



**BME**

*Budapesti Műszaki és Gazdaságtudományi Egyetem*

**KJIT**

*Közlekedés- és Járműirányítási Tanszék*



# Related concepts

Concepts related to vehicles moving in an environment:

- State estimation
- Localisation
- Mapping
- SLAM (simultaneous localization and mapping)
- Navigation, motion planning

General term describing this field: **Robot Mapping**

# SLAM and related topics

- **State estimation:** in this context the state to be estimated is the position (and orientation) of the vehicle and the positions of the landmarks or other features that are recognized in the environment
- **Localizaton:** determining the position of the vehicle on a given map
- **Mapping:** bulding a map of the environment with known vehicle positions
- **SLAM:** simultaneous localization and mapping, vehicle position and map is unknown
- **Navigation:** given a map the vehicle autonomously chooses a path and reaches the destination

# SLAM introduction

## Applications of SLAM:

- Robot vacuum cleaner and lawn mower
  - UAV
  - Explorating mines
  - Mars rovers
- } Dynamic environment
- } Static environment

SLAM algorithms do not tell where the vehicle should go. An additional logic of human interaction should guide the robot to explore the whole field. (SLAMP: SLAM and Planning)

- 1986: Formulating the problem
  - R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986
- 1989: first implementation (with Extended Kalman filter)
  - P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. *1st International Symposium on Roboti Experimental cs*, Montreal, June, 1989
- 1995: appearance of SLAM acronym

# SLAM introduction

- Mapping well structured, static, limited regions nowadays are considered a solved problem
- No general robust and effective methods for mapping unstructured, dynamic, extended regions
- The environment can be represented with:
  - Landmark map: using discrete objects (measurement from optical camera)
  - Volumetric map:
    - Occupancy grid map (2D)
    - Surface map (2.5D)
    - Full 3D map
  - Graphs
- Building the map can be:
  - Incremental – online
  - Batch, multipass – offline

# Difficulties in SLAM

- When observing a landmark again we should recognize that it is already registered. Data Association problem
- Usually the environment contains dynamic objects too
- We should distinguish the static and dynamic objects
- Objects that move slowly or rarely pose difficulties

These difficulties let to the topics of SLAMIDE and DATMO

SLAMIDE: SLAM In Dynamic Enviroments

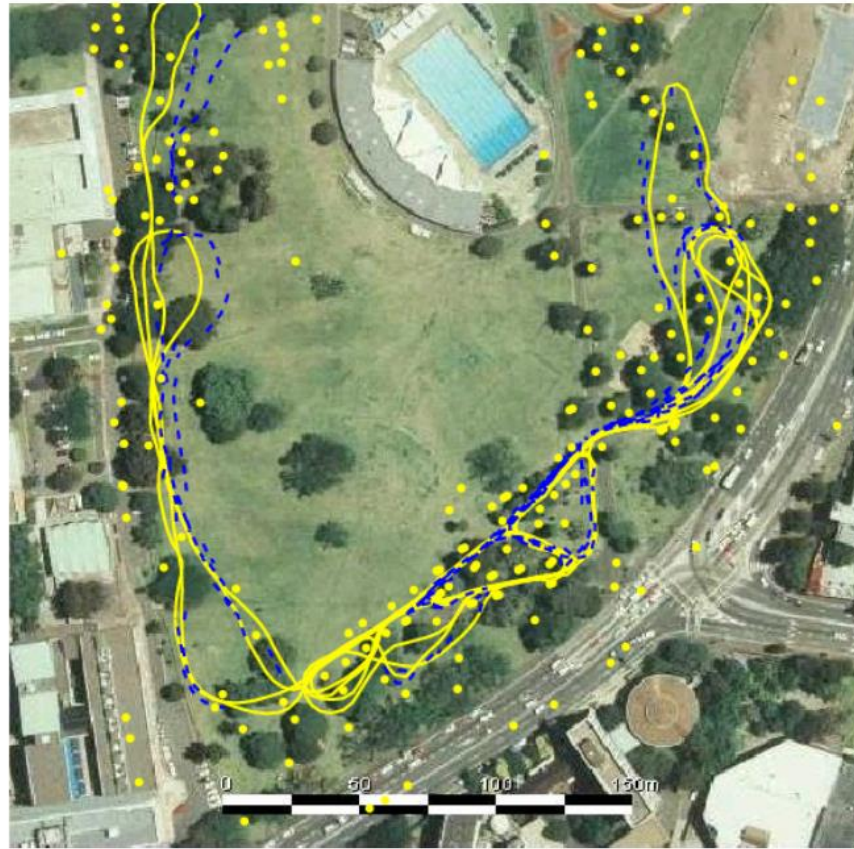
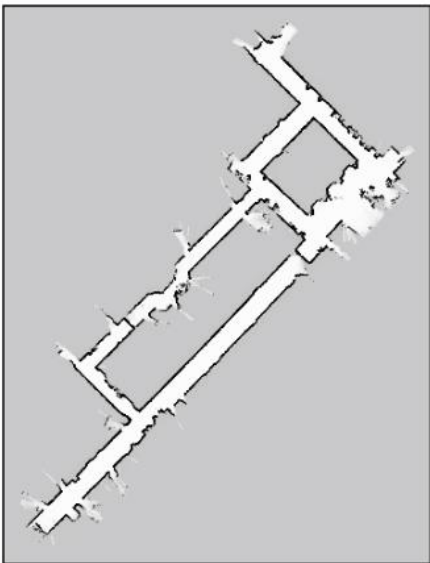
DATMO: Detection And Tracking Of Moving Objects

# SLAM classification

- Grouping by different aspects
  - Volumetric or feature based mapping
  - Topologic or geometric map
  - Static or dynamic environment
  - Active or passive control
  - Single or multiple robots
- Probabilistic approach: Bayes estimation (Kalman filter, Particle filter)
- There are algorithms using graphs

# Volumetric vs Feature based map

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járőrnyitási Tanszék



Courtesy by E. Nebot



# Volumetric vs Feature based map

- Feature based map
  - Needs well defined and easily recognisable landmarks
  - Map won't be detailed
- Volumetric map
  - Occupancy grid map: 2 dimensional representations
  - Surface map: occupancy grid map with surface elevation data (2.5D)
  - Full 3D map: computationally expensive

# Mapping with occupancy grid map

- First mesh the region
- Every cell is assigned a probability that measures its occupancy
- The object occupying a cell can be arbitrary
- The resolution of the mesh should be fine tuned
- Visualization is simple: gradient colors that are proportional to the probabilities
- Make the mesh in a way that the center of the cells have integer coordinates. Round the coordinates of a newly found object to integer numbers to immediately get which cell it occupies.

# Metric vs topologic map

## Metric map

- Direct method
- Needs little preliminary processing
- Contains geometric features of the environment
- Usually realized by occupancy grid map

## Topologic map

- Represents the environment by the connections between their regions
- A topologic map is generated from geometric information
- Usually using graphs

# Single or multiple robots

- Multo-robot SLAM poses additional data association problems: maps generated by individual robots should be combined. Even more complicated when the initial relative positions of the robots were unknown.
  - A possible solution: robots try to localize themselves in each other's maps. This method requires implementing  $n^2$  filters for  $n$  robots. (B. Stewart, J. Ko, D. Fox and K. Konolige. A hierarchical bayesian approach to mobile robot map structure estimation. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003)
  - An other solution: maps are represented by special graphs (Markov random fields) which are compared by algorithms using tree structures ([S. Thrun and Y. Liu. Multi-robot SLAM with Sparse Extended Information Filers, Robotics Research. Vol.15., pp254-266, 2005](#))

# Additional categorization

- Active or passive control: the robot can navigate autonomously or can be controlled remotely
- Known correspondence: if landmarks are labeled it is easier to associate the measurements
- Static vs dynamic environment: mapping in a dynamic environment is a much harder problem

# Notation

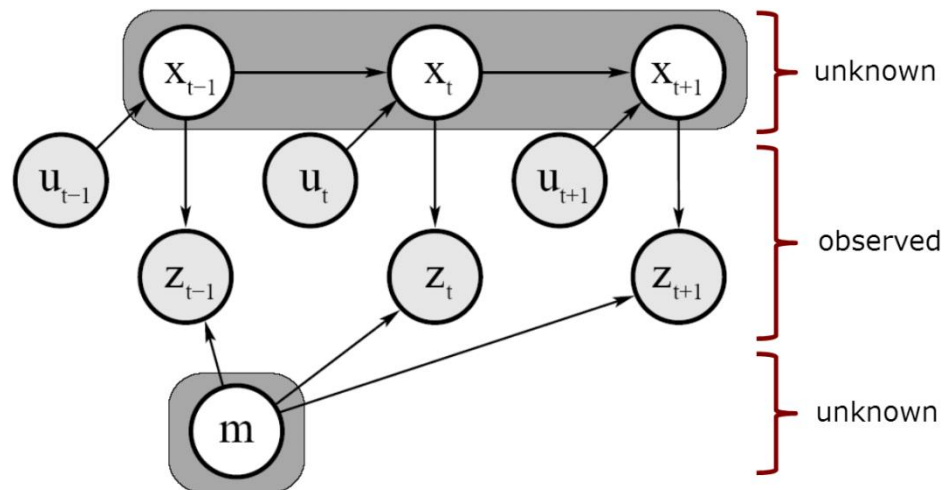
- Map of the environment:  $m$ , vector containing the coordinates of the landmarks
- Position of the vehicle at time  $t$  is  $x_t$
- Trajectory of the vehicle is the sequence of its positions:  $\{x_0, x_1, x_2, \dots, x_t\}$  that is written as :  $x_{0:t}$  or  $x^t$
- Sensor measurements at time  $t$  is  $z_t$
- Control signal at time  $t$  is  $u_t$ . This is not the remote command or the signal from an algorithm responsible for navigation. In SLAM the control signal is a measurement coming from the sensors of the vehicle measuring its velocity, orientation, acceleration etc.

# The SLAM problem

- The position  $\mathbf{x}_0$  defines the starting point of the robot. This can be an arbitrary point. Usually the origin is chosen or a point that defines the position of the map to be created in an existing bigger map.

## Illustrating the problem:

- The state  $\mathbf{x}$  and map  $m$  is unknown
- The measurement  $\mathbf{z}$  and the control signal  $\mathbf{u}$  is known



# Formulating the SLAM problem

- The full SLAM problem is to estimate the vehicle trajectory  $x_{0:t}$  and the map  $m$  given  $u_{1:t}$  and  $z_{1:t}$  which is determining the

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

conditional PDF

- Online SLAM: only the actual  $x_t$  position is estimated:

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

- The full SLAM problem can be solved with a smoothing algorithm while the online SLAM needs filtering.
  - Smoothing: estimate the current and past values
  - Filtering: estimate the current values



# Modeling

We need probabilistic approach because the measurements are always noisy and the models are always inaccurate. We will get distributions of the variables in question and from the them we can create point estimates.

- Motion model
  - It gives what  $x_t$  state we expect if the previous state was  $x_{t-1}$  and the control signal is  $u_t$
  - Knowing the motion model is equivalent to specifying the conditional PDF  $p(x_t|x_{t-1}, u_t)$
- Measurement model
  - It gives what  $z_t$  measurement we expect if the system is at state  $x_t$
  - Knowing the measurement model is equivalent to specifying the conditional PDF  $p(z_t|x_t, m)$

# Modeling

- The motion or transition model, depending on the sensors, can be
  - Odometric model
  - Velocity model
- The measurement or sensor model can describe
  - Distance measurement
  - Bearing measurement
  - Distance and bearing measurements

# SLAM in Bayes-formalism

- The full SLAM problem is to give the

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

conditional PDF

In the following, for brevity the state  $x$  will contain  $m$ .

- Using Bayes-theorem we have:

$$p(x_{0:t} | z_{1:t}, u_{1:t}) = \frac{p(z_{1:t} | x_{0:t}, u_{1:t}) p(x_{0:t} | u_{1:t})}{p(z_{1:t} | u_{1:t})}$$

where

$$p(z_{1:t} | u_{1:t}) = \int p(z_{1:t} | x_{0:t}, u_{1:t}) p(x_{0:t} | u_{1:t}) dx_{0:t}$$

# SLAM in Bayes-formalism

$$p(x_{0:t}|z_{1:t}, u_{1:t}) = \frac{p(z_{1:t}|x_{0:t}, u_{1:t})p(x_{0:t}|u_{1:t})}{p(z_{1:t}|u_{1:t})}$$

- $p(z_{1:t}|x_{0:t}, u_{1:t})$  **likelihood**, this factor contains the sensor model
- $p(x_{0:t}|u_{1:t})$  **prior distribution**, gives the distribution of the predicted state
- $p(z_{1:t})$ : **evidence or marginal likelihood**, acts as a normalizing factor. Can be computed by marginalizing out the state from the likelihood.
- $p(x_{0:t}|u_{1:t}, z_{1:t})$  **posterior distribution**: gives the distribution of the state taken into account the measurements

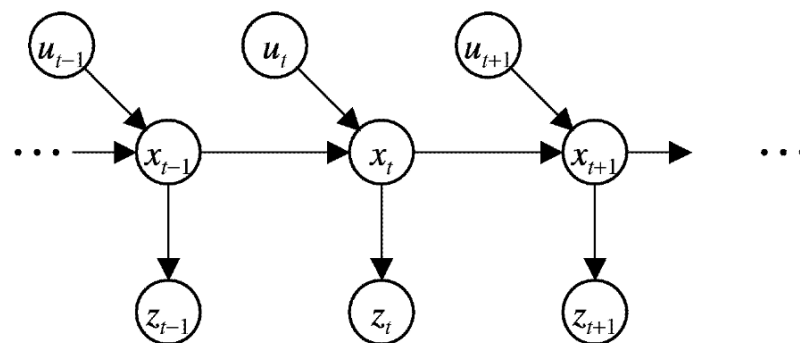
Applying the Bayes formalism is to determine these factors, which is the hard part of the problem

# SLAM in Bayes-formalism

- In real-time implementation the full SLAM problem becomes intractable very soon because of the constantly increasing dimensionality.

Simplify the problem:

- Assume that the system can be Modelled by a first order Markov-chain. The state  $x_t$  depends only on  $x_{t-1}$  and
- The measurement  $z_t$  depends only on  $x_t$ .
- Online SLAM: estimate  $x_t$  only



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

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

# SLAM in recursive Bayes-estimation

- We estimate only  $x_t$  the problem can be solved in a recursive way
- Previous estimates won't be changed or re-estimated when new measurements arrive
- The first order Markov chain assumption does not require to recalculate estimations because only the previous state has effect on the current one.

With the above simplifications and some formula manipulation we have the online posterior in the form:

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

# SLAM in recursive Bayes-estimation

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

In recursive form the factors are:

- $p(z_t | x_t)$  sensor model or likelihood
- $p(z_t | z_{1:t-1}, u_{1:t})$  normalizing factor, given as:

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

- $p(x_t | z_{1:t-1}, u_{1:t})$  the prior factor assembles as:

$$p(x_t | z_{1:t-1}, u_{1:t}) = \int \underbrace{p(x_t | x_{t-1}, u_t)}_{\text{motion model}} \underbrace{p(x_{t-1} | z_{1:t-1}, u_{1:t-1})}_{\text{previous posterior}} dx_{t-1}$$

# SLAM in recursive Bayes-estimation

- Computing the integrals is hard, no general analytic solution exists. We have to approximate.
- In some special cases the problem can be solved exactly. When dealing with linear systems and normal noise models the Kalman filter can be used, which gives the optimal solution.
- The normal distribution has two parameters, the expected value ( $\mu$ ) and the covariance ( $\Sigma$ ), which describe the distribution completely. In the recursion these two values represent the PDF.
- In SLAM the motion and sensor models are generally nonlinear thus the Extended Kalman Filter (EKF) can be used.
- The first SLAM implementation used EKF



# EKF SLAM formalism

- The state  $x_t$  contains the position of the vehicle and the coordinates of the  $M$  landmarks ( $2M+3$  dimension):

$$x_t = (x, y, \theta, m_{1x}, m_{1y}, m_{2x}, m_{2y}, \dots, m_{Mx}, m_{My})^T$$

- The motion model has  $w_t \sim \mathcal{N}(0, R_t)$  additive normal noise:

$$x_t = g(x_{t-1}, u_t) + w_t$$

- The sensor model has  $v_t \sim \mathcal{N}(0, Q_t)$  additive normal noise:

$$z_t = h(x_t) + v_t$$

- The motion model can be decoupled into two parts: the first part describes the vehicle motion and the second part describes the motion of the landmarks which is actually standing still.
- The vehicle motion model usually contains trigonometric functions.
- The model is usually nonlinear too.
- For simplicity we assume that the landmarks are labelled.

# EKF linearization

- $g(\cdot)$  és  $h(\cdot)$  functions are linearized (Taylor expansion):
  - $g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1})$
  - $h(x_t) \approx h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t)$
- $\mu_{t-1}$  is the expected value of the previous estimate:
$$\mu_{t-1} = E[p(x_t | z_{1:t}, u_{1:t})]$$
- $\bar{\mu}_t$  expected value of the prior distribution:
$$\bar{\mu}_t = E[p(x_t | z_{1:t-1}, u_{1:t})]$$
- The Jacobians are:

$$G_t = \left. \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}} \right|_{x_{t-1} = \mu_{t-1}}$$

$$H_t = \left. \frac{\partial h(x_t)}{\partial x_t} \right|_{x_t = \bar{\mu}_t}$$

# EKF SLAM

The linearized motion model:

$$x_t = g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1}) + w_t$$

The linearized measurement model:

$$z_t = h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t) + v_t$$

The usual filtering algorithm read as:

1.  $\bar{\mu}_t = g(\mu_{t-1}, u_t)$  //  $A_t\mu_{t-1} + B_tu_z$
2.  $\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^T + R_t$
3.  $K_t = \bar{\Sigma}_tH_t^T (H_t\bar{\Sigma}_tH_t^T + Q_t)^{-1}$
4.  $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$  //  $C_t\bar{\mu}_t$
5.  $\Sigma_t = (I - K_tH_t) \bar{\Sigma}_t$

# EKF SLAM algorithm

**Initialization:** we have to give the initial distribution from which we can start the recursion. This means giving two quantities, the expected value (initial position) and the covariance (initial uncertainty) of the state.

- The initial position of the vehicle can be arbitrary. Typical choice is the origin.
- The positions of the landmarks are arbitrary too. The chosen values has no effect, because at the beginning we are infinitely uncertain about these values.
- The vehicle part of the covariance matrix is a  $3 \times 3$  block of zeroes, theoretically, since there is no uncertainty about the values we set as the initial position. For practical/numerical reasons it is good to introduce some uncertainty.
- Other parts of the covariance matrix can be infinitely large, which means we have no information about the positions of the landmarks. This is why it doesn't matter what initial coordinates we set for the landmarks.

# EKF SLAM algorithm

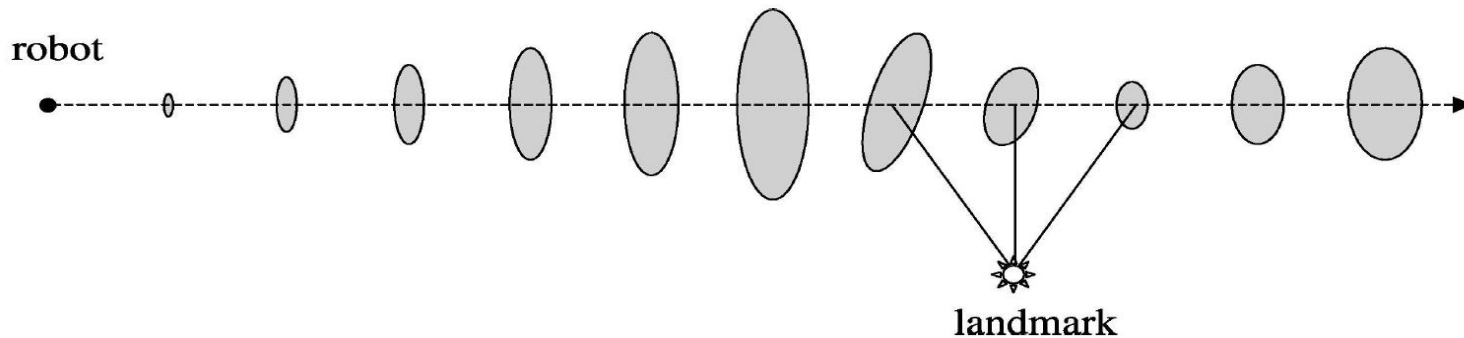
- Initial expected value and covariance:

$$\mu_0 = (0 \ 0 \ 0 \ \dots \ 0)^T$$
$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ \infty & \infty & \infty & \infty & \dots & \infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \infty & \dots & \infty \end{pmatrix}$$

- The motion of the vehicle and the landmarks can be decoupled thus we shall not work with matrices of dimension  $2M + 3$  since it would. No need to waste computational power on identity operations and multiplications with zero.
- We use block matrices and augment them to  $2M + 3$  dimension.
- E.g. compute only those parts of the  $H$  matrix that correspond to the actual observation.
- The matrices that are symmetric in theory are sometimes violate this property because of numerical errors. We should force these matrices to be symmetric manually.
- Alternatively we can use certain computational technics that do not introduce this type of error.

# Loop closure

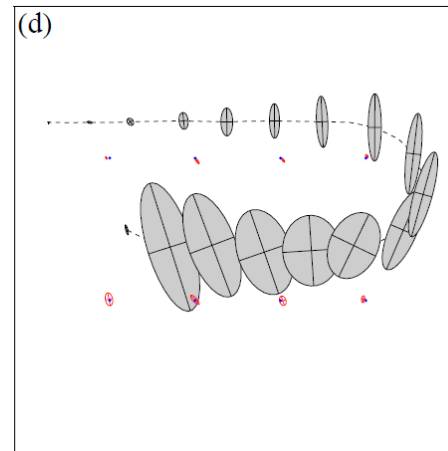
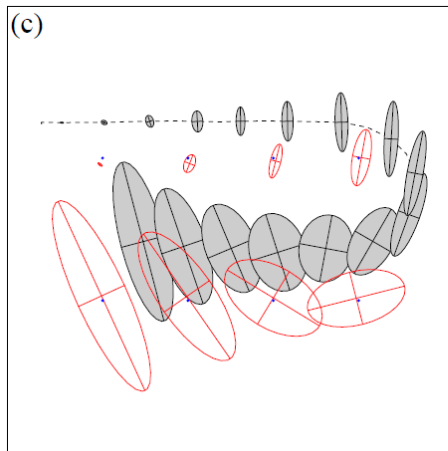
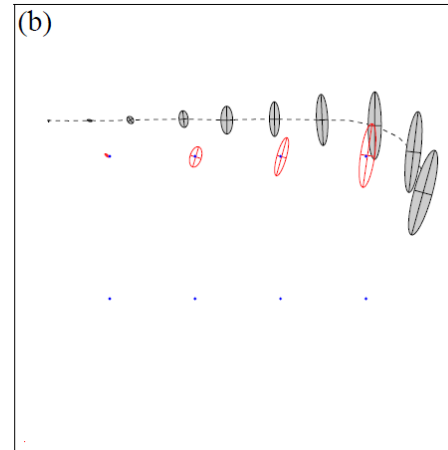
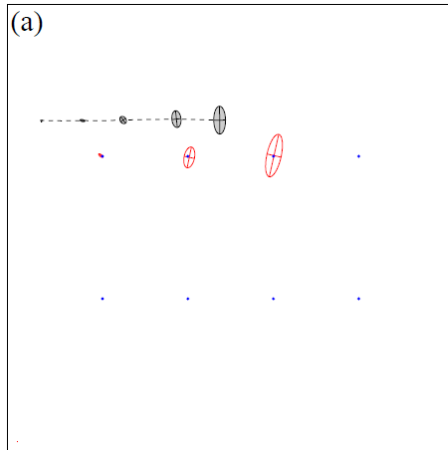
- If we would know the landmark positions:
  - The vehicle starts and its position becomes more and more uncertain
  - Observing a known landmark helps decreasing the uncertainty
  - A következő iránypontig újra nő a bizonytalanság



- If the initial positions of the landmarks are unknown then the uncertainty of the vehicle position grows over time and so do the observed landmark positions.
- This continues until a landmark is observed a second time when returning to an already visited place.

# Loop closure – illustration

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék



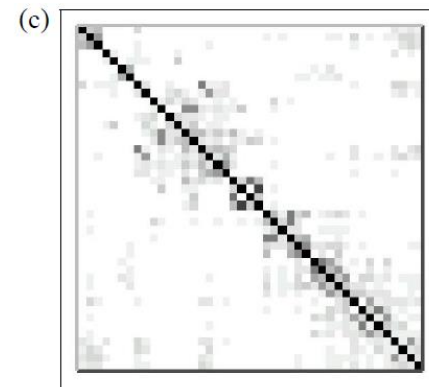
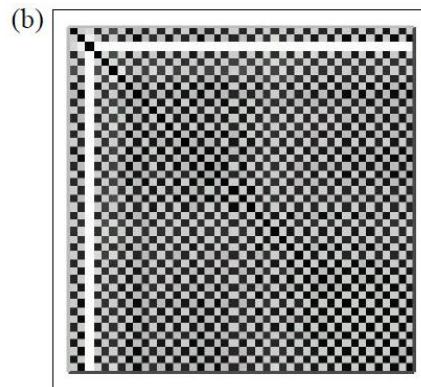
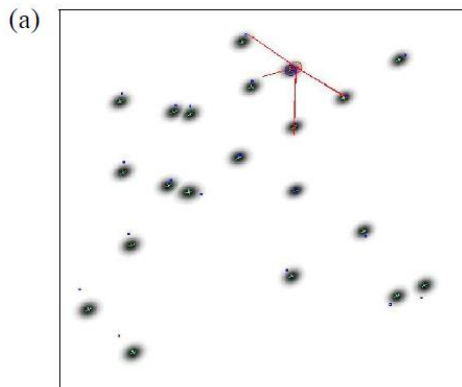
# Loop closure – illustration

- We have 8 landmarks on the map. Uncertainties are represented by ellipses.
- The vehicle starts and the position error grows which causes that the error in the newly observed landmarks' positions gets larger and larger.
- Vehicle position error correlates with the landmark position errors.
- A newly observed landmark cannot reduce the position error, since there is no prior information about where it should be.
- Observing the first landmark the second time causes the errors to decrease in the vehicle position. This implies the error reduction in all the landmark positions.
- In one step all the errors decreased. This effect doesn't go back in time, previous vehicle positions won't get fixed or reestimated.
- All the information accumulates in the covariance matrix.



# Visualize uncertainty

- (a) Uncertainty of a point is visualized by the error ellipse drawn around the point. The axes are proportional to the variance of the coordinates.
- (b) By normalizing the covariance matrix we get the correlation matrix, which is easy to display. The white band tells us that there is no linear dependence between the heading of the vehicle and the positions of the landmarks.
- (c) Inverse covariance matrix – Precision matrix: while the covariance matrix shows the unconditional correlation the precision matrix shows partial correlation. We generally have only local dependencies that can be exploited for building submaps.



# Unknown correspondence

- If landmarks have no labels we can't assign measurement to landmarks with absolute certainty.
- We have to solve the data association problem: the arrived measurement should be assigned to an already seen landmark or register a new one?
- Mahalanobis distance helps to decide
- Mahalanobis distance: defines the distance between a point ( $x$ ) and a distribution with expected value  $\mu$  and covariance  $\Sigma$ :

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

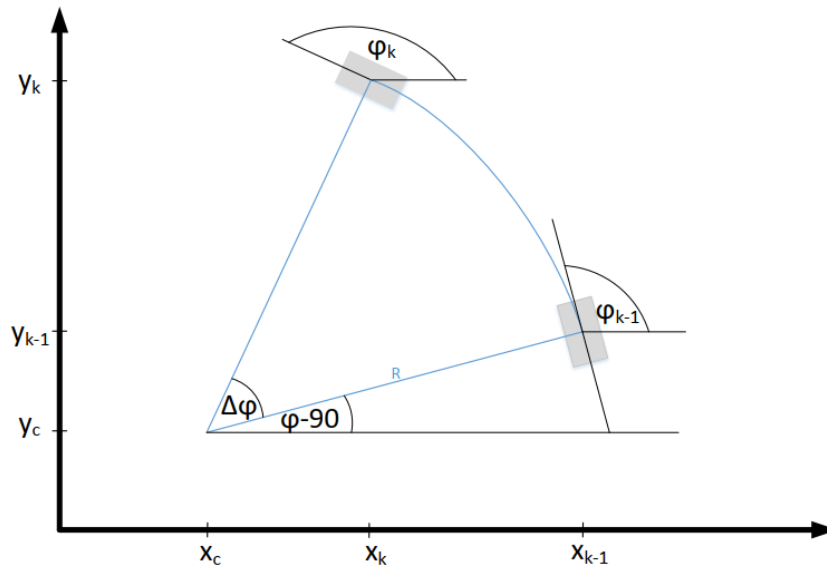
- Association: use maximum likelihood method to estimate the correspondence. We choose the configuration that has minimum distance.

# EKF SLAM summary

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedés- és Járműirányítási Tanszék

- The EKF is recursive, the information accumulates in the covariance matrix.
- Incremental mapping, can be used in real time.
- Can be used to build feature based/landmark maps
- Landmarks have to be separable from each other.
- Has dimensionality  $2M+3$
- Computational cost scales quadratically with the number of landmarks.
- Has limited capabilities in dynamic environment
- Convergence is proven (with linear models and normal noise)
- Can be divergent with nonlinear models
- Gaussian constrain implies that the distribution is unimodal no alternate possibilities are taken into account, only the most probable one.
- In extended areas the algorithm gets to be untractable because of high dimensionality
- There are other solutions → FastSLAM

# Robot motion



$$x_c = x_{k-1} + R \cos(\varphi_{k-1} + \pi/2)$$

$$y_c = y_{k-1} + R \sin(\varphi_{k-1} + \pi/2)$$

$$x_k = x_c + R \cos(\varphi_{k-1} + \Delta\varphi - \pi/2)$$

$$y_k = x_c + R \sin(\varphi_{k-1} + \Delta\varphi - \pi/2)$$

$$\varphi_k = \varphi_{k-1} + \Delta\varphi.$$

# Sources

- S. Thrun, W. Burgard, D. Fox (2005): Probabilistic Robotics, The MIT Press
- Simo Särkkä (2013): Bayesian Filtering and Smoothing, Cambridge University Press
  - [http://users.aalto.fi/~ssarkka/pub/cup\\_book\\_online\\_20131111.pdf](http://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf)
- A. Doucet & A. M. Johansen (2008): A Tutorial on Particle Filtering and Smoothing
  - [www.cs.ubc.ca/~arnaud/doucet\\_johansen\\_tutorialPF.pdf](http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf)
- M. Montemerlo: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association, PhD thesis, School of computer science, Carnegie Mellon University, Pittsburgh, USA, 2003,