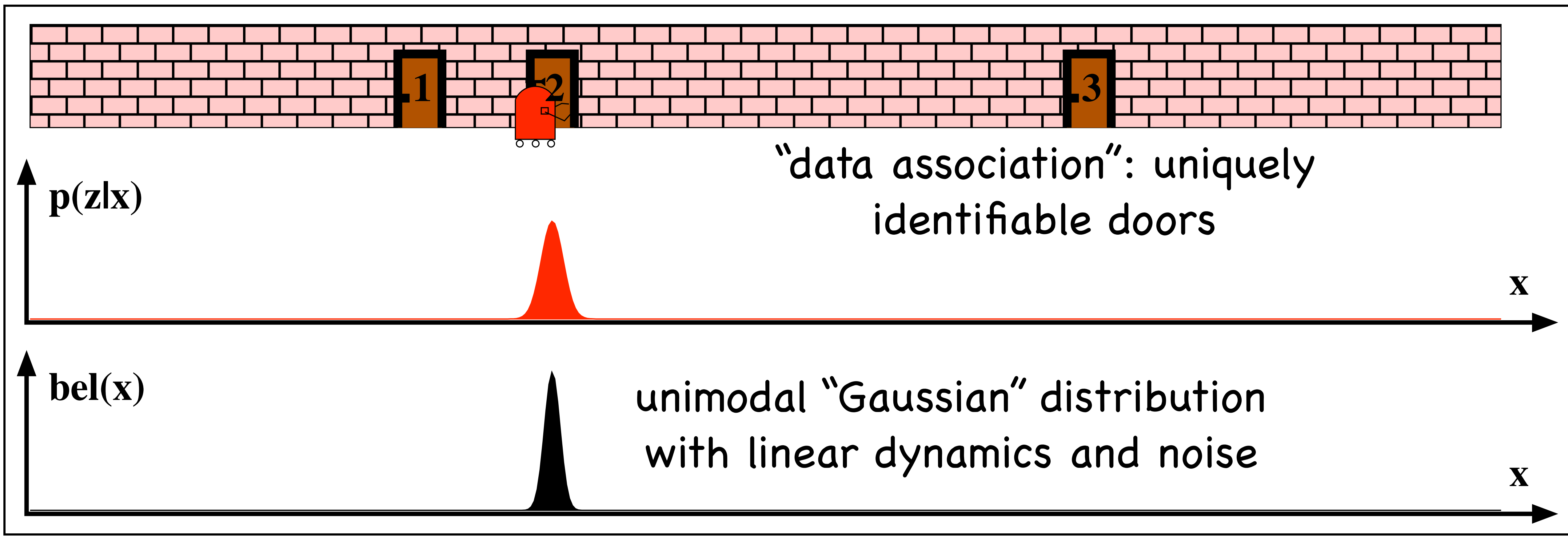
Robot can sense "door" or "wall" at location



Evolution of unimodal Gaussian-based Kalman Filter

variance of mode corresponds to uncertainty

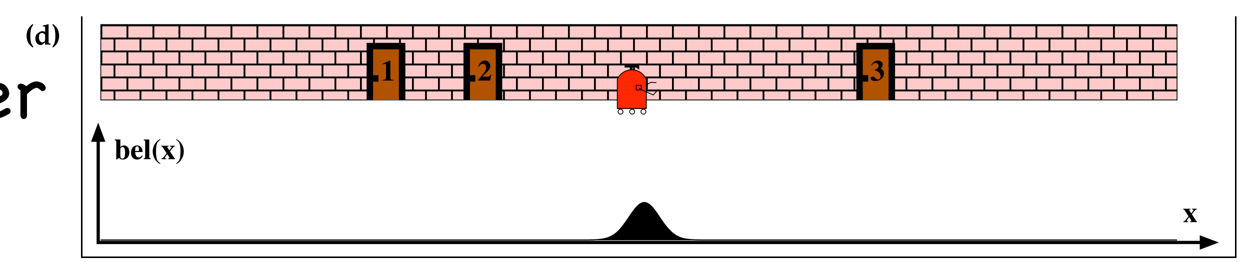
Kalman Filter



“data association”: uniquely identifiable doors

unimodal “Gaussian” distribution with linear dynamics and noise

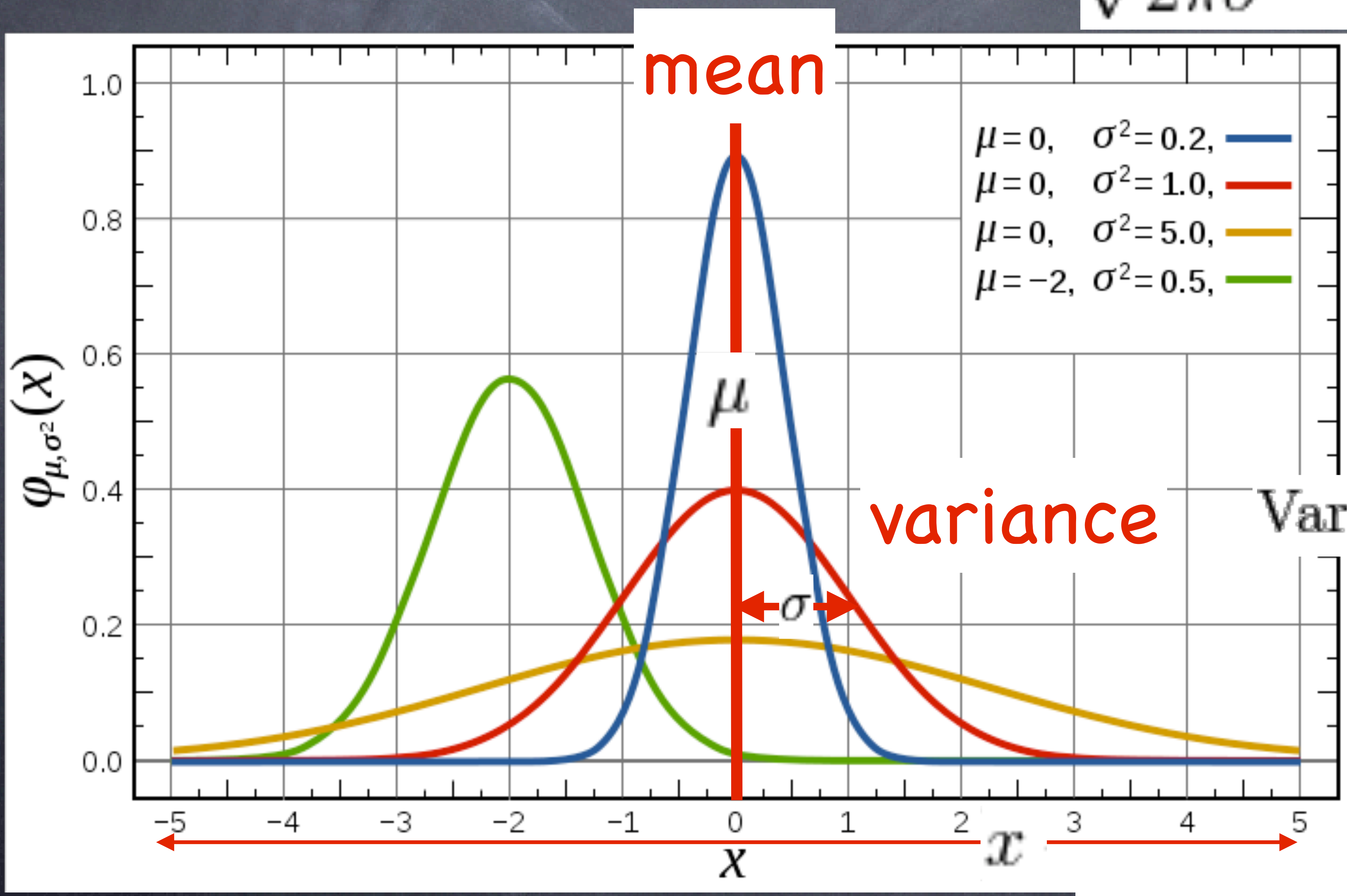
Gaussian-based Kalman Filter



variance of mode corresponds to uncertainty

Tangent: 1D Gaussian distribution

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



- mean: average or expected value
- Note: x here is only for this example

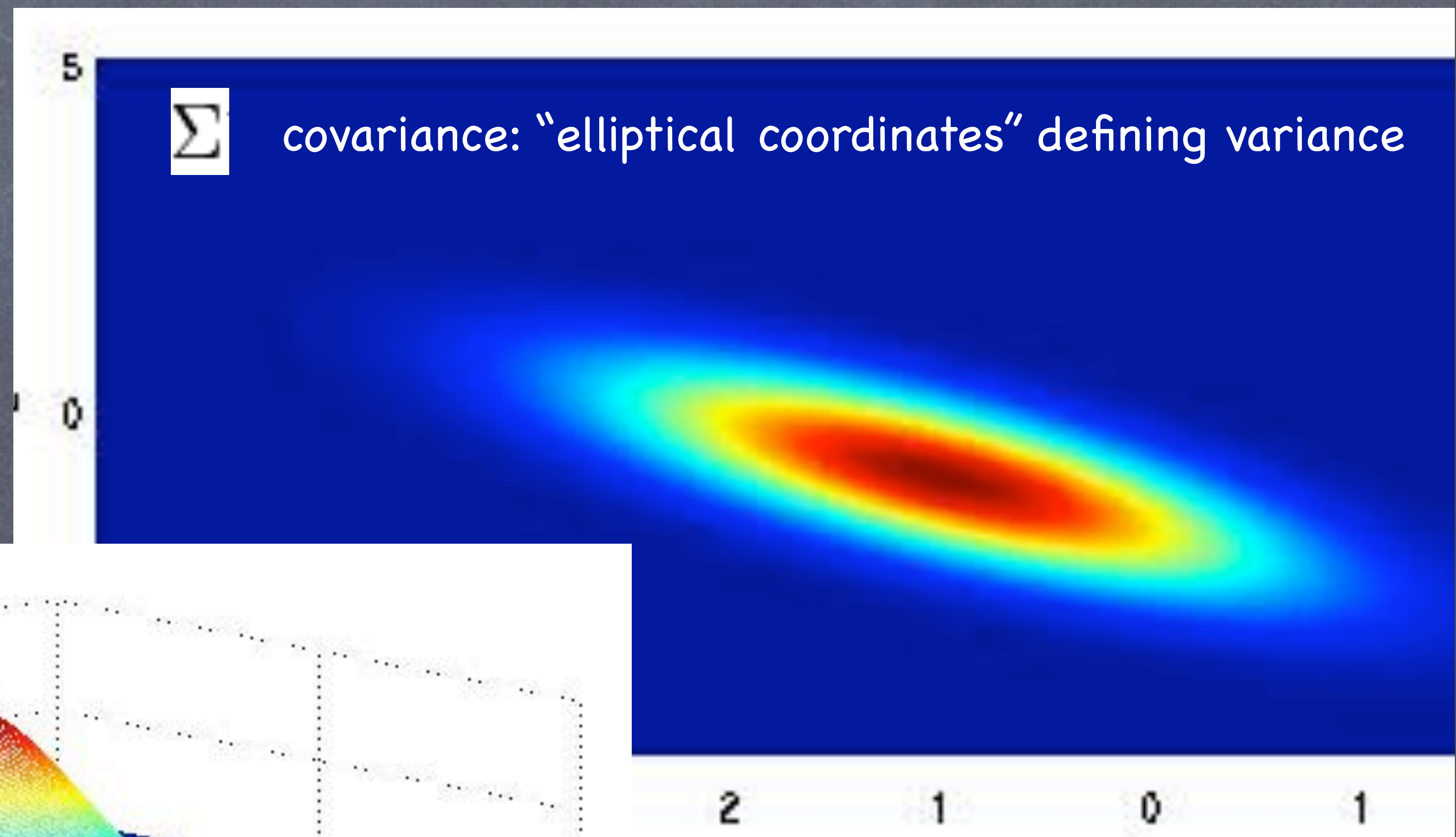
$$\mathrm{Var}(X) = \mathrm{E}[(X-\mu)^2]$$

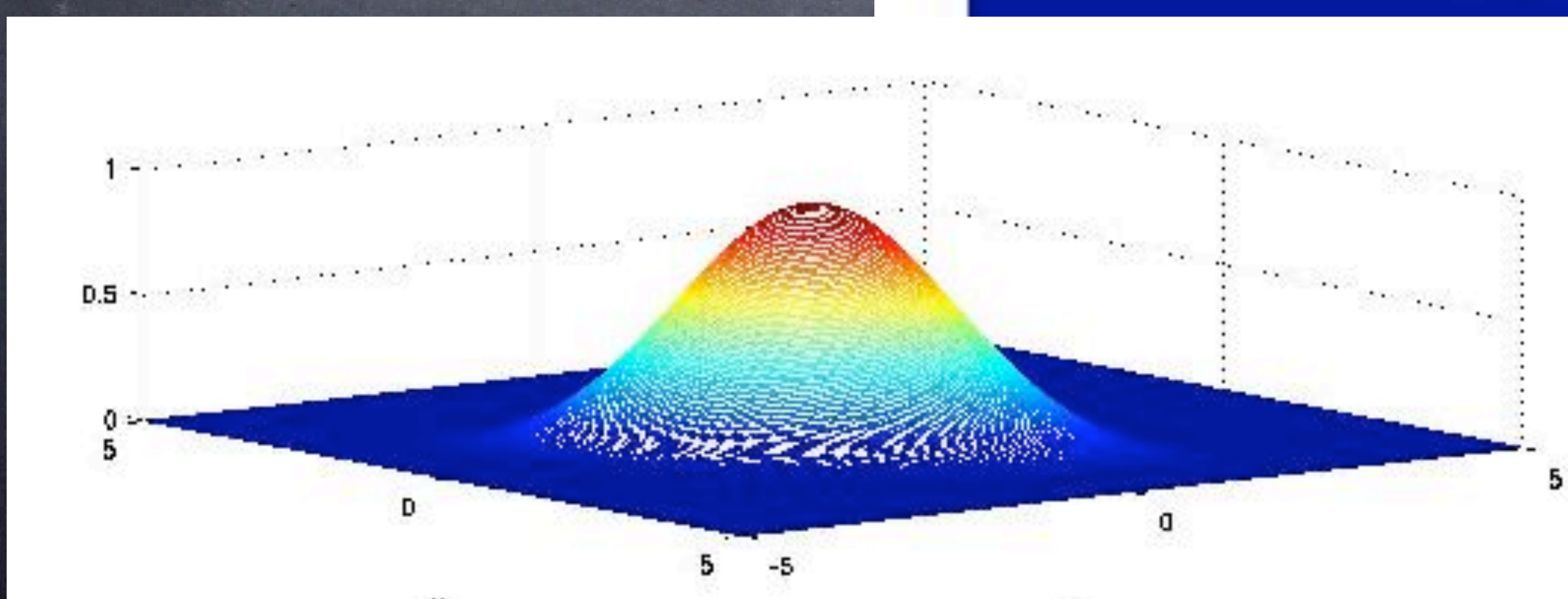
- variance: expected difference from mean, squared

evaluate along x

Multivariate Gaussian

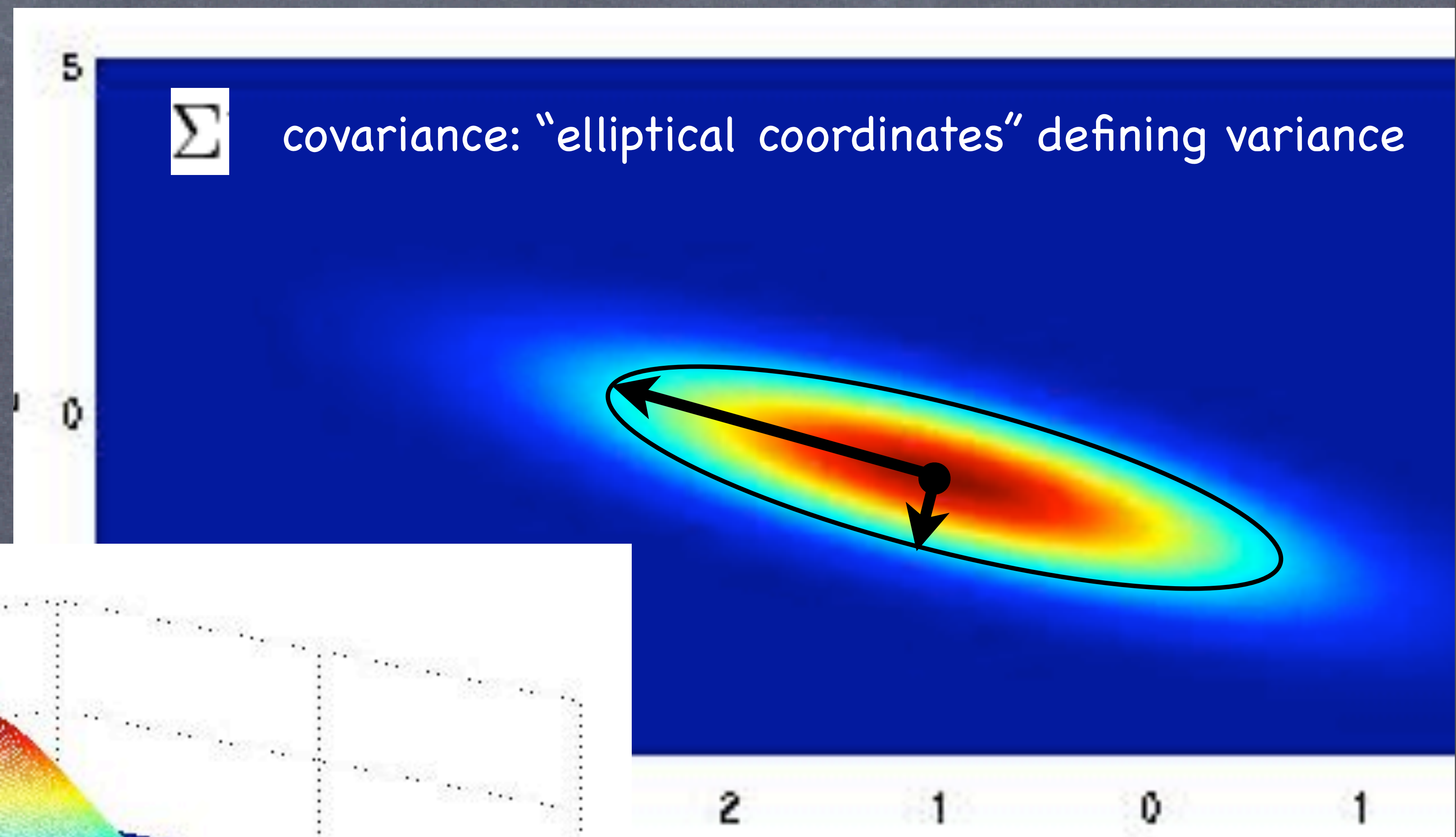
$$\frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}[x-\mu]^\top\Sigma^{-1}[x-\mu]\right)$$

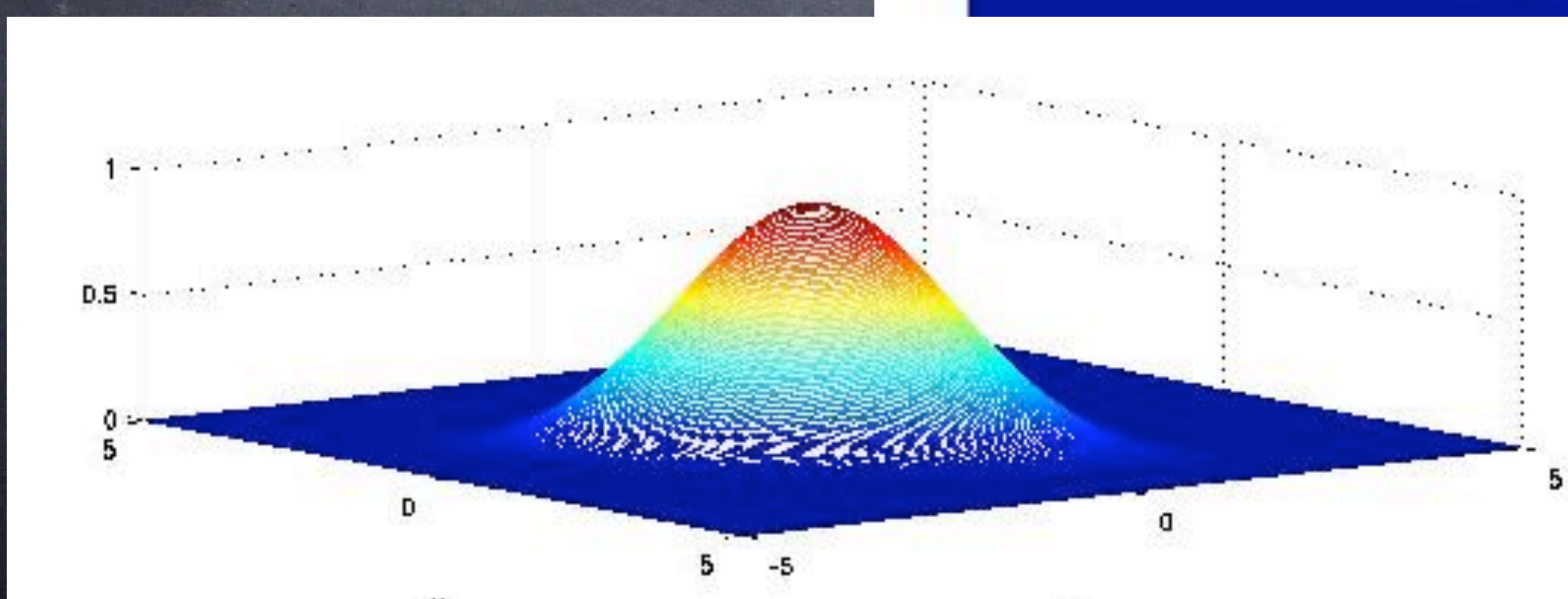




Multivariate Gaussian

$$\frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}[x-\mu]^\top\Sigma^{-1}[x-\mu]\right)$$





Kalman Filter

Predict

Predicted (*a priori*) state $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}$

Predicted (*a priori*) estimate covariance $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_{k-1}$

Update

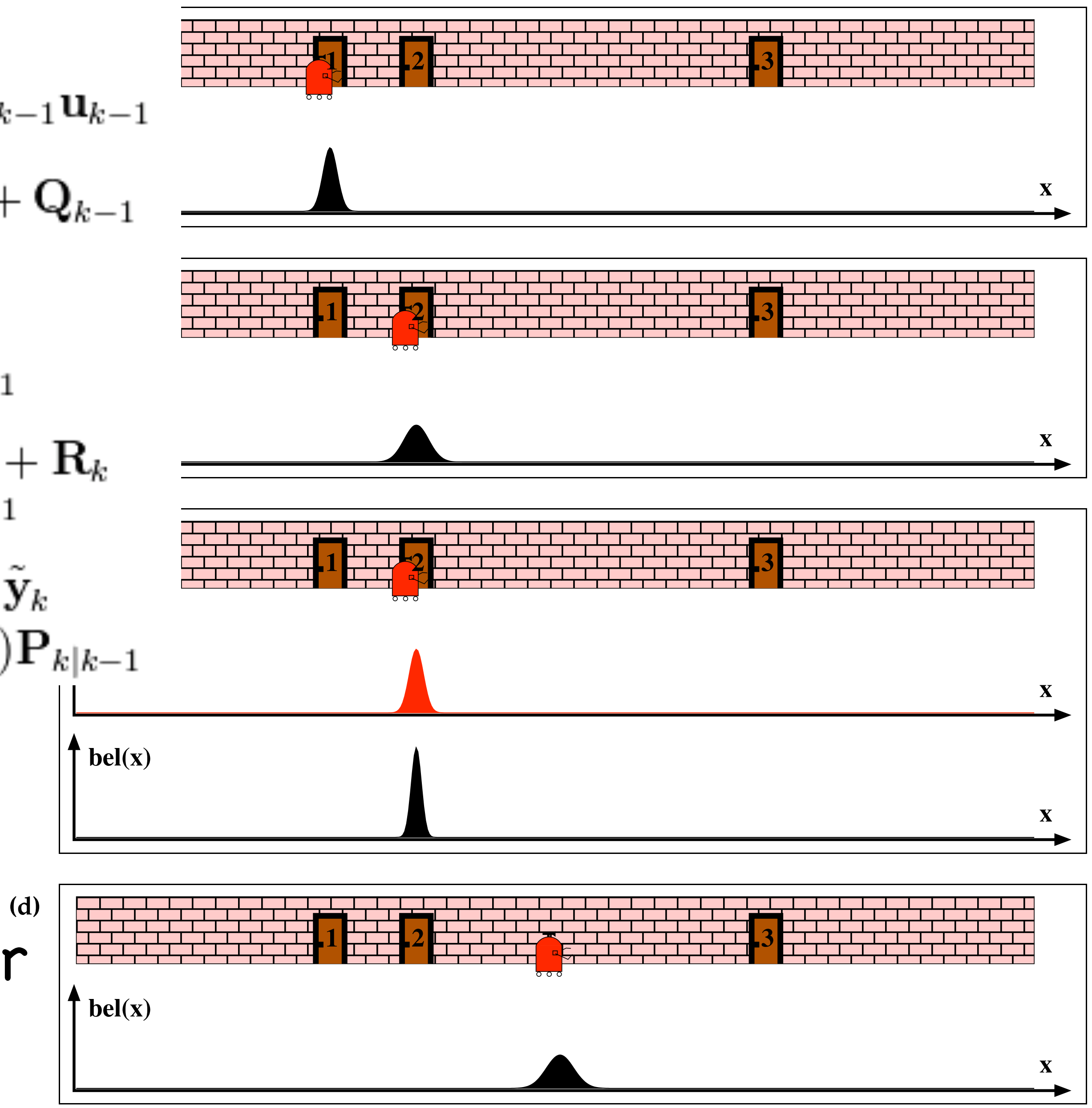
Innovation or measurement residual $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

Innovation (or residual) covariance $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$

Optimal Kalman gain $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$

Updated (*a posteriori*) state estimate $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

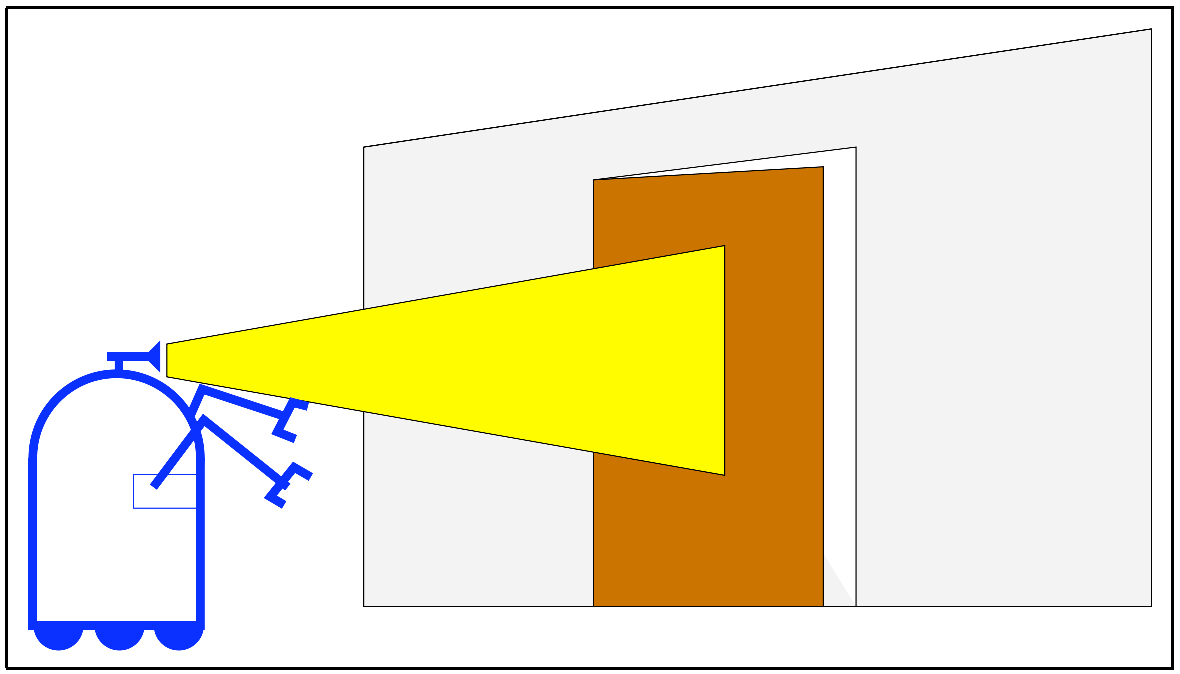
Updated (*a posteriori*) estimate covariance $\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}$



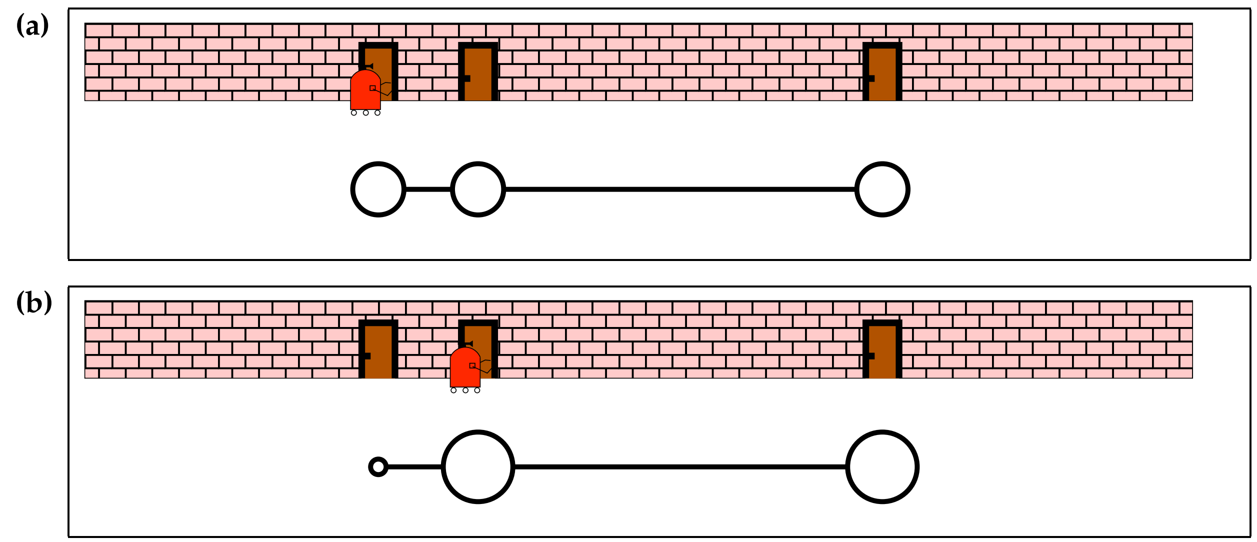
Evolution of unimodal Gaussian-based Kalman Filter

variance of mode corresponds to uncertainty

Filtering with Topological Map

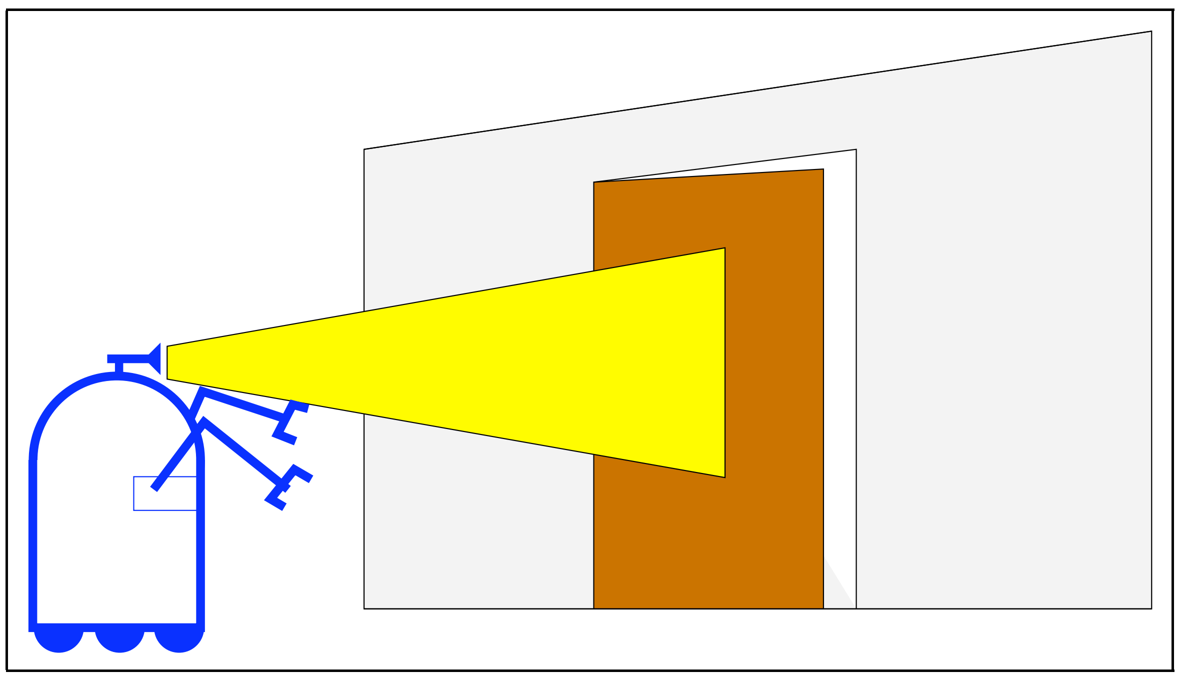


Robot can sense "door" or "wall" at location

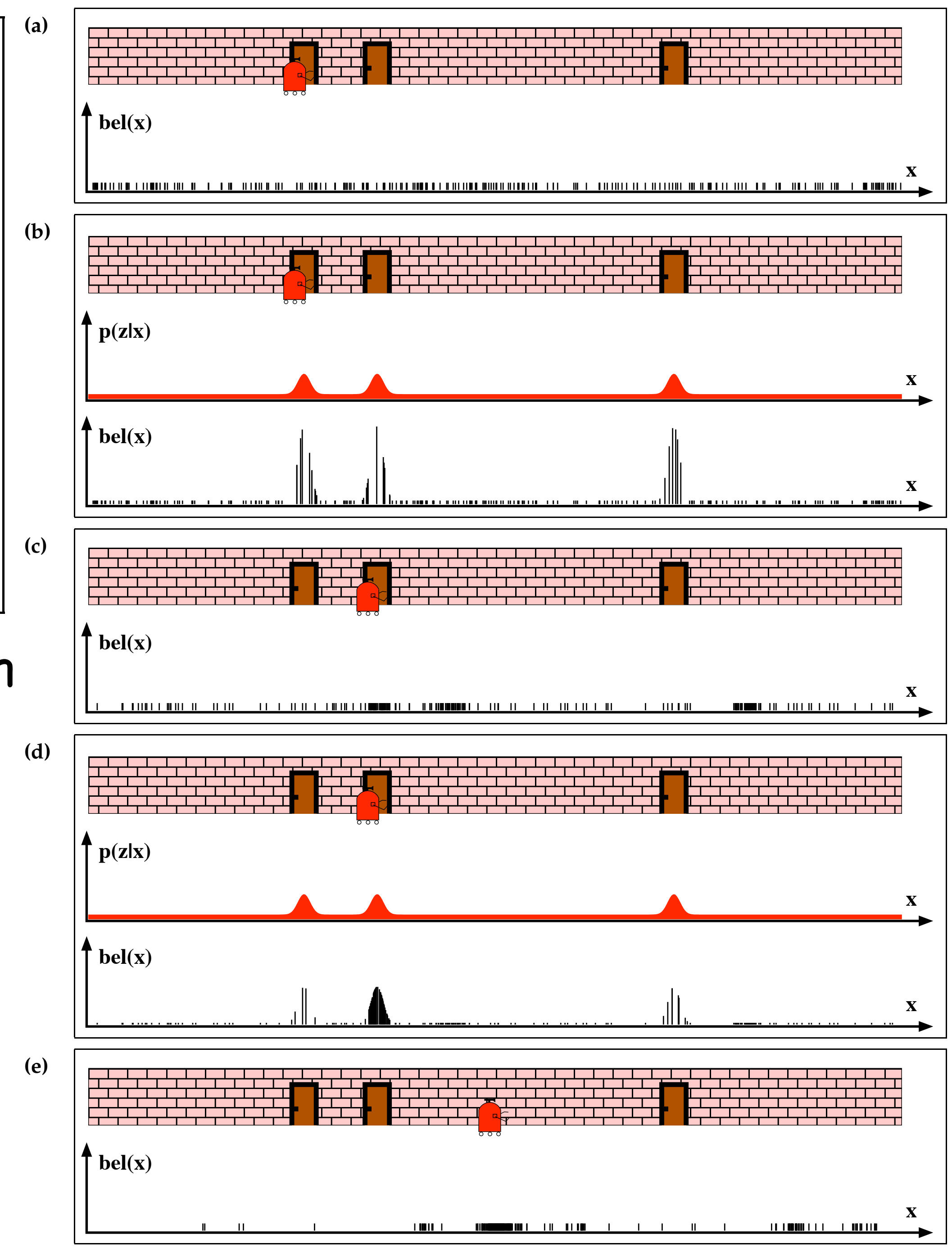


Evolution of discrete topological filter

Particle Filter

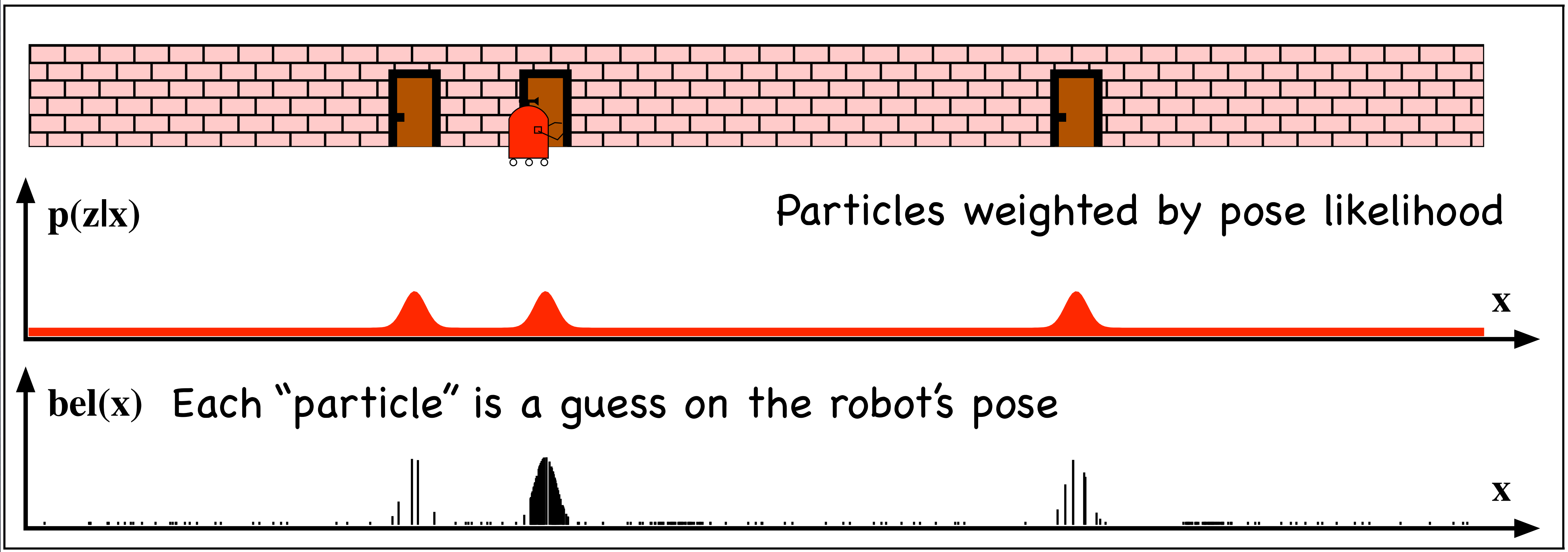


Robot can sense “door” or “wall” at location

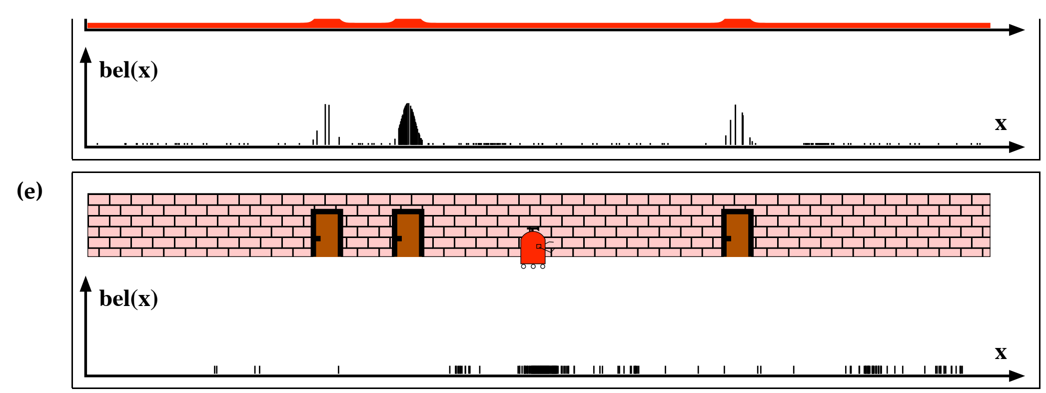


Evolution of multi-hypothesis particle filter

Particle Filter



Evolution of multi-hypothesis particle filter



Issues and soccer example

- How to get an estimate from a distribution?
- Will the distribution converge?
- How to recover if there is an incorrect convergence?
- More next topic