ČVUT
v Praze
in Prague

Simultaneous localization and mapping

Václav Hlaváč

Czech Technical University in Prague (ČVUT)

Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC)

Prague 6, Jugoslávských partyzánů 3
Czech Republic

hlavac@ciirc.cvut.cz
http://people.ciirc.cvut.cz/hlavac/

Courtesy to several authors of presentations on the web, especially P. Abbeel, R. Siegwart.

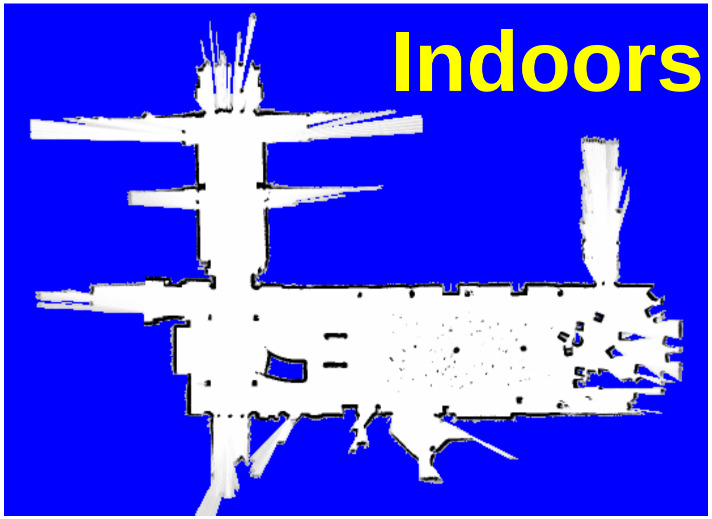
SLAM = Simulataneus Localization and Mapping

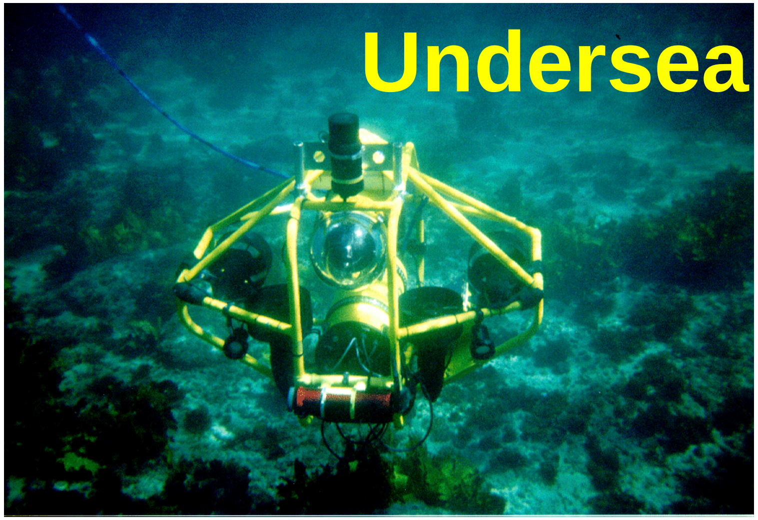
ČVUT
v Praze
in Prague

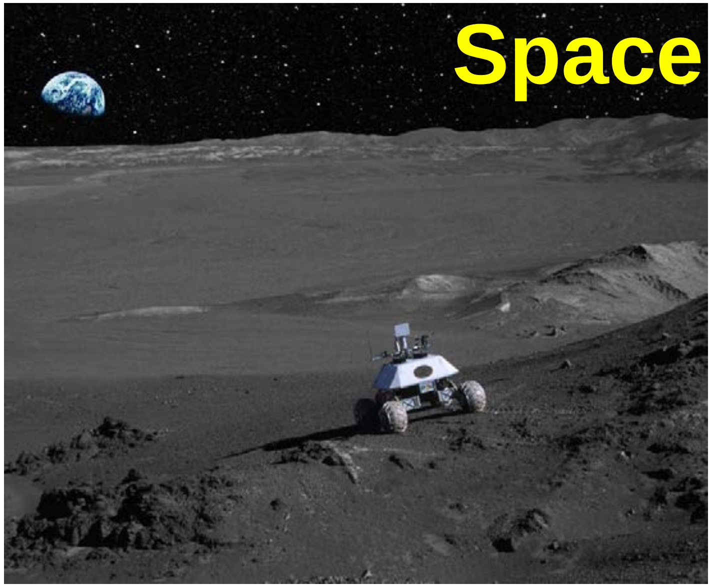
- SLAM task:
  Robot navigation in a previously unknown (static) environment while building and updating a map of its workspace continuously using on-board sensors only.
- When is SLAM necessary?
  - When a robot must be truly autonomous (no human input).
  - When there is no prior knowledge about the environment.
  - When we cannot place beacons (also in GPS-denied environments).
  - When the robot needs to know where it is.
- SLAM keeps being a challenge in probabilistic robotics.

SLAM Applications

ČVUT
v Praze
in Prague







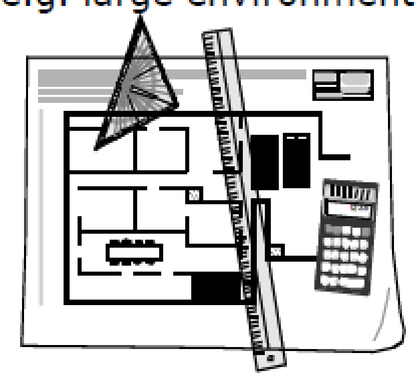
Simpler relevant tasks than SLAM

ČVUT
v Praze
in Prague

- **Pure localization**: the map is known and the location has to be estimated along the way.
- **Mapping with known poses**: the poses are known and the map is estimated along the way.

Mapping:

By hand: hard & costly
(e.g. large environment)



Automatic Map Building:

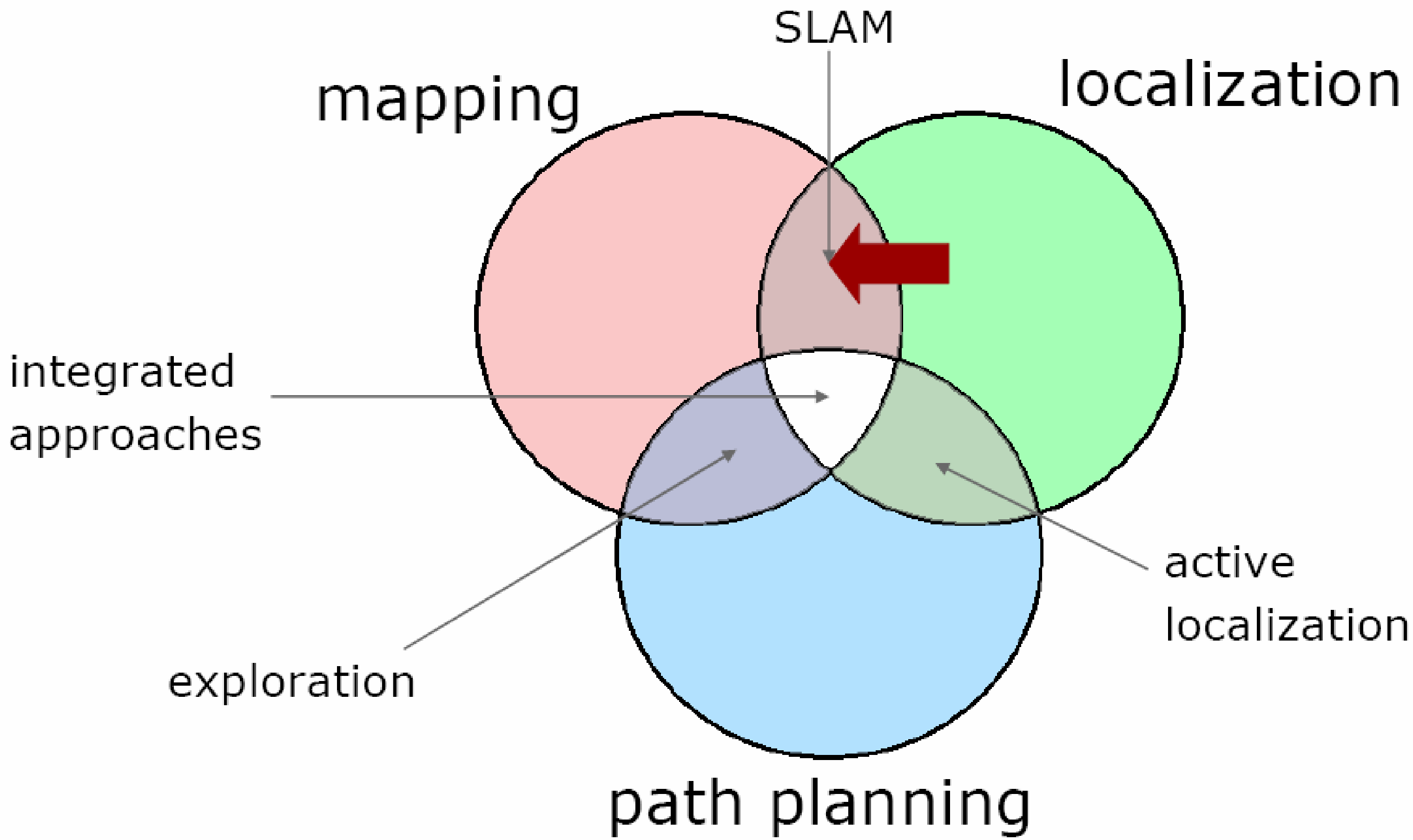
More challenging, but:

- ✓ Automatic
- ✓ The robot learns its environment
- ✓ Can adapt to dynamic changes

Courtesy: R. Siegwart

Where does SLAM fit?

ČVUT
v Praze
in Prague



SLAM – task formulation

ČVUT
v Praze
in Prague

- **Inputs**:
  - Time sequence of proprioceptive and exteroceptive measurements made as robot moves through an initially unknown environment.
    - The robot controls.
    - Observations of nearby features.
  - No external coordinate reference.
- **Outputs**:
  - Localization: A robot pose estimate associated with each measurement in the coordinate system of the map.
  - Mapping: An update to the map of the robot environment.
  - Path of the robot.

SLAM is an incremental task

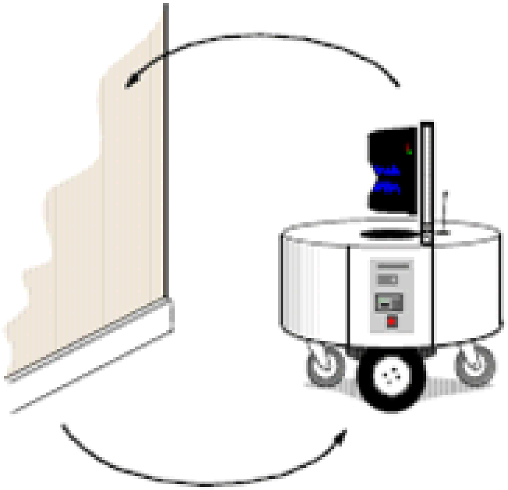
ČVUT
v Praze
in Prague

- **State/Output**:
  - Map of the environment, which has been observed so far.
  - Robot pose estimate with respect to the map.
- **Action/Input**:
  - Move to a new position/orientation.
  - Acquire additional observations.
- **Update state**:
  - Re-estimate robot pose.
  - Revise the map appropriately.

- Errors come from inaccurate measurement of actual robot motion (noisy action) and the distance from obstacle/landmark (noisy observation).
- Small errors will quickly accumulate over time steps.

SLAM difficulties (1)

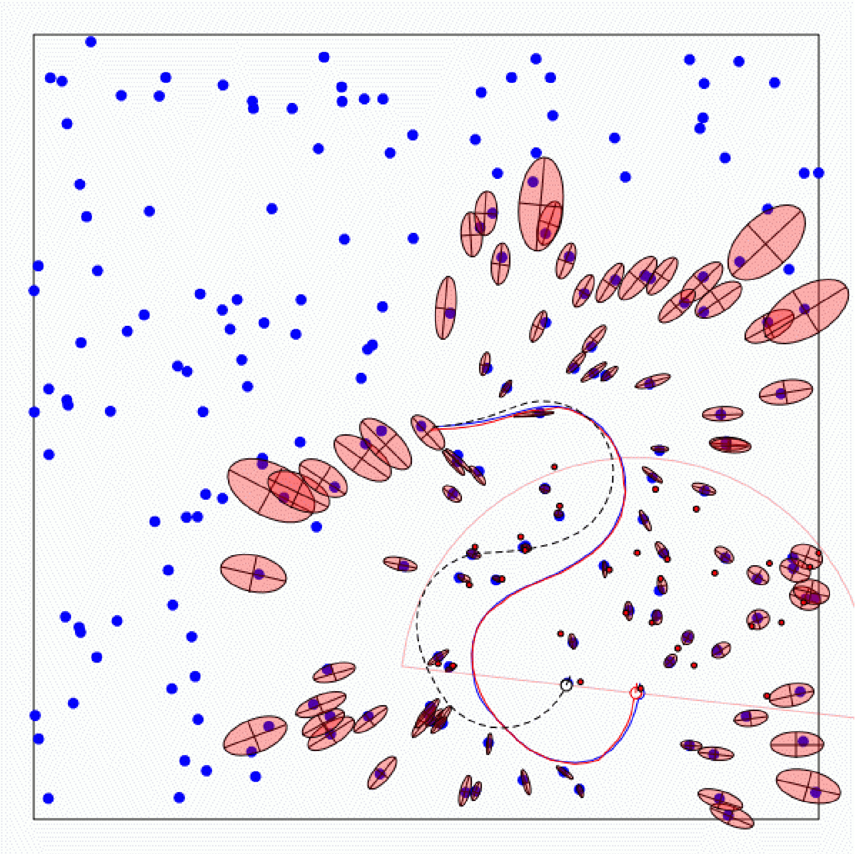
- SLAM is a chicken or egg problem.
  - A map is needed for localizing a robot.
  - A good robot position estimate is needed to create/update the map.
- Consequently, SLAM is regarded as hard problem in robotics.



SLAM difficulties (2)

ČVUT
v Praze
in Prague

- SLAM is considered one of the most fundamental problems for (mobile) robots to be truly autonomous.
- Variety of approaches have been tried to approach SLAM problem.
- Probabilistic methods rule!
- History of SLAM dates to mid-1980s.



Why is SLAM hard?

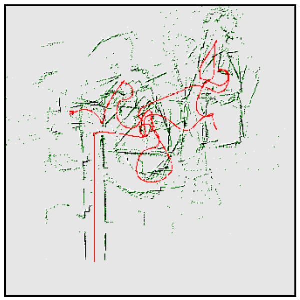
ČVUT
v Praze
in Prague

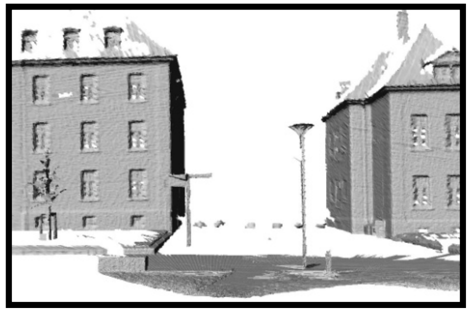
- Uncertainty at every level of the problem.
- Many ingredients:
  - Autonomous, persistent, collaborative robots.
  - Mapping is multi-scale in generic environments.
- Map-making $\sim$ learning:
  - Difficult also for humans.
  - Humans make mapping mistakes.
- Scaling issues:
  - Large spatial extent $\Rightarrow$ combinatorial expansion.
  - Persistent autonomous operations.

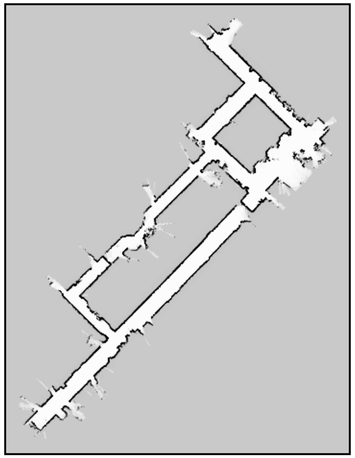
Robot world representations

ČVUT
v Praze
in Prague

- Grid maps or scans

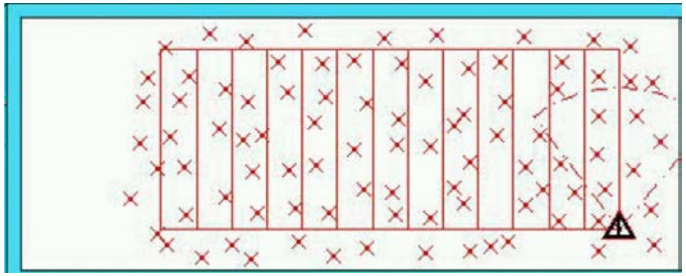


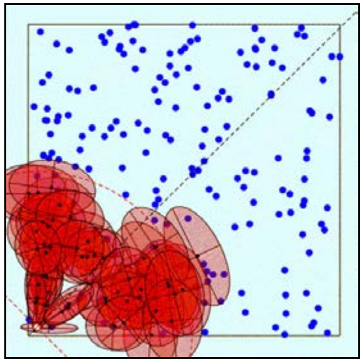


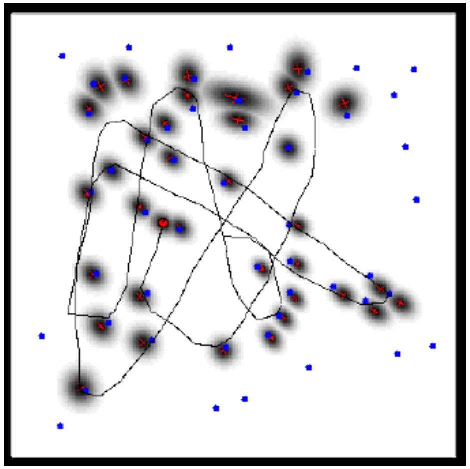


[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

- Landmark-based



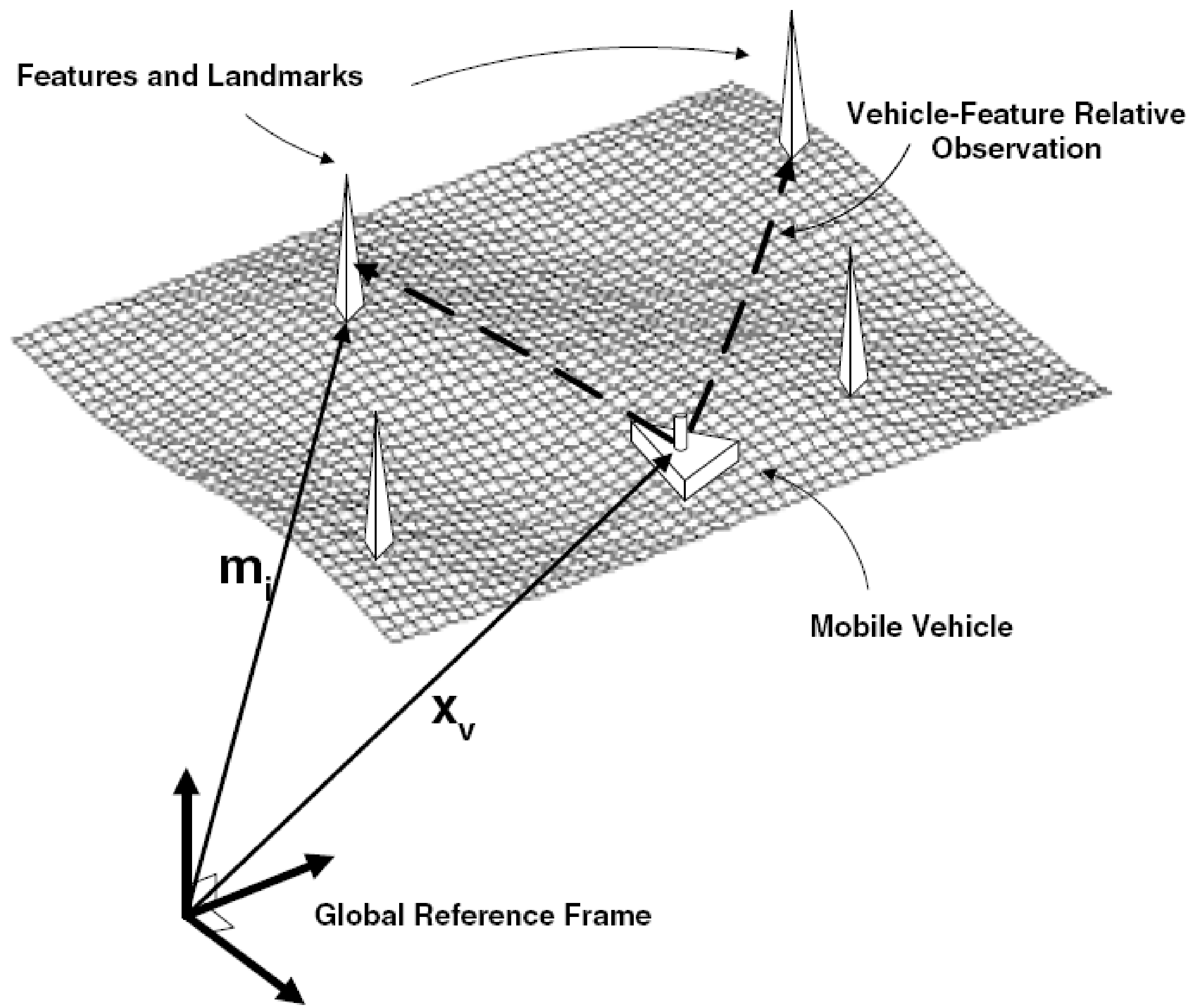




[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;…

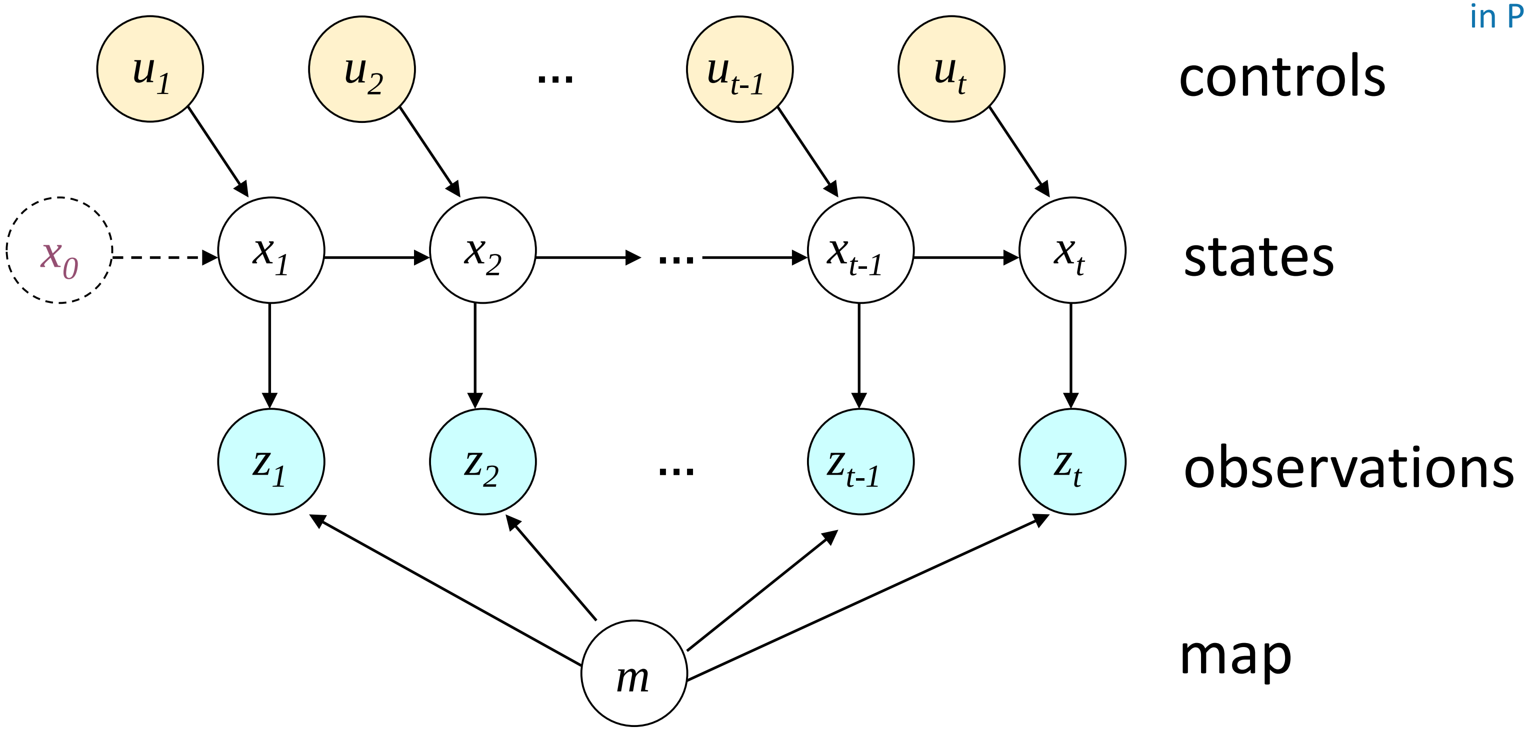
Structure of the Landmark-based SLAM task

ČVUT
v Praze
in Prague



Markovian assumption

ČVUT
v Praze
in Prague

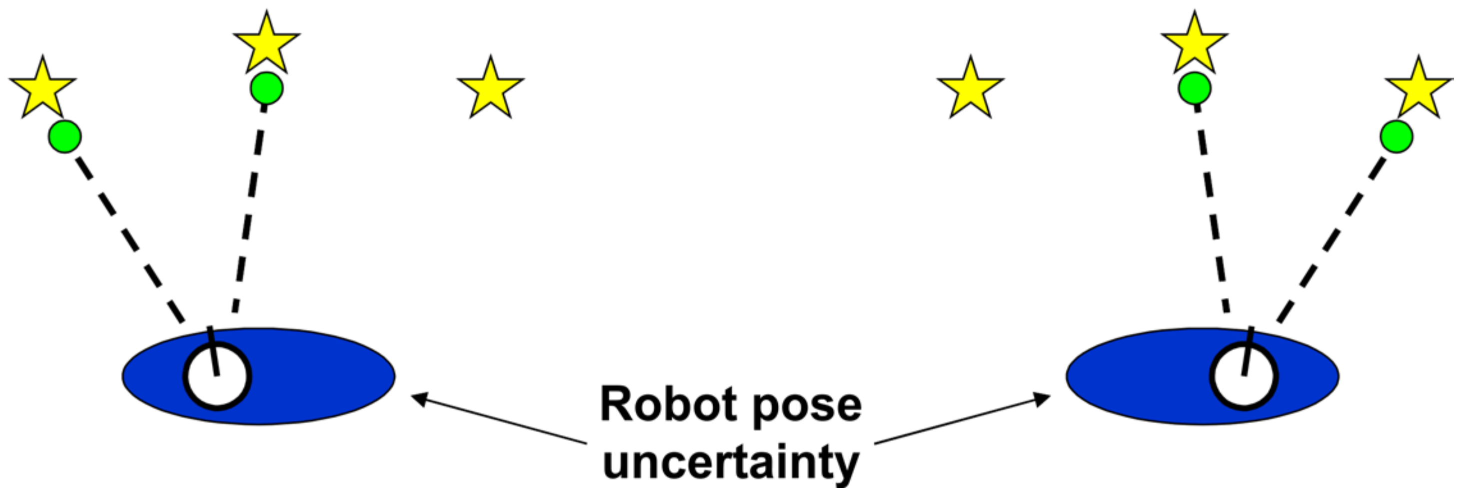


State transition: $p(x_t \mid x_{t-1}, u_t)$

Observation function: $p(z_t \mid x_t)$

Why is SLAM a hard problem?

ČVUT
v Praze
in Prague

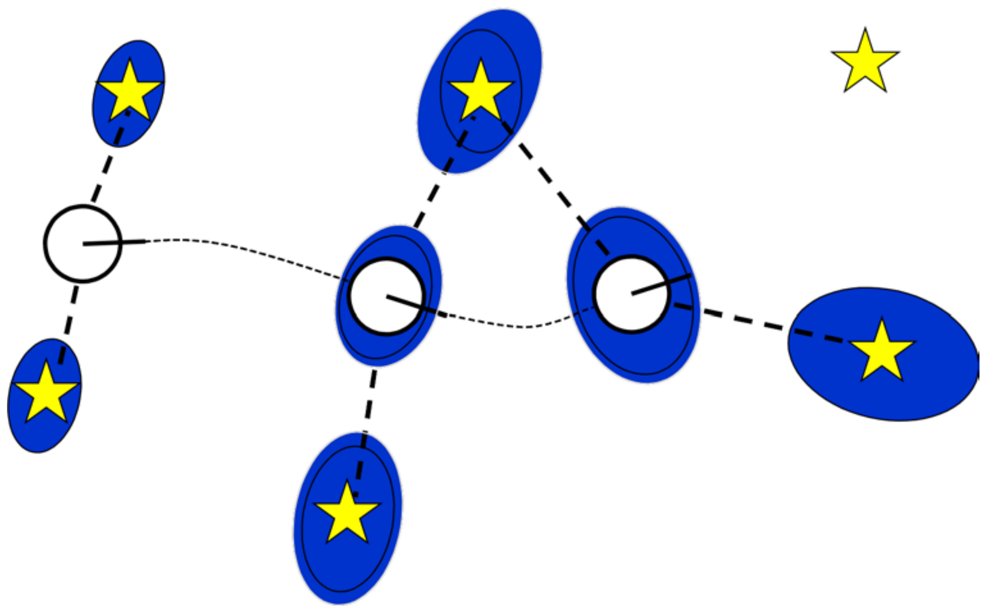


- In the real world, the mapping between observations and landmarks is unknown.
- Picking wrong data associations can have catastrophic consequences.
- Pose error correlates data associations.

Why is SLAM a hard problem?

ČVUT
v Praze
in Prague

SLAM: robot path and map are both **unknown**.



Robot path error correlates errors in the map.

SLAM

ČVUT
v Praze
in Prague

- Full SLAM:

Estimates the entire path and map!

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

- Online SLAM:

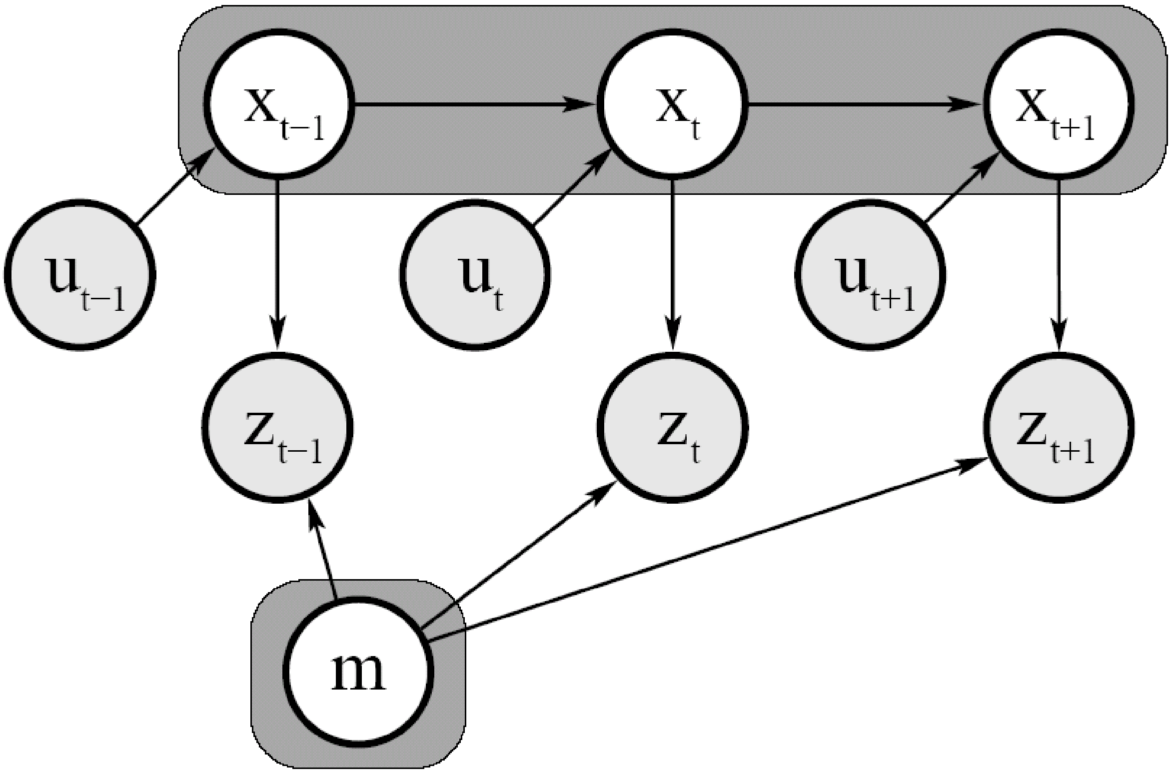
$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \dots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 dx_2 \dots dx_{t-1}$$

Integrations typically done one at a time

Estimates the most recent pose and map!

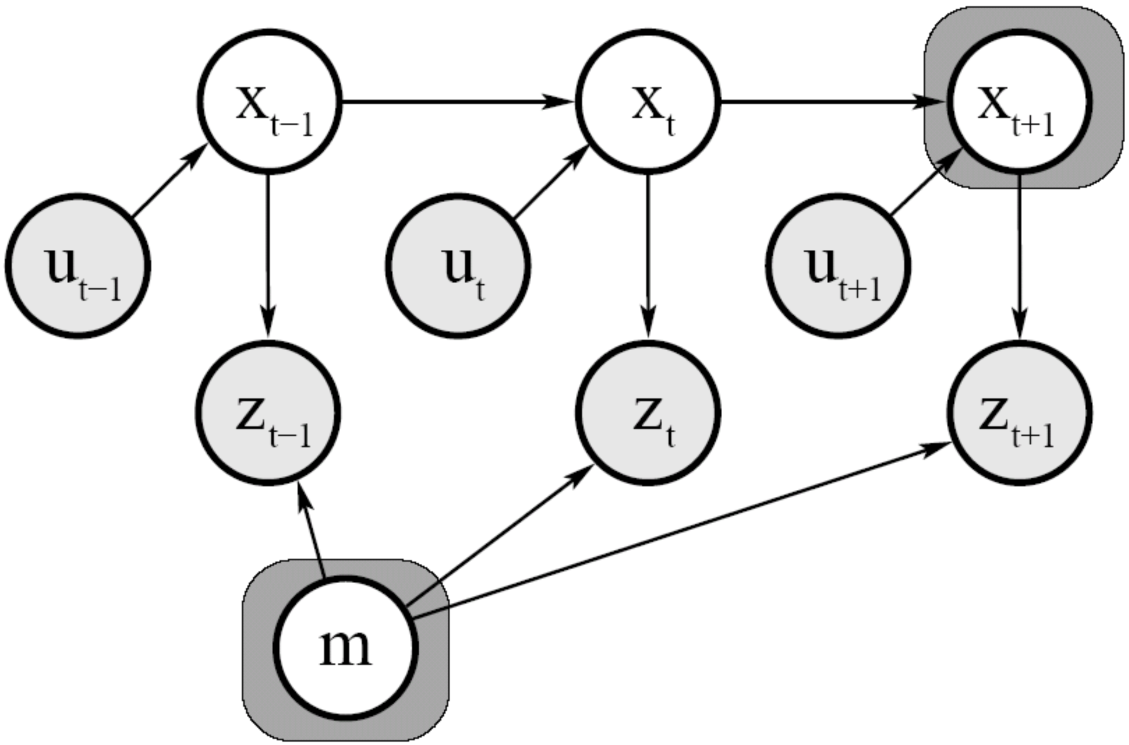
Graphical model of Full SLAM

ČVUT
v Praze
in Prague



$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

Graphical model of online SLAM



$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1\, dx_2 \ldots dx_{t-1}$$

Techniques for generating consistent maps

ČVUT
v Praze
in Prague

1. Scan matching = Given a scan and a map (or scan-scan, map-map), find the rigid transformation that aligns them the best. Two approaches.
   - Optimize over x: $p(z|x,m)$
   - Reduce scan-map to a point cloud and run the Iterative Point Cloud algorithm (ICP).
2. Parametric method – (Extended) Kalman filter
   - Represent the distribution of the robot location x_t(and map m_t) by a Gaussian distribution.
   - Update μ_t and $\sum_t$ sequentially.
3. Sample-based method – Particle filter
   - Represent the distribution of robot location x_t (and map m_t) by a large amount of simulated samples.
   - Resample x_t (and m_t) at each time step.

Scan Matching, optimal *x*: $p(z|x,m)$

ČVUT
v Praze
in Prague

Maximize the likelihood of the pose *t* and map relative to the pose *t-1* and to the map.

$$\hat{x}_t = \arg\max_{x_t} \left\{ p(z_t \mid x_t, \hat{m}^{[t-1]}) \cdot p(x_t \mid u_{t-1}, \hat{x}_{t-1}) \right\}$$

current measurement

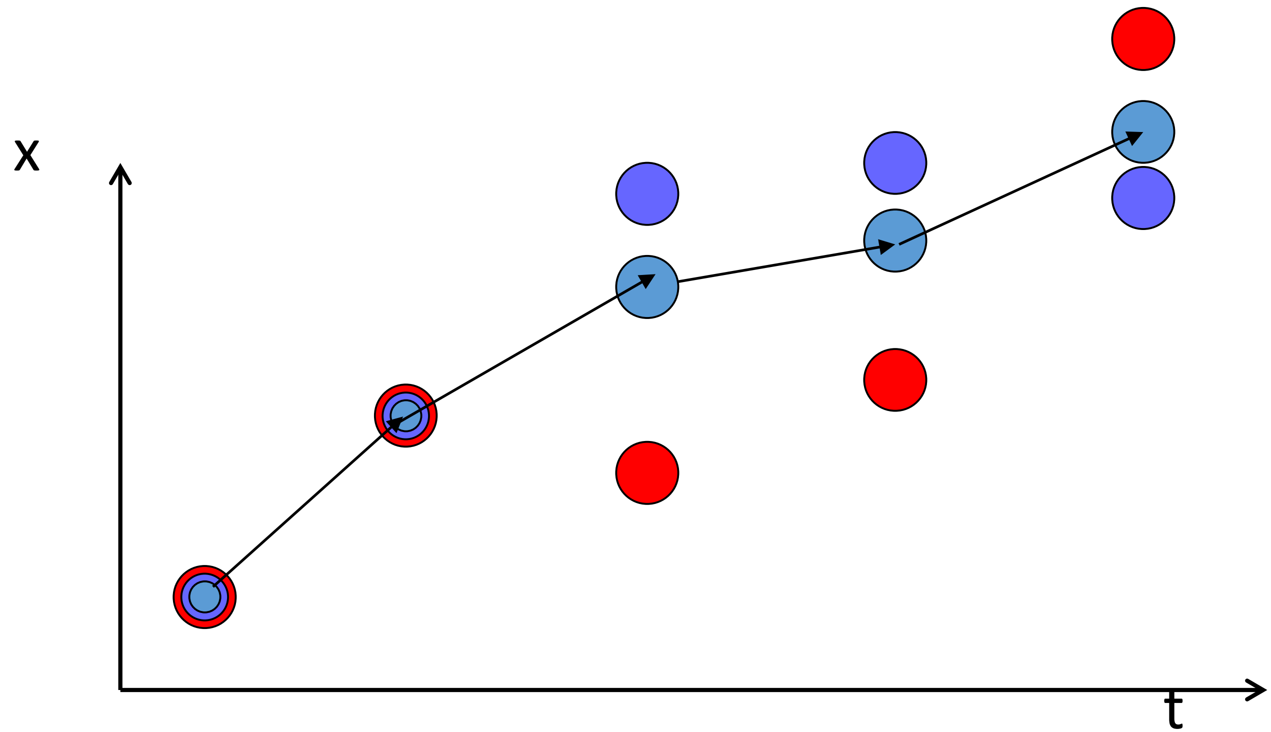
robot position

map constructed so far

Calculate the map $\hat{m}^{[t]}$ according to “mapping with known poses” based on the poses and observations.

General filter- example

ČVUT
v Praze
in Prague



Measurement (Observation)

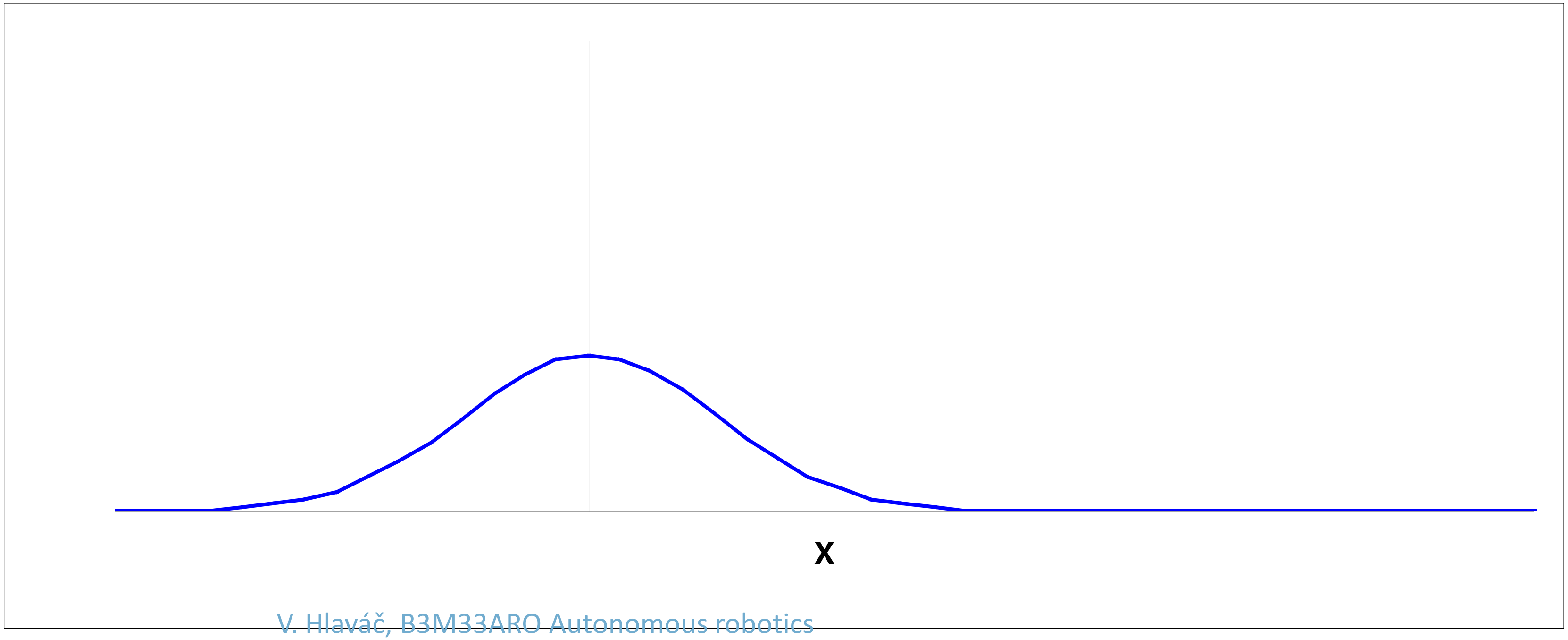
Prediction

Estimation

The Kalman filter, a simple example Where are you (1)

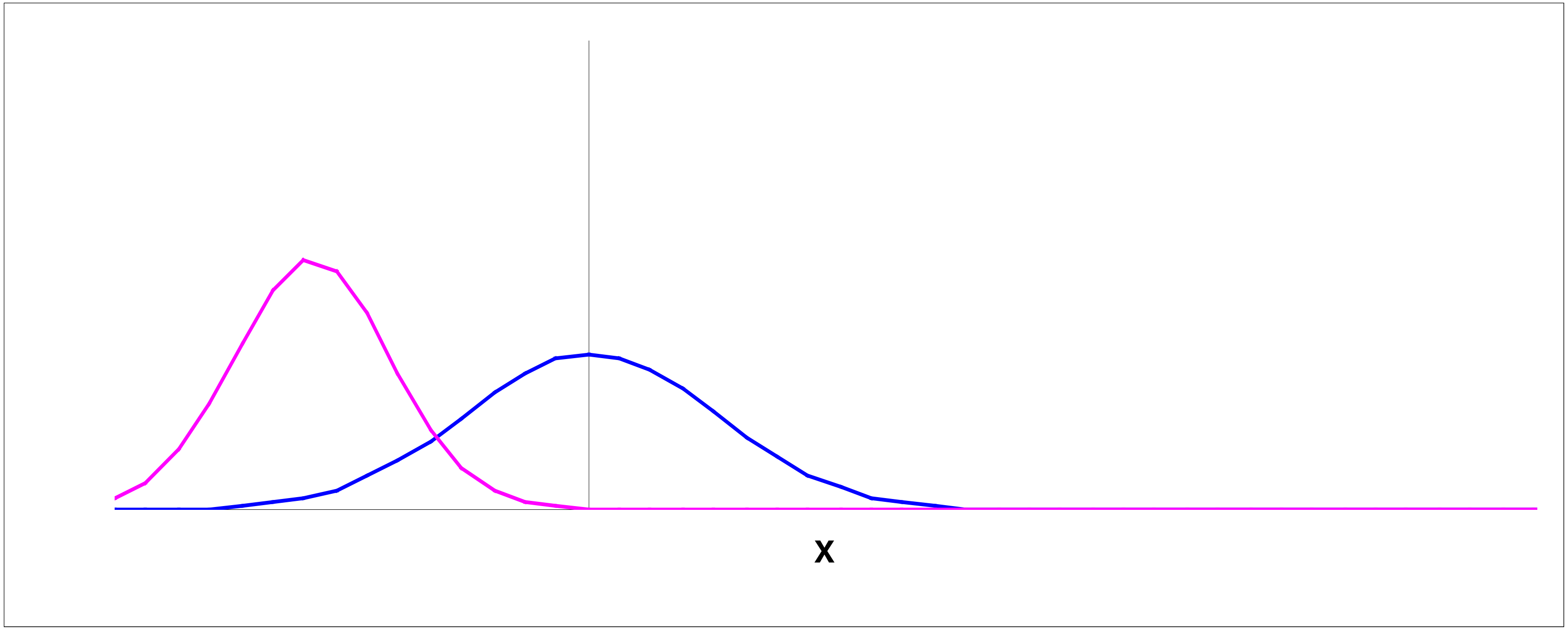
ČVUT
v Praze
in Prague

- Where are you (in 1D)?
- [from star sighting]: at time t_1, you are in z_1, with accuracy σ_{z1}
- Best estimate: $x(t_1)=z_1$
- Variance of the error: $(\sigma_x(t_1))^2 = (\sigma_{z1})^2$



Where are you (2)

- [from GPS]: at time $t_2 \simeq t_1$, you are in z_2, with the accuracy σ_{z2}



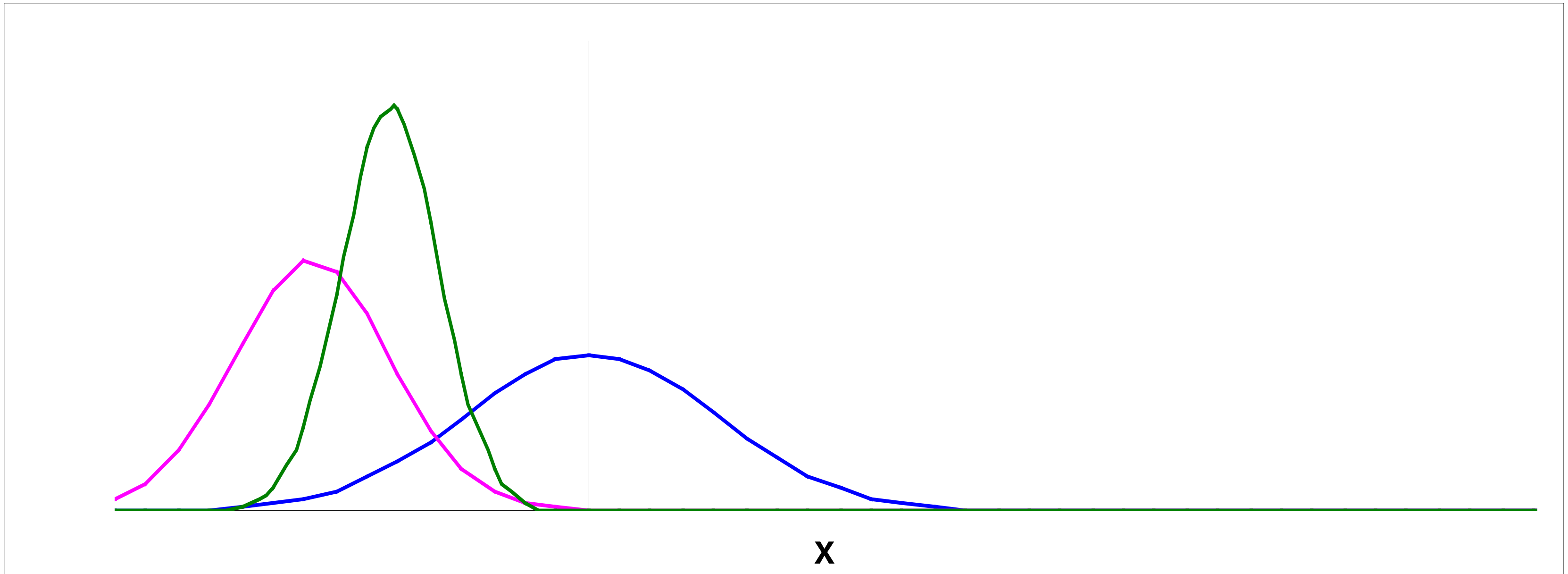
How should we combine the information?

- The best estimate:

$$X(t_2) = [(\sigma_{z2}^{\,2})/(\sigma_{z1}^{\,2}+\sigma_{z2}^{\,2})]\, z_1 + [(\sigma_{z1}^{\,2})/(\sigma_{z1}^{\,2}+\sigma_{z2}^{\,2})]\, z_2$$

- Variance of the error:

$$1/(\sigma)^2 = 1/(\sigma_{z1})^2 + 1/(\sigma_{z2})^2$$



Kalman filter update equations

ČVUT
v Praze
in Prague

Predictor — Corrector — Kalman gain

$$x(t_2) = x(t_1) + K(t_2)\,(z_2 - x(t_1))$$

$$\sigma_x(t_2)^2 = \sigma_x(t_1)^2 - K(t_2)\,\sigma_x(t_1)^2$$

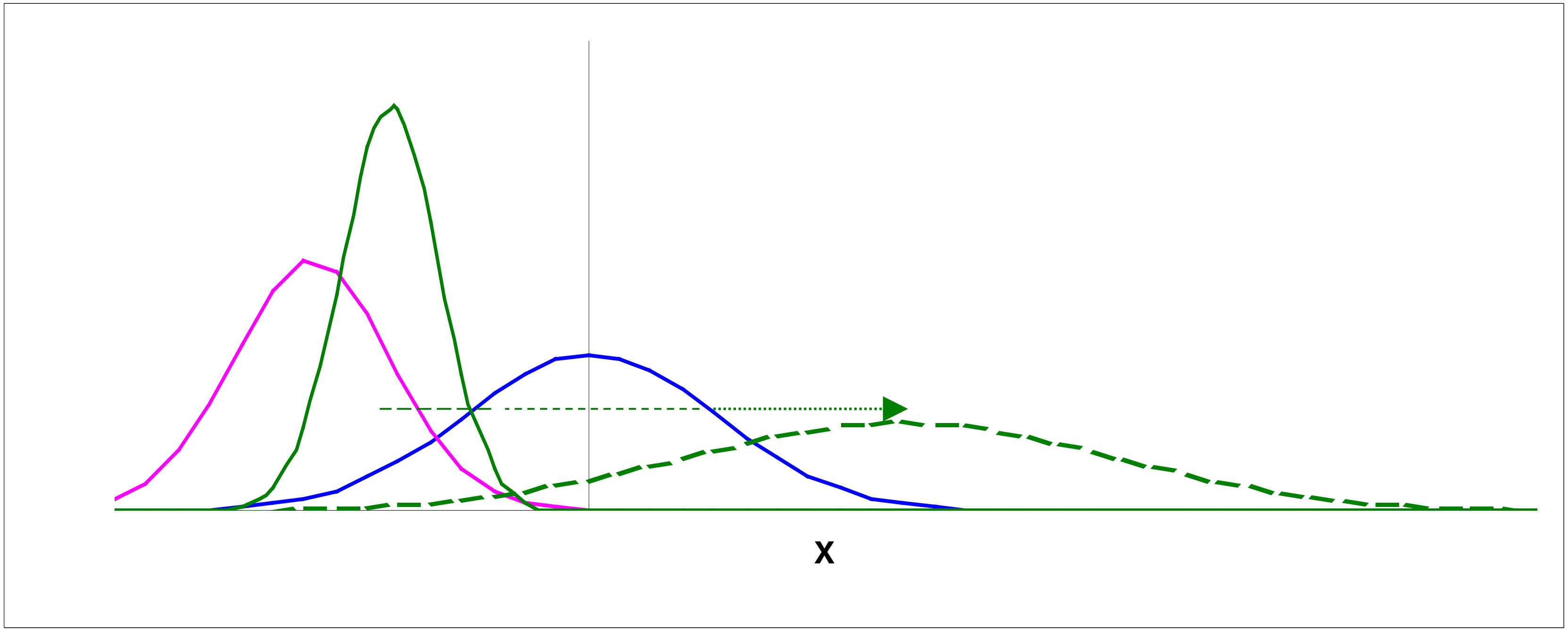
Where: $K(t_2) = (\sigma_{z1}^{\;2}) / (\sigma_{z1}^{\;2} + \sigma_{z2}^{\;2})$

And so on for the next measurement…

Moving objects

- $t_2 \cong t_1 \quad t_3 \rightarrow t_2$
- The motion equation:

$$dx/dt = u + w$$



$$x(t_3-) \;=\; x(t_2) \quad + \; u\,[t_3 - t_2]$$

$$\sigma_x(t_3-)^2 = \sigma_x(t_2)^2 \; - \sigma_w{}^2[t_3 - t_2]$$

ČVUT
v Praze
in Prague

Combining the information

ČVUT
v Praze
in Prague

Predictor   Corrector

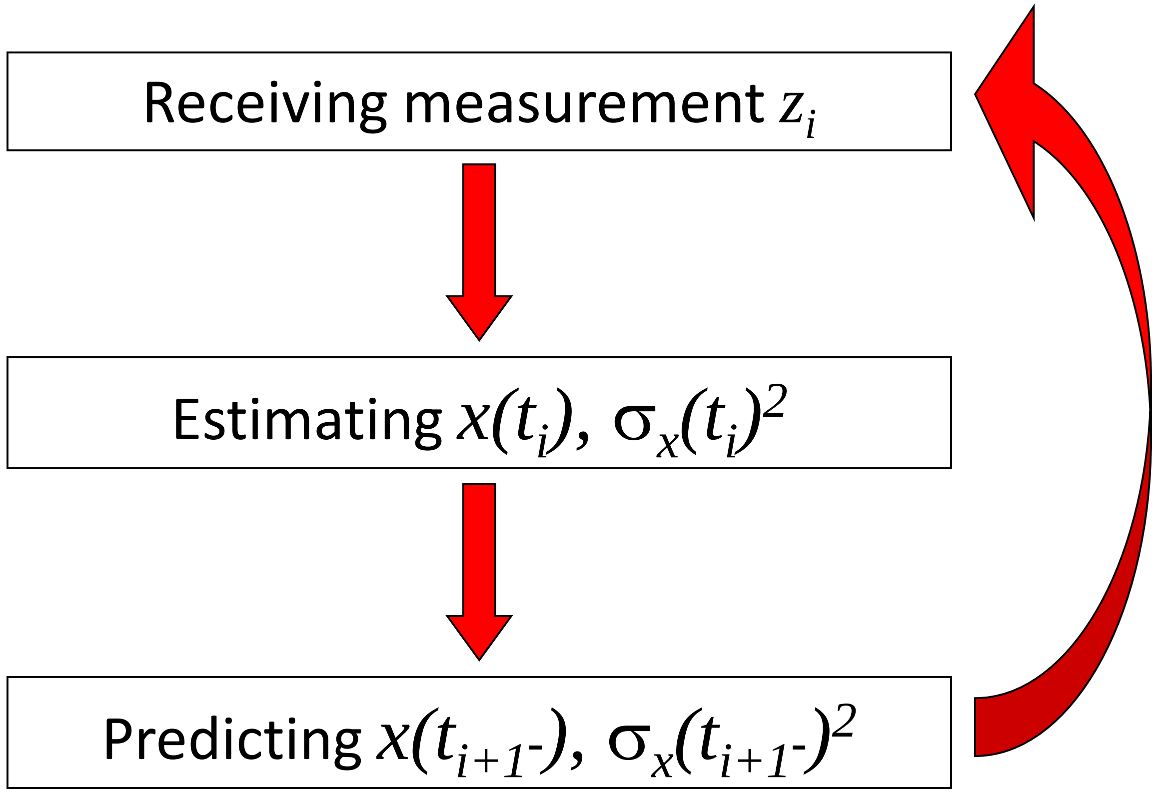
$$x(t_3) = x(t_{3-}) + K(t_3)/(z_3 - x(t_{3-}))$$

$$\sigma_x(t_3)^2 = \sigma_x(t_{3-})^2 - K(t_3)\,\sigma_x(t_{3-})^2$$

Where: $K(t_{3-}) = (\sigma_{t3-}^{\;2}) / (\sigma_{t3-}^{\;2} + \sigma_{z3}^{\;2})$

And so on for the next measurement…

General scheme



Kalman filter- a general model

ČVUT
v Praze
in Prague

- The Model:
  - Equations:
    - $X_t = AX_{t-1} + Bu_{t-1} + w_{t-1}$
    - $Z_t = HX_t + v_t$
  - w_t: model error ("Brownian motion"). Gaussian white noise.
  - v_t: measurment error. Gaussian white noise.
- The "best" estimator (in Minimal Mean Square Error sense) is:

$$\hat{X}_t = E[X_t \mid z_t, z_{t-1}, \ldots]$$

- Kalman filter gives the "best" estimation for the given model (linear, Gaussian noise).

Kalman Filter Algorithm

ČVUT
v Praze
in Prague

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

(E)KF-SLAM

ČVUT
v Praze
in Prague

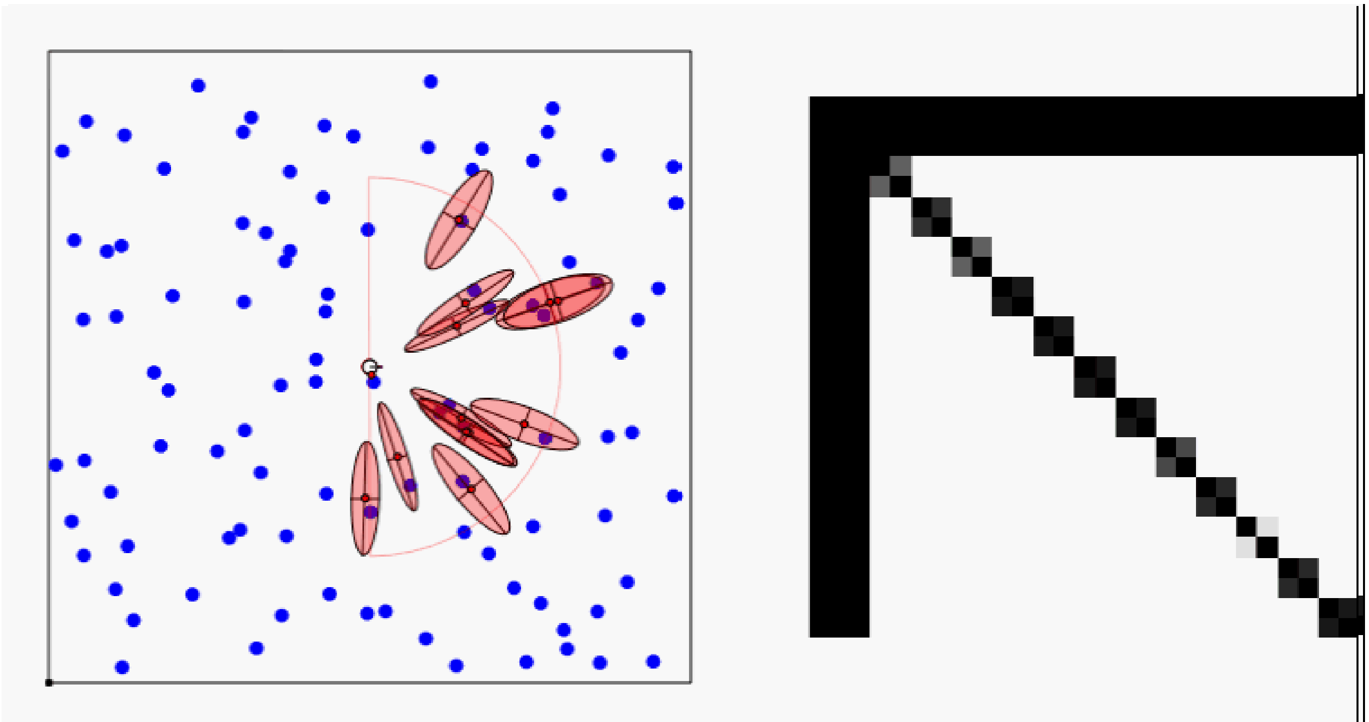
- Map with N landmarks:(3+2N)-dimensional Gaussian

$$Bel(x_t, m_t) = \left\langle \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N} & \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{pmatrix} \right\rangle$$

- Can handle hundreds of dimensions

Classical Solution – The EKF

ČVUT
v Praze
in Prague

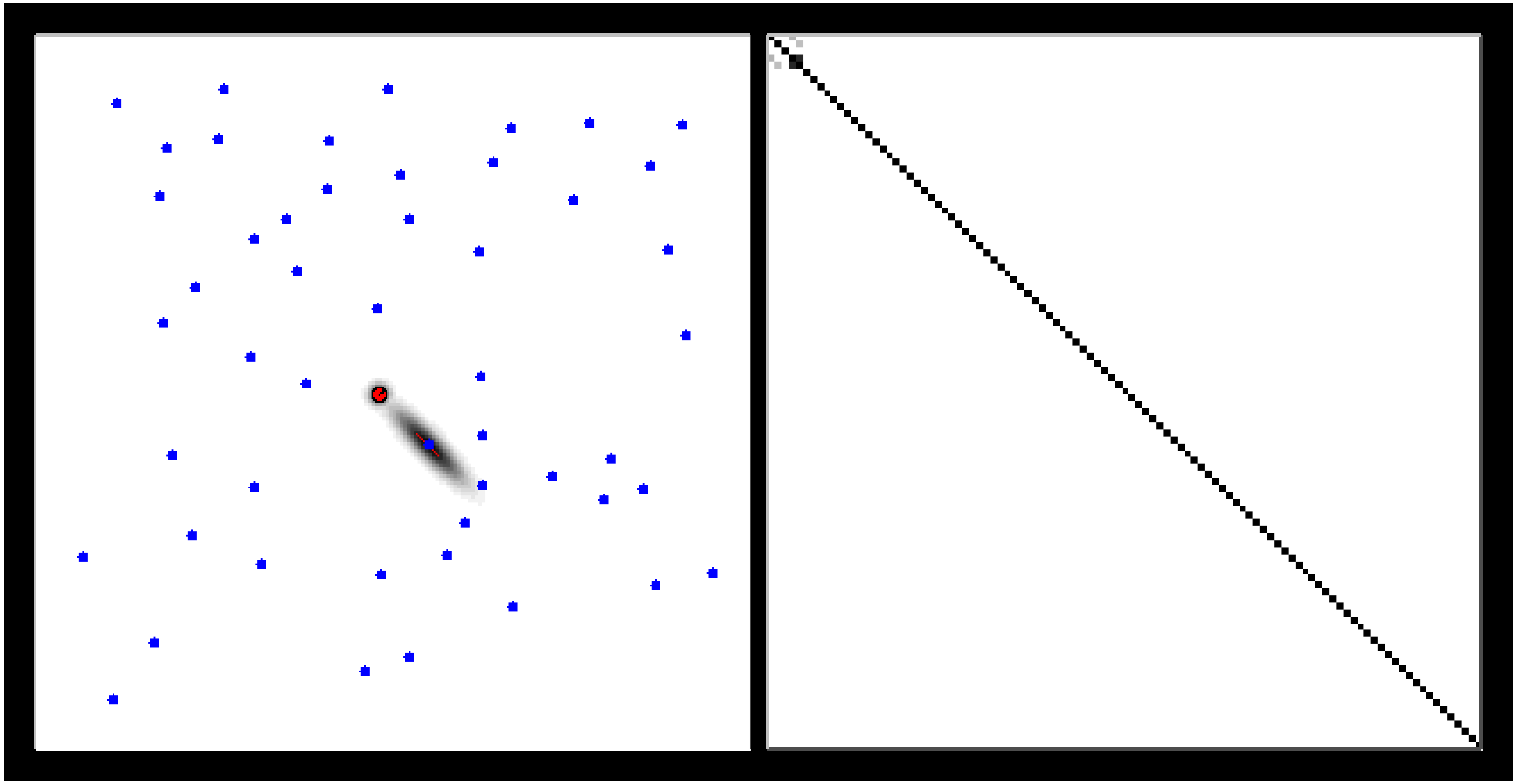


Blue path = true path   **Red path** = estimated path   **Black path** = odometry

- Approximate the SLAM posterior with a high-dimensional Gaussian [Smith & Cheesman, 1986] …
- Single hypothesis data association

EKF-SLAM

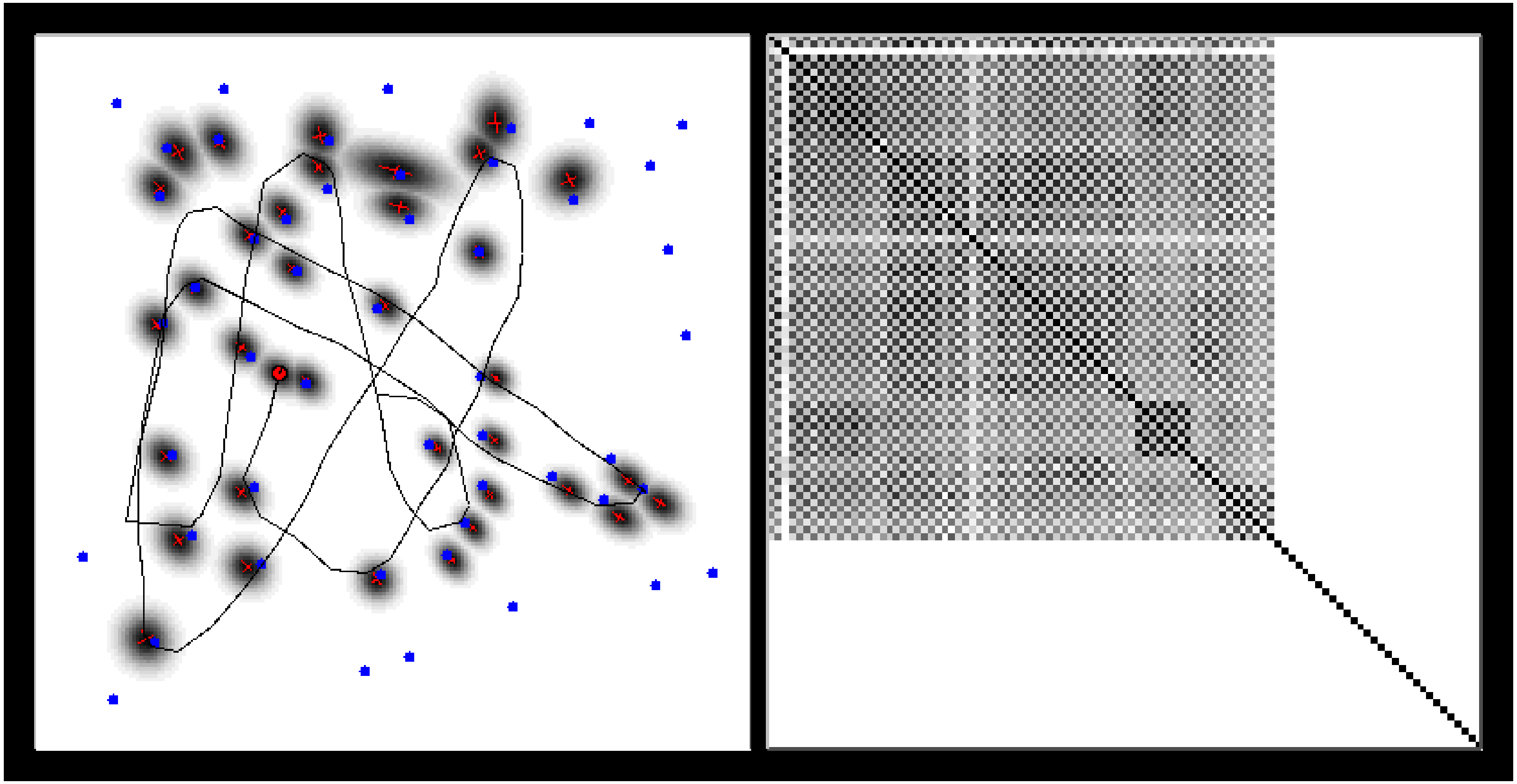
ČVUT
v Praze
in Prague



Map Correlation matrix

EKF-SLAM

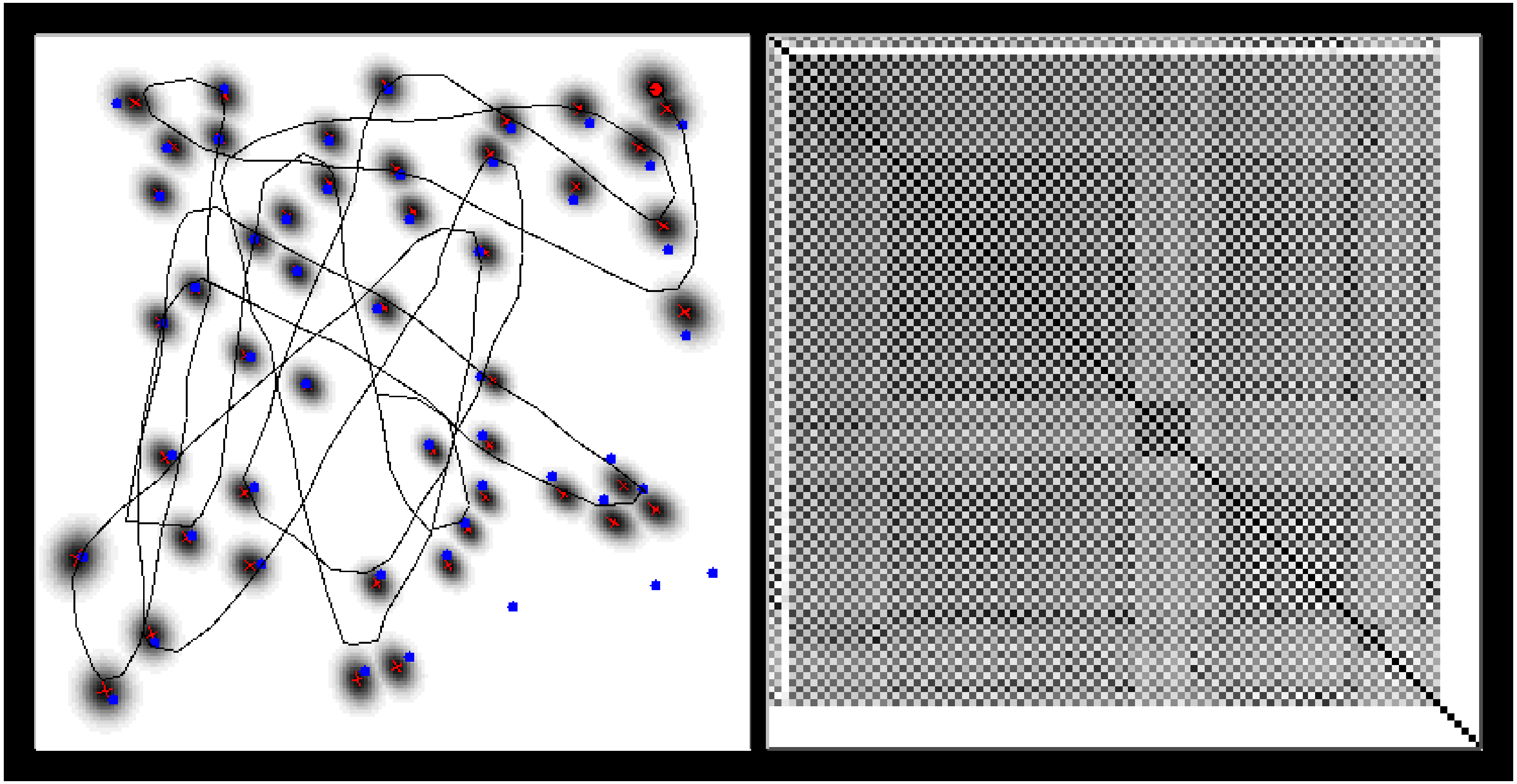
ČVUT
v Praze
in Prague



Map Correlation matrix

EKF-SLAM

ČVUT
v Praze
in Prague



Map Correlation matrix

Properties of KF-SLAM (Linear Case)

ČVUT
v Praze
in Prague

[Dissanayake et al., 2001]

Theorem:

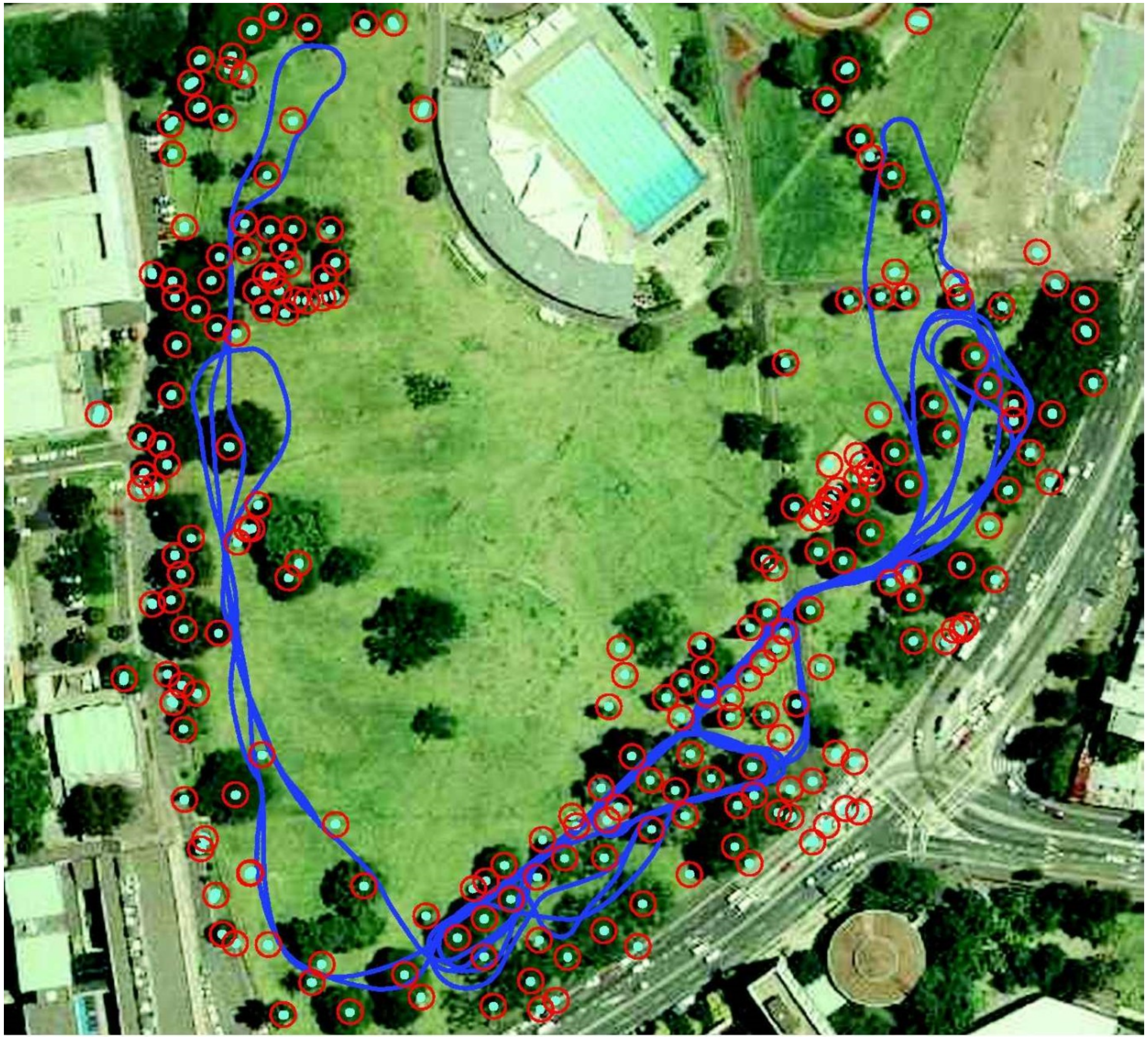
The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made.

Theorem:

In the limit the landmark estimates become fully correlated

Victoria Park data set

ČVUT
v Praze
in Prague



[courtesy E. Nebot]

Victoria Park Data Set Vehicle

ČVUT
v Praze
in Prague

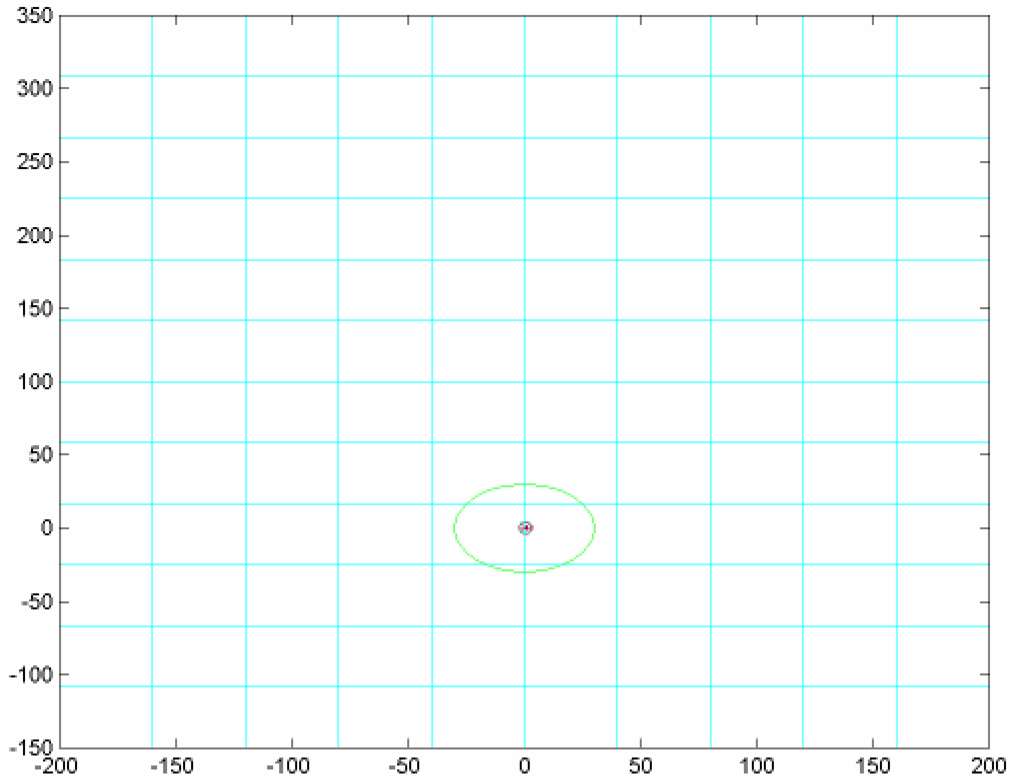
[courtesy E. Nebot]

Data acquisition

[courtesy E. Nebot]

SLAM

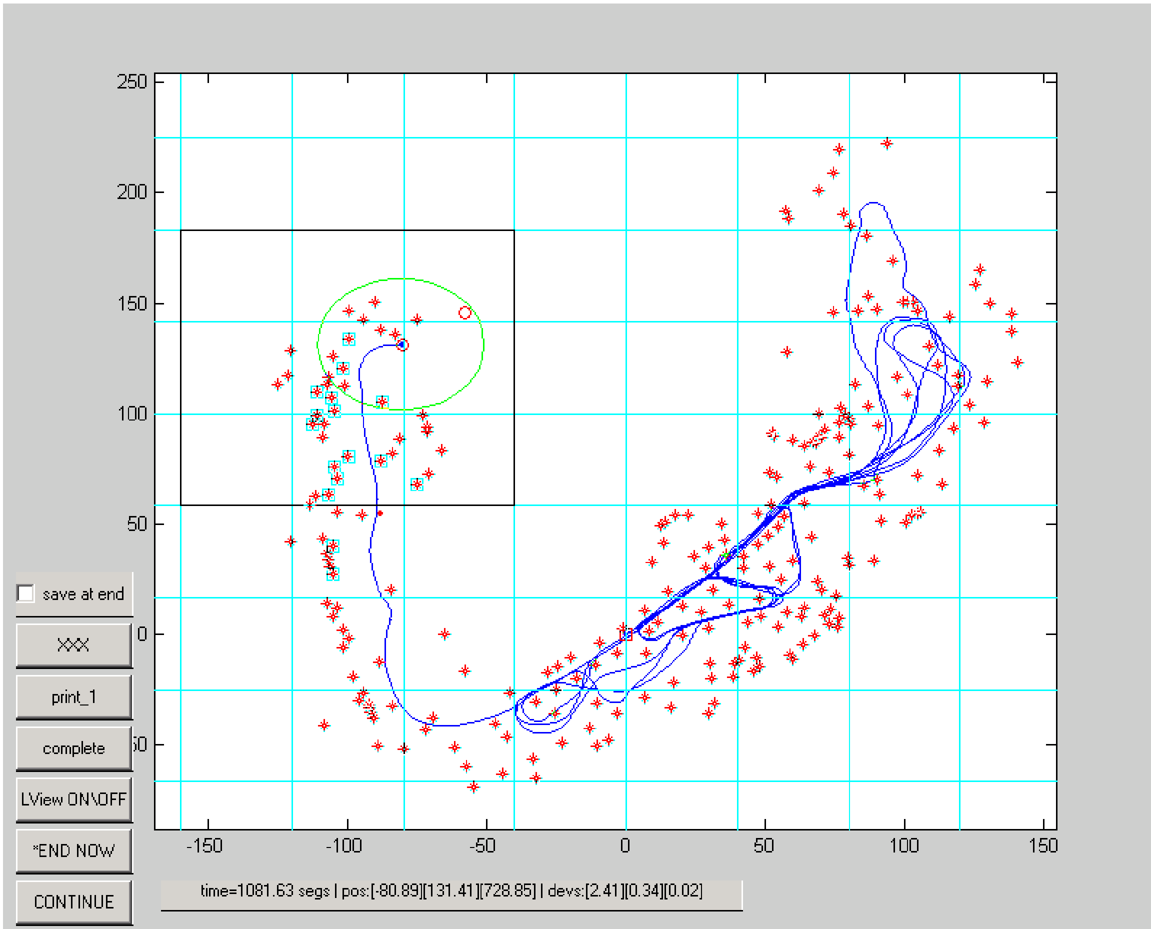
ČVUT
v Praze
in Prague



[courtesy E. Nebot]

Map and Trajectory

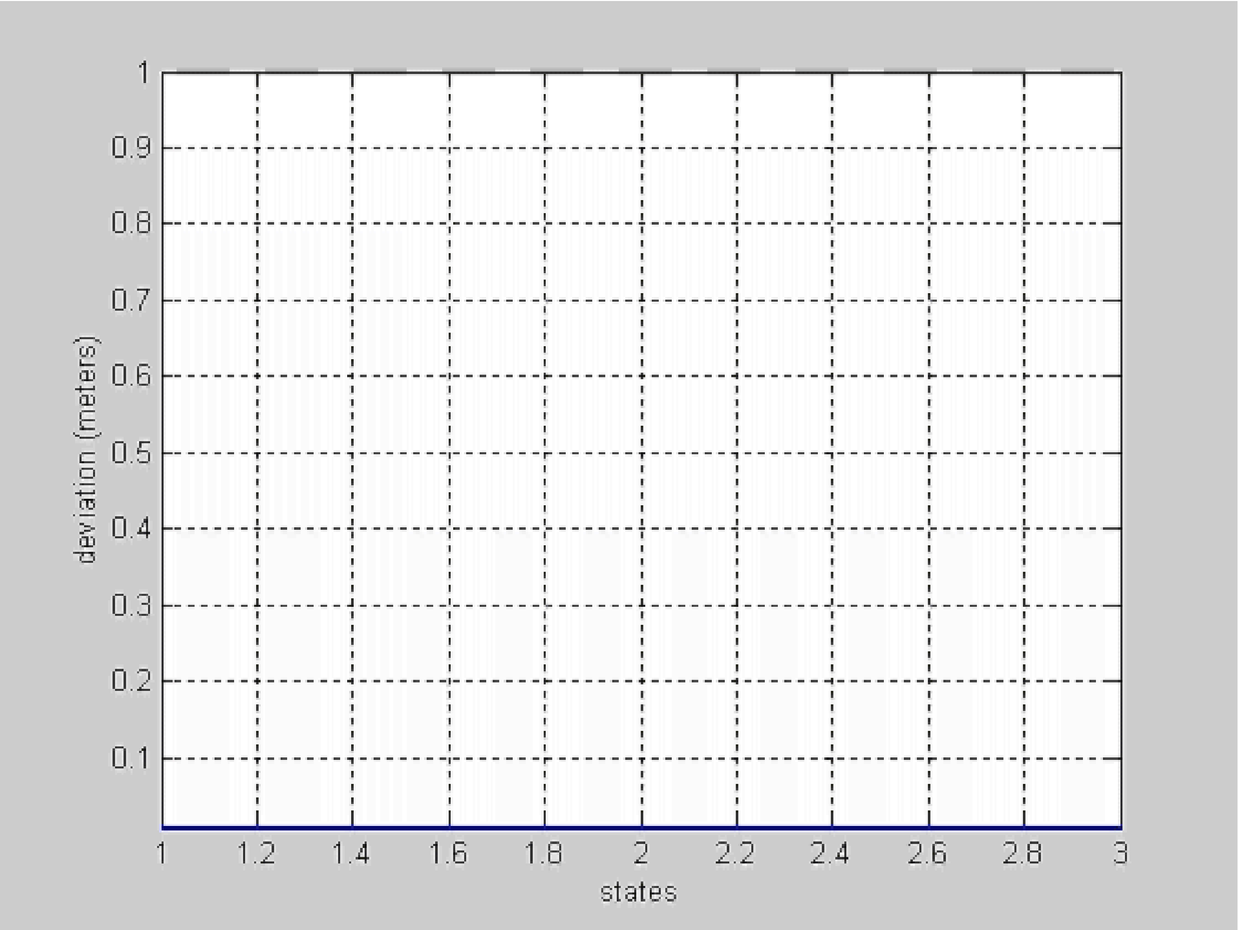
ČVUT
v Praze
in Prague



Landmarks
Covariance

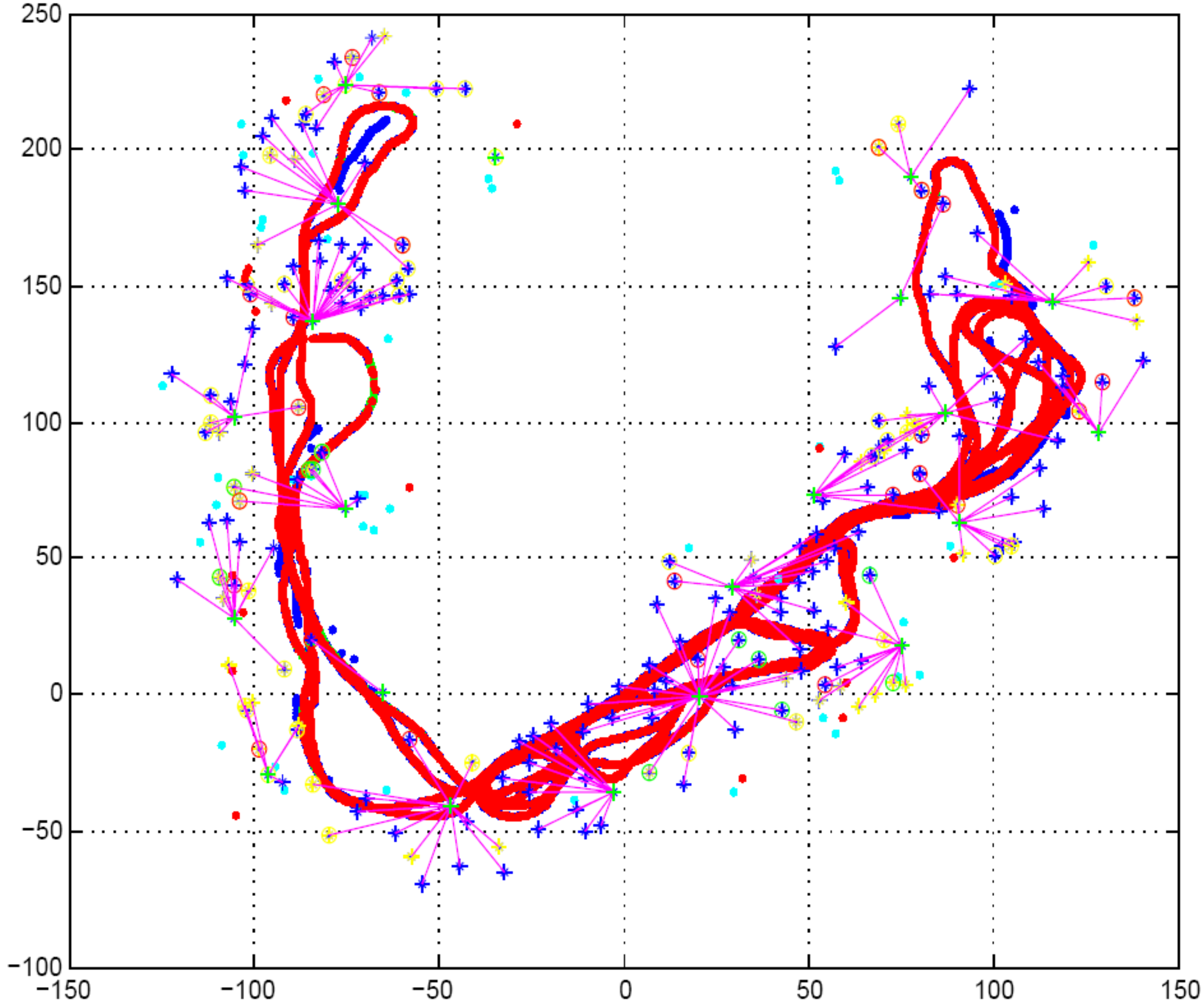
[courtesy E. Nebot]

Landmark Covariance



[courtesy E. Nebot]

Estimated Trajectory



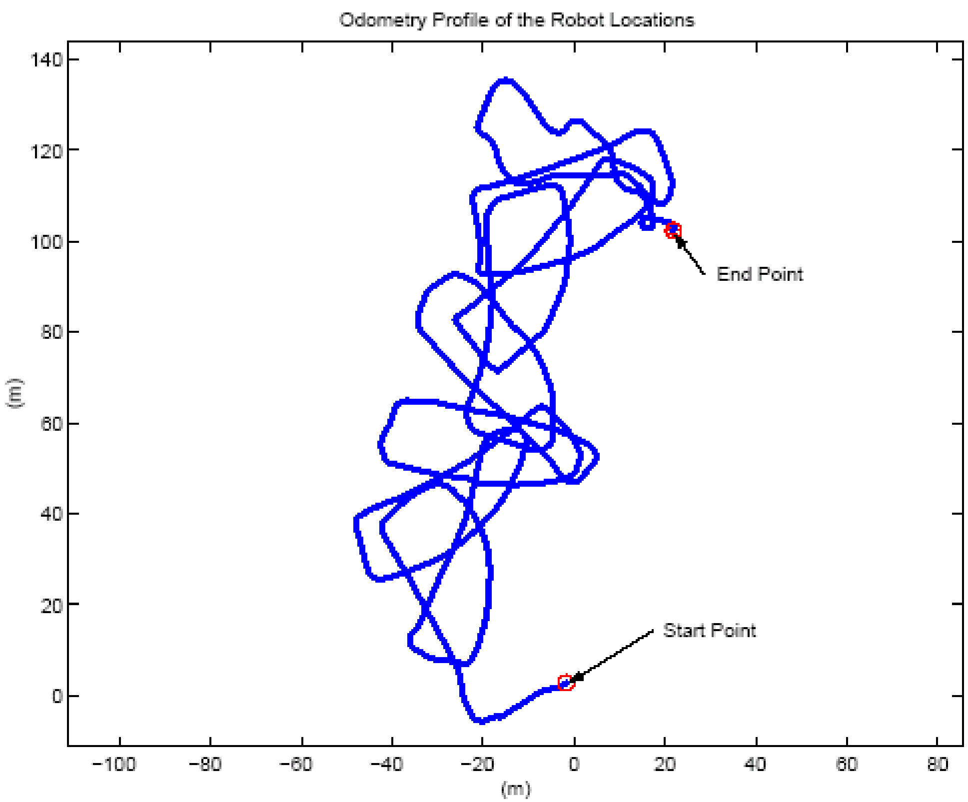
 [courtesy E. Nebot]

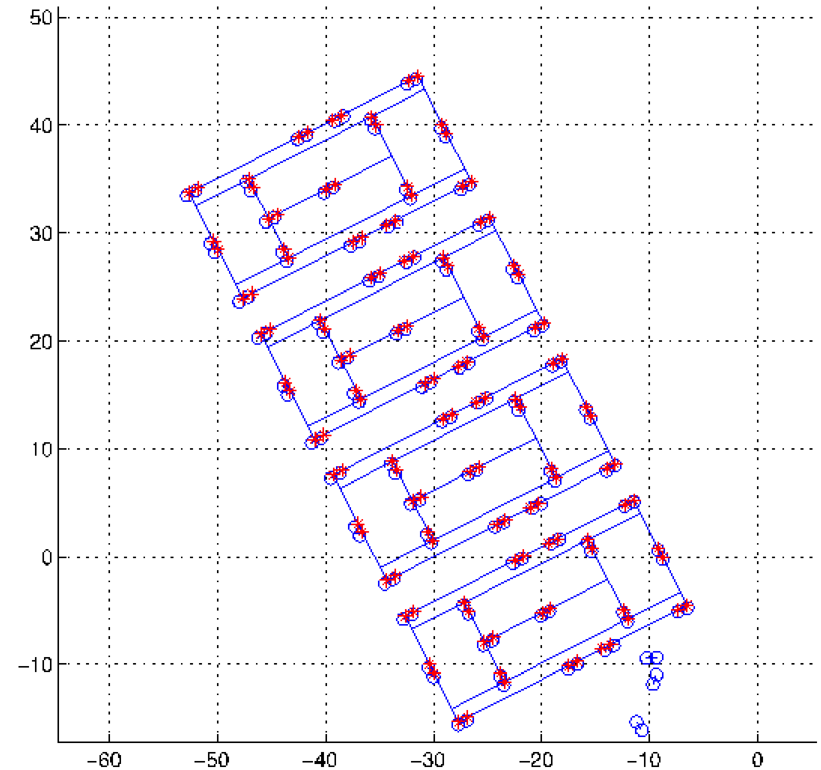
EKF SLAM Application

[courtesy John Leonard]

EKF SLAM Application

ČVUT
v Praze
in Prague





odometry

estimated trajectory

[courtesy John Leonard]

Approximations for SLAM

ČVUT
v Praze
in Prague

- Local submaps

  [Leonard et al.99, Bosse et al. 02, Newman et al. 03]

- Sparse links (correlations)

  [Lu & Milios 97, Guivant & Nebot 01]

- Sparse extended information filters

  [Frese et al. 01, Thrun et al. 02]

- Thin junction tree filters

  [Paskin 03]

- Rao-Blackwellisation (FastSLAM)

  [Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]

EKF-SLAM Summary

ČVUT
v Praze
in Prague

- Quadratic in the number of landmarks: $O(n^2)$
- Convergence results for the linear case.
- Can diverge if nonlinearities are large!
- Have been applied successfully in large-scale environments.
- Approximations reduce the computational complexity.

Particle filtering (Sequential Monte-Carlo)

ČVUT
v Praze
in Prague

- Kalman Filter – linear system, Gaussian noise.
- Kalman extensions (EKF,UKF) for non-linear systems.
- Particle Filtering – general filtering problems.

  - Hammersley & Morton 1954, Rosenbluth 1955
  - … …
  - Gordon et. al. 1993 (Re-sampling)
  - Van der Merwe, Doucet, de Freitas, Wan …(90-)

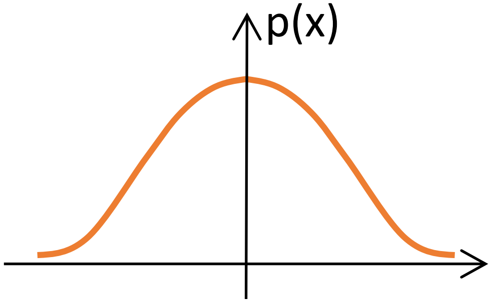
- Application areas: statistics, physics, engineering, finance…

Computer vision: CONDENSATION

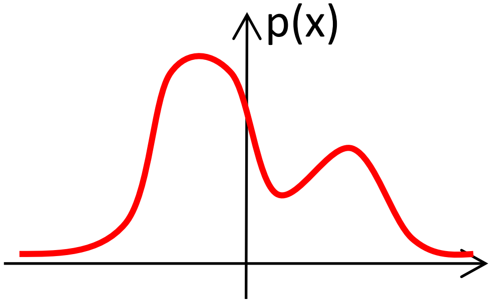
(conditional density propagation)

Michael Isard & Andrew Blake – ECCV 1996

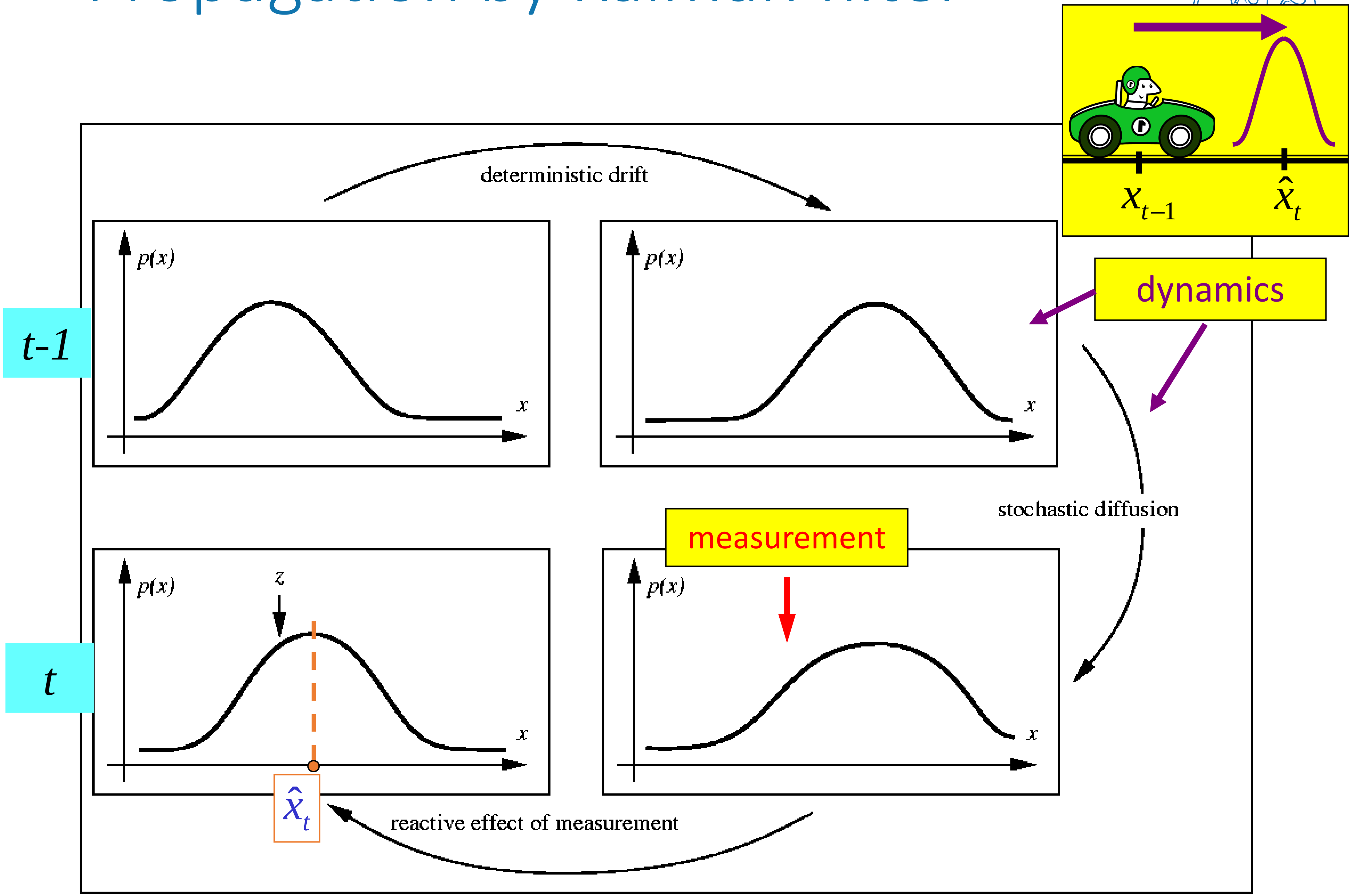
ČVUT
v Praze
in Prague



- Kalman filter in contour tracking:
  - Major assumption:
    Gaussian PDF of object's state –
  - Works relatively poorly in clutter:
    Multi-modal density -

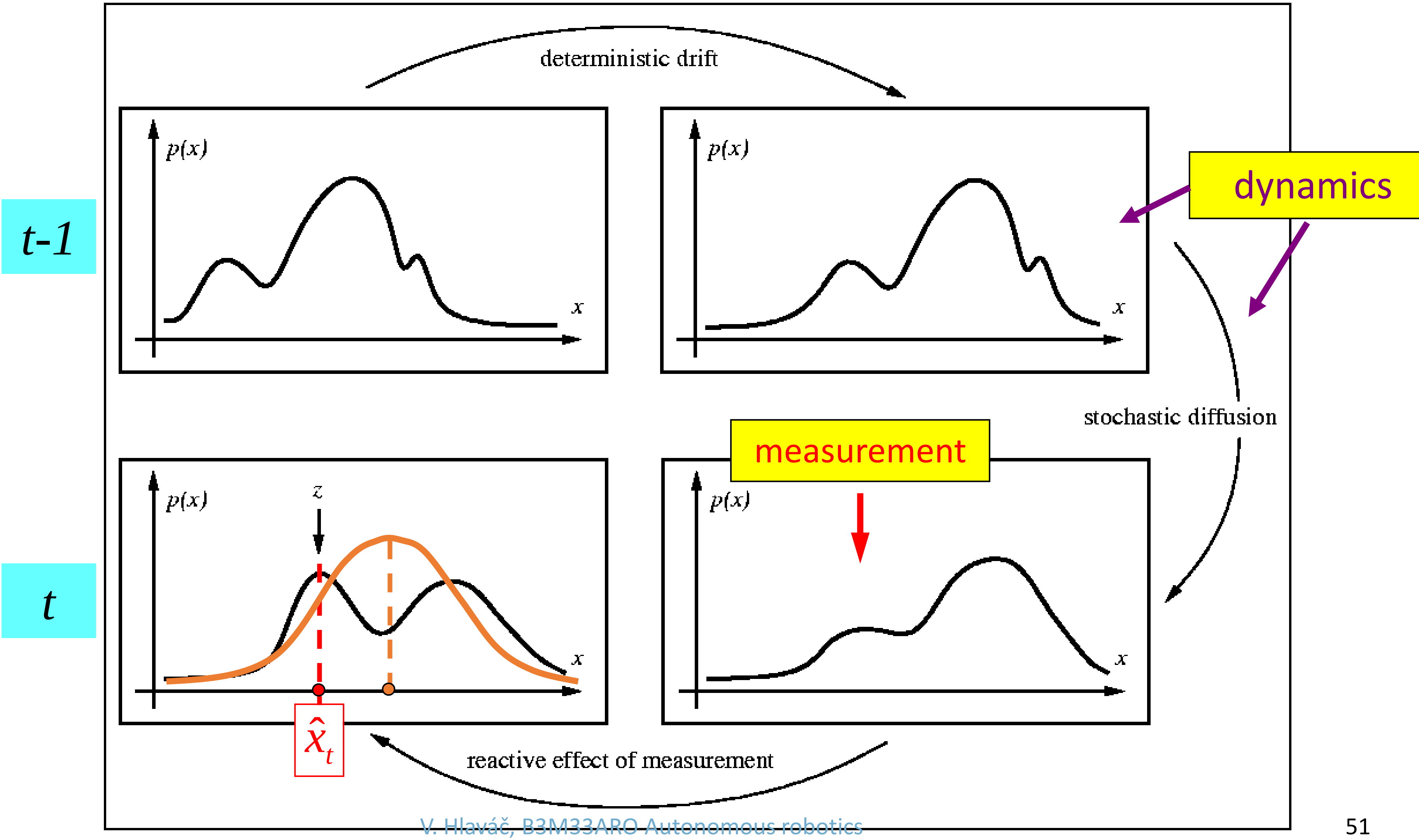


Propagation by Kalman filter

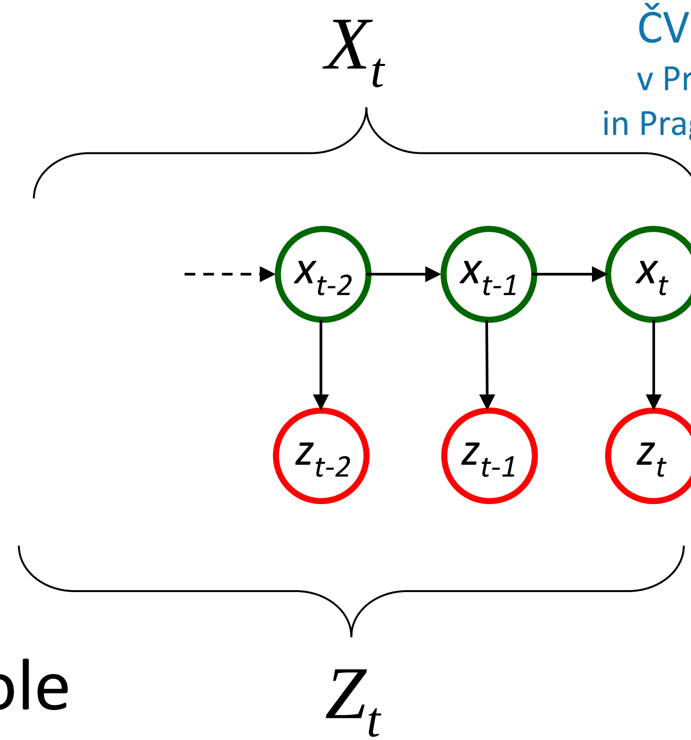


Propagation by Multi-Modal PDF

ČVUT
v Praze
in Prague

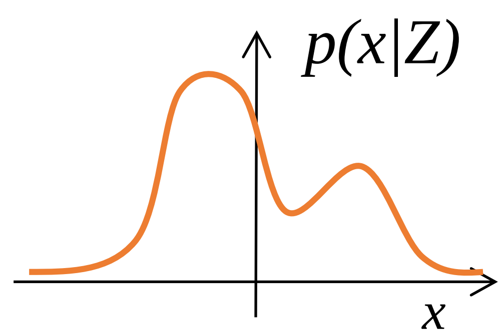


Discrete-time propagation



- Probabilistic model:
  - x_t – object state at time t.
    $X_t=\{x_1,\ldots,x_t\}$
  - z_t – image features at time t.
    $Z_t=\{z_1,\ldots,z_t\}$
- The goal – given Z_t , find the most likely x_t .
- Better to approximate the whole posterior density:

$$p(x_t \mid Z_t)$$



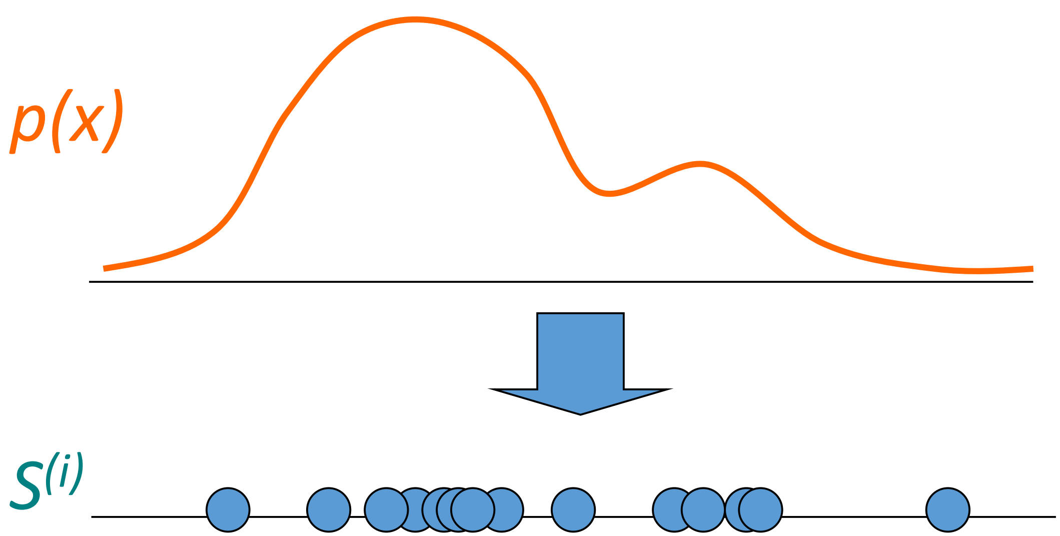
Factored sampling (1)

$$p(x \mid z) = k p(z \mid x)\, p(x)$$

Posterior — Observation (Likelihood) — Prior

We use iterative sampling to approximate the complex posterior $p(x/z)$:

1) **Sample** "particles" from $p(x)$ – $\{s^{(1)},\ldots,s^{(N)}\}$



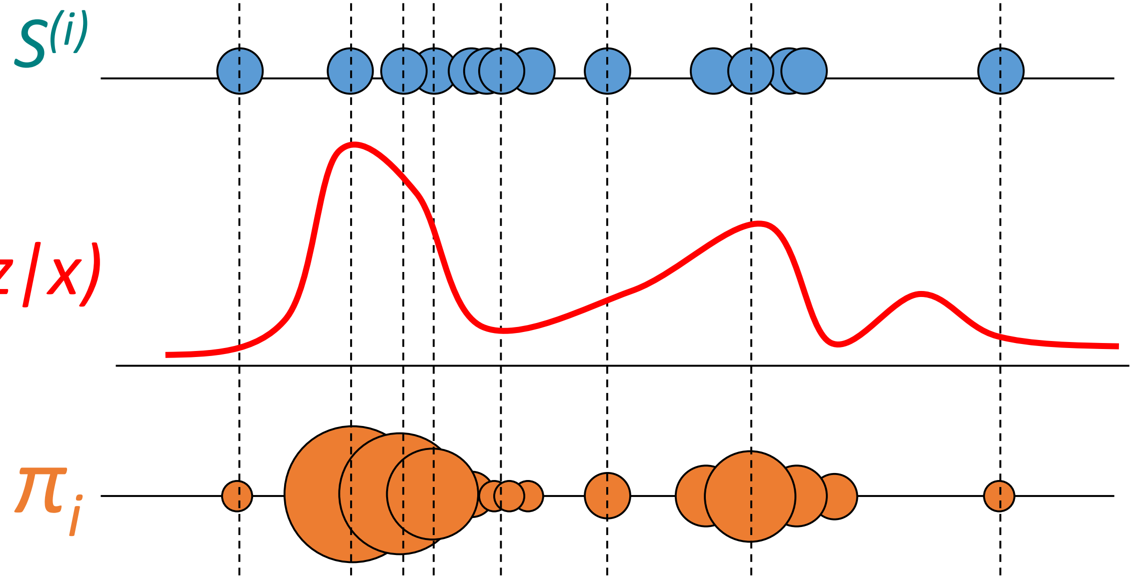
ČVUT
v Praze
in Prague

Factored sampling (2)

ČVUT
v Praze
in Prague

$$p(x \mid z) = k p(z \mid x) p(x)$$

2) **Weight** them according to the observation $p(z|x)$:



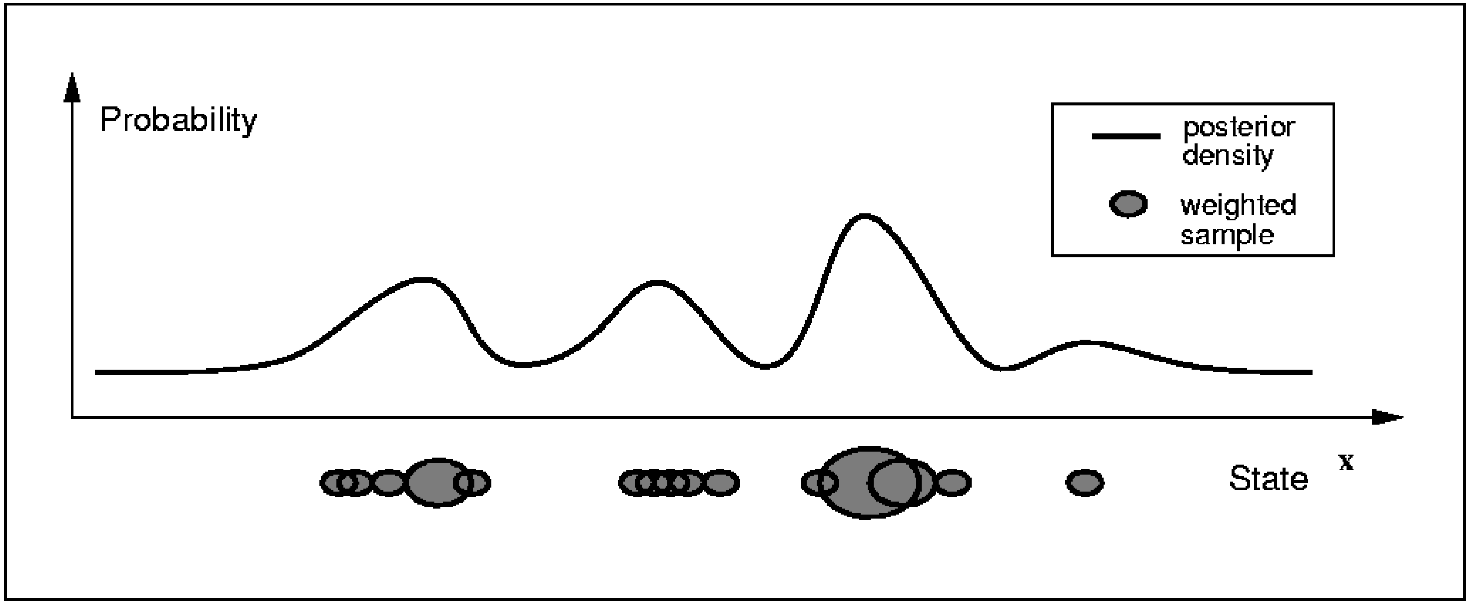
$$\pi_i \propto p(z \mid x = s^{(i)}) \qquad \sum \pi_i = 1$$

Factored sampling (3)

ČVUT
v Praze
in Prague

$$p(x \mid z) = k p(z \mid x)\, p(x)$$

3) The **“weighted particles”** are approximation of *p(x|z)*



Choosing $x' = x_i$ according to π_i will have a distribution that approximates the posterior $p(x|z)$. Accuracy increases with N.

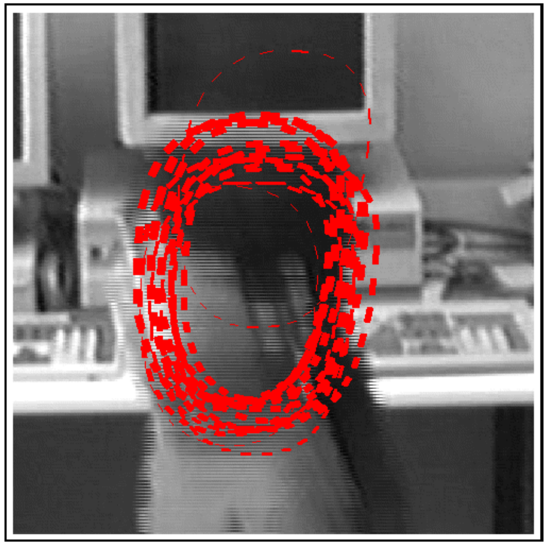
Factored sampling (4)

ČVUT
v Praze
in Prague

Can calculate easily **statistics** of the posterior:

$$E[g(x)\,|\,z] \approx \sum_{i=1}^{N} g(s^{(i)})\pi_i$$

- *e.g., the mean - g(x)=x:*



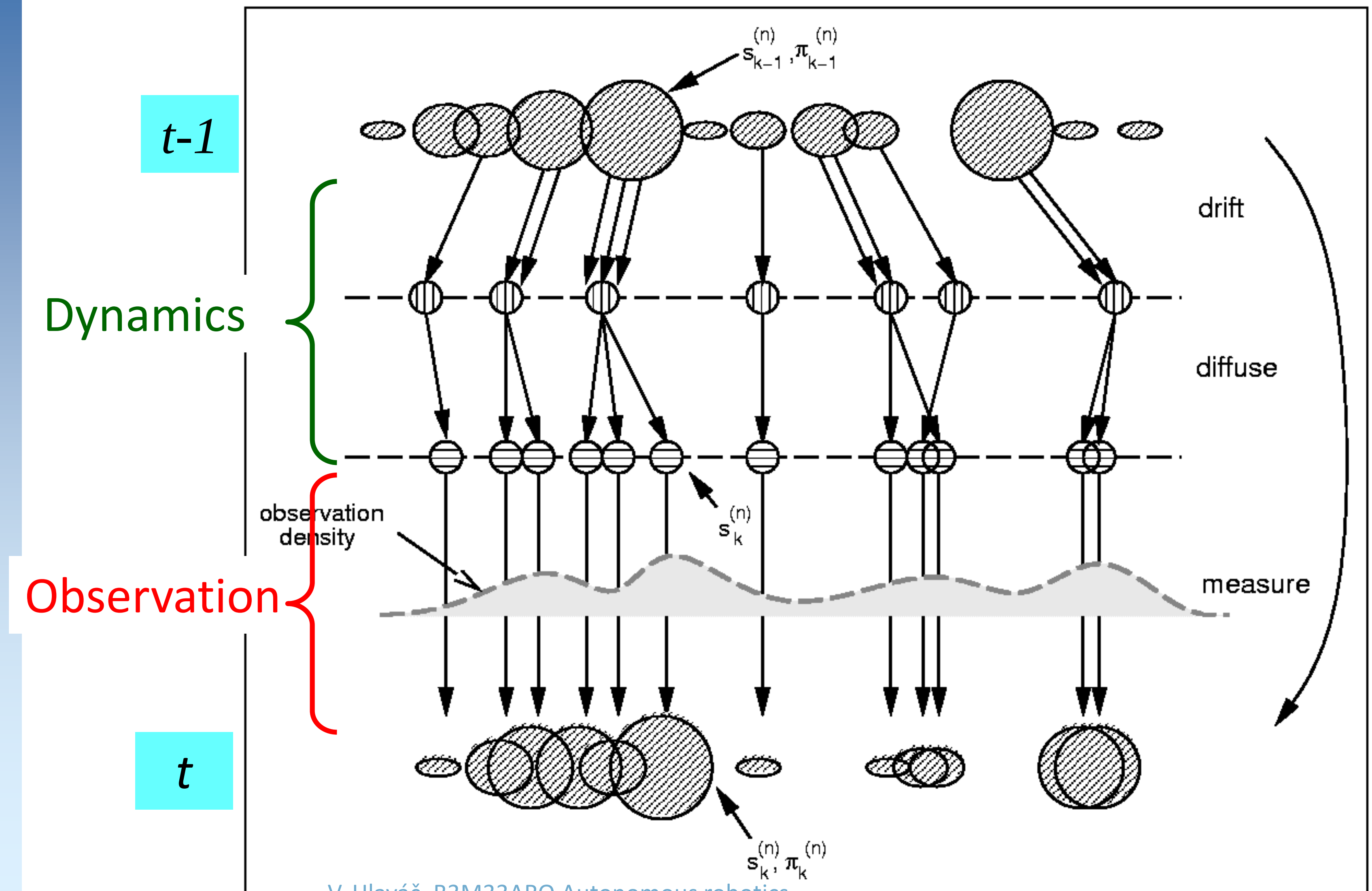
weighted samples

the mean

Condensation

ČVUT
v Praze
in Prague

one iteration:



Condensation, illustrative video