Survey: Simultaneous Localisation and Mapping (SLAM)

Ronja Güldenring
Master Informatics
Project Intellgient Robotics
University of Hamburg

Introduction

EKF-SLAM

FastSLAM

Loop Closure

Introduction

EKF-SLAM

FastSLAM

Loop Closure

The Idea

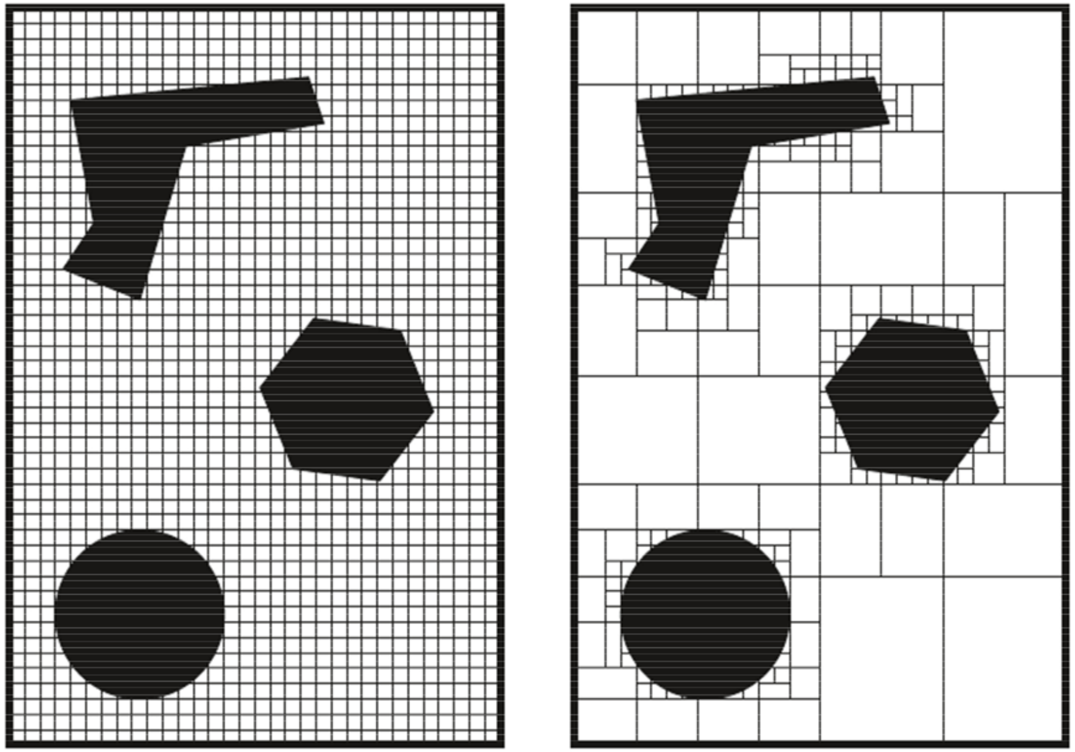
- Basic procedures
  - Map → Localisation
  - Localisation → Map
- No prior knowledge
- SLAM‘s objective: optimum of concurrent localisation and mapping

Typical sensors

- Odometry
  - Encoder in wheels
- Laser scanner
  - 2D: scan in a plane
  - 3D: scans in different planes
- Camera
  - 2D: rgb images
  - 3D: rgb images and depth information

Grid based mapping

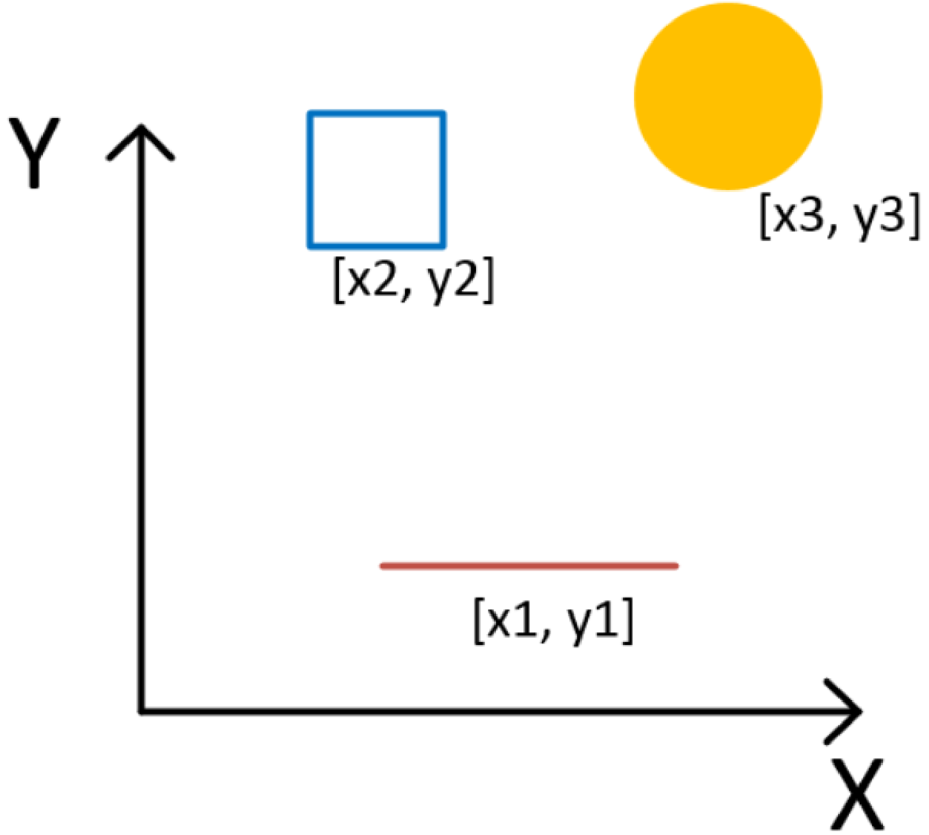
- Environment is divided in grid
  - Value of every cell is the probability of the occupancy of the grid
  - Val = 0 → cell is occupied
  - Val = 1 → cell is free
- Scan-matching via ICP
  - Align scans according to robot position and already existing grid map



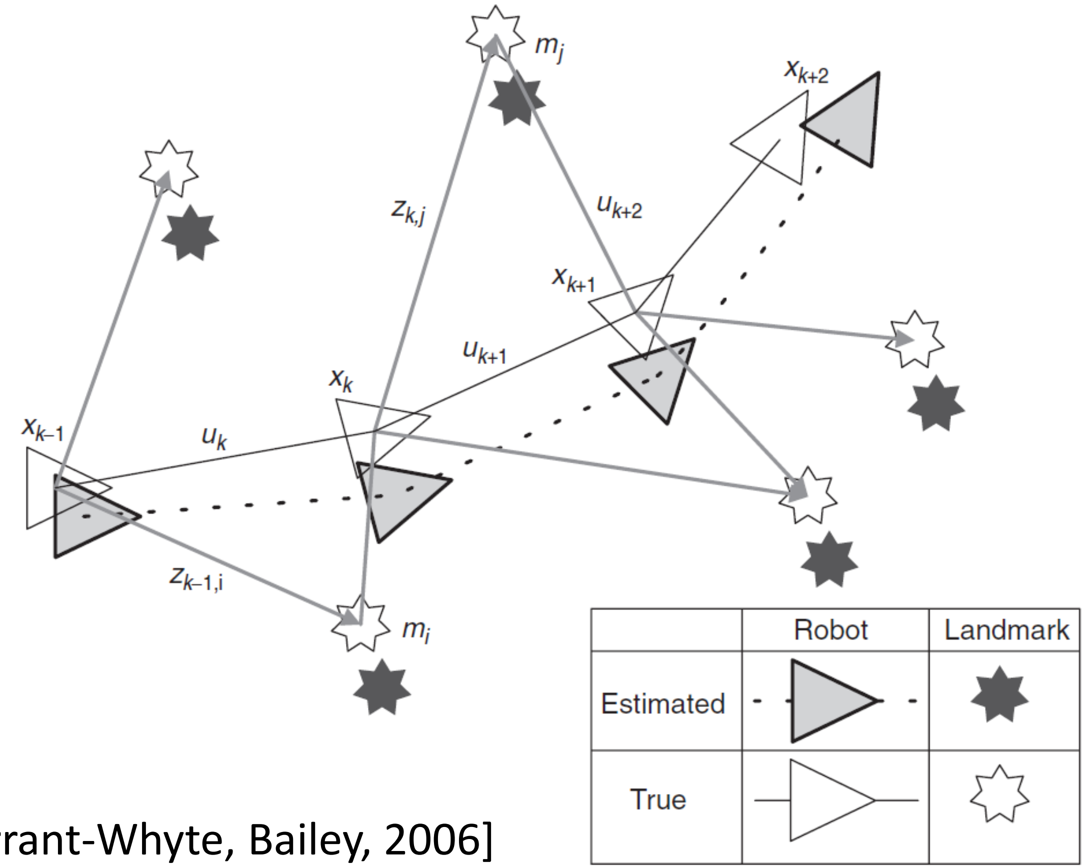
[Hertzberg, Lingemann, Nüchter, 2012]

Feature based map

- Relative position of features are saved in map
- Extracting features of the environment
    - Re-obeservable from different position or angles
    - Unique
    - Stationary
- Example
    - SIFT
    - Hough-Transform



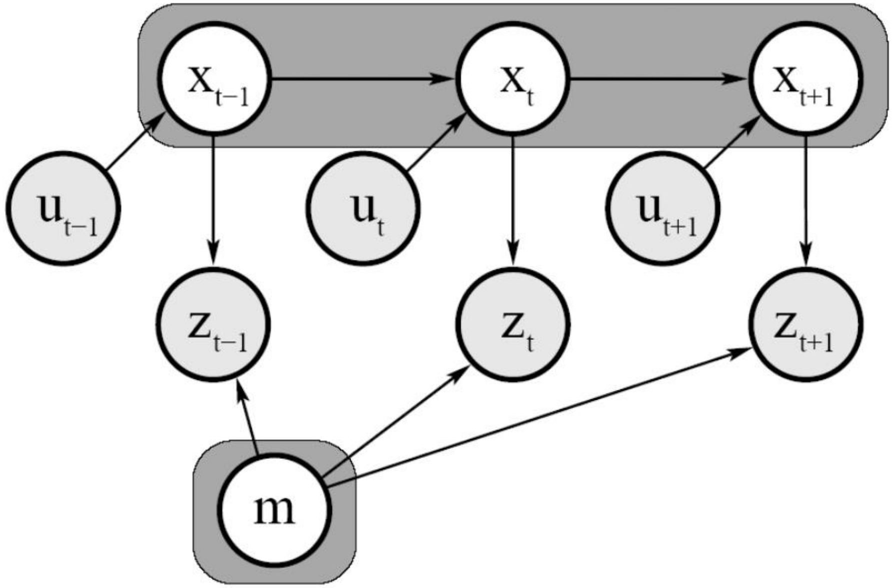
SLAM Scenario



[Durrant-Whyte, Bailey, 2006]

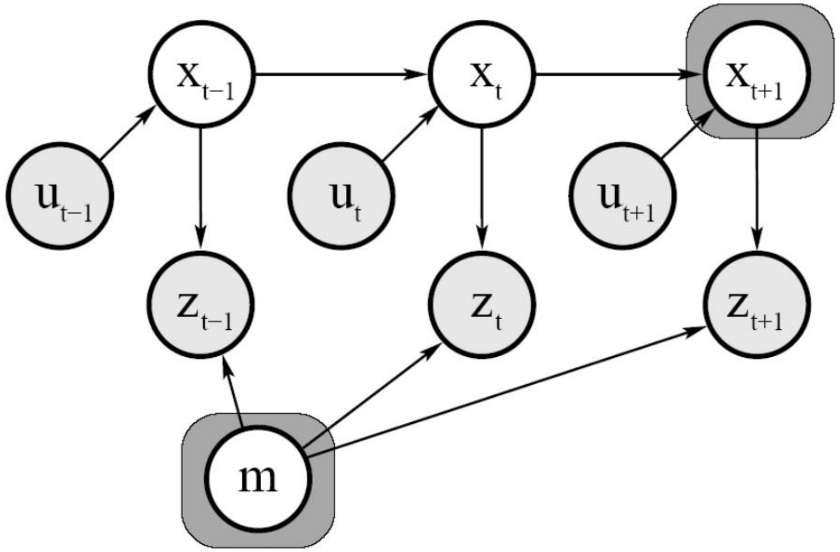
- k: time instant
- x_k: robot location
- u_k: control vector between time $k-1$ and k.
- m: map of landmarks and their respective location
- z_k: „observation“: measurement between robot and landmarks

Full vs Online SLAM?



Full Slam:
Estimation of the entire path:

$$p(x^k, m | z^k, u^k)$$



Online Slam:
Estimation of the most recent pose:

$$p(x_k, m | z^k, u^k)$$

[http://ais.informatik.uni-freiburg.de/teaching/ss12/robotics/slides/12-slam.pdf]

Introduction

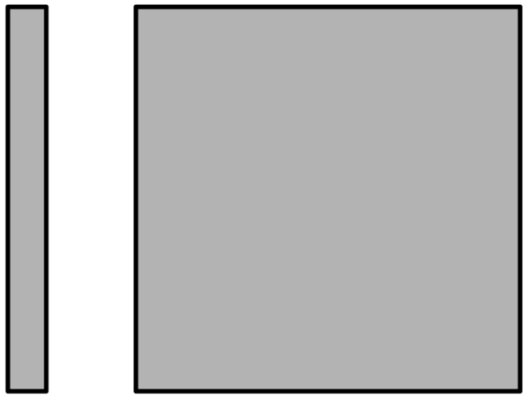
EKF-SLAM

FastSLAM

Loop Closure

EKF-SLAM

- First popular SLAM-approach
- Online SLAM solution
- Based on extended kalman filter



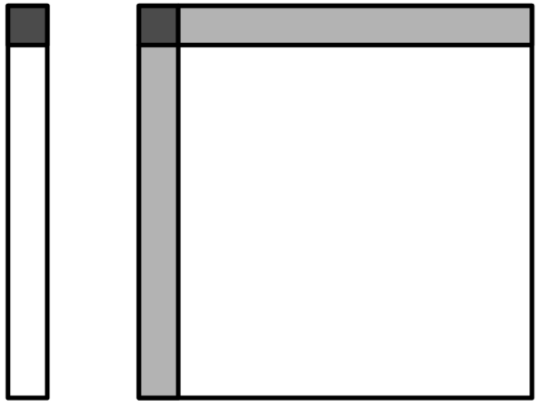
[Solà 2014]

- Sketch
  1. Predict current system state
  2. Correct current system state
  3. Add new landmarks

$$\bar{x} = \begin{bmatrix} \bar{R} \\ \hline LM \end{bmatrix} = \begin{bmatrix} \bar{R} \\ \hline LM_1 \\ \vdots \\ \hline LM_l \end{bmatrix} \quad P = \begin{bmatrix} P_{RR} & P_{RLM} \\ P_{LMR} & P_{LMLM} \end{bmatrix} \quad (2.1)$$

EKF-SLAM
1. Predict current state

- Model of the physical system $f(\,)$
- Robot position will be predicted
  - control vector u
  - Perturbation n, Covariance-matrix N
- Landmarks are static



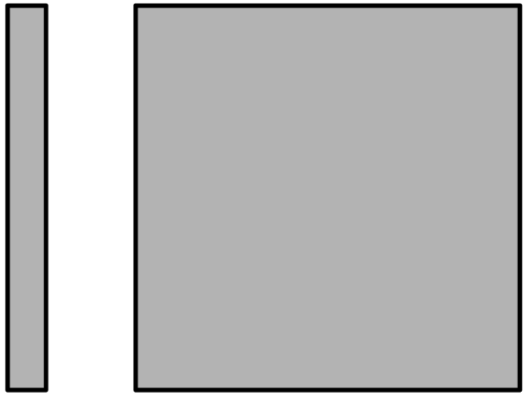
[Solà 2014]

$$R \leftarrow f(R, u, n, N) \qquad (2.2)$$
$$M \leftarrow M \qquad (2.3)$$

EKF-SLAM

2. Correct current state

- Current state will be updated according to obeservation function $h(\,)$
- Measurement
  - Odometry
  - Landmark re-detection
- Covariance-matrix R
- $O(L^2)$

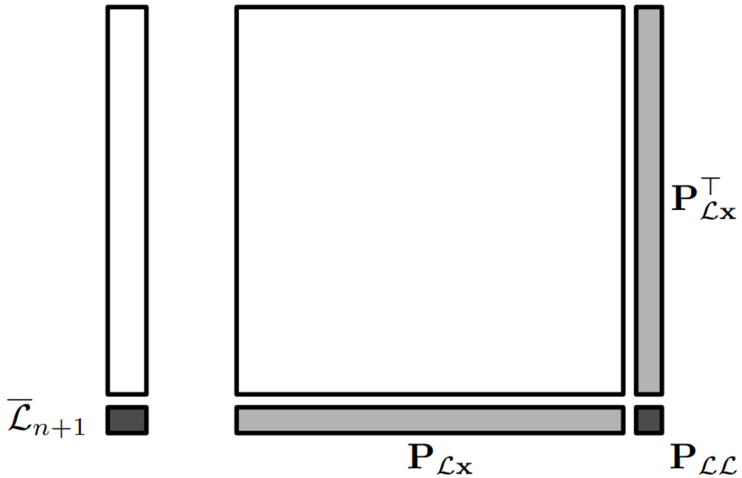


[Solà 2014]

EKF-SLAM

3. Add new landmarks

- New landmarks
- Dynamic system state
- System state grows with the number of landmarks



[Solà 2014]

$$\bar{x} \leftarrow \begin{bmatrix} \bar{x} \\ \hline LM_{L+1} \end{bmatrix} \quad P \leftarrow \begin{bmatrix} P & P_{XLM} \\ P_{LMX} & P_{LMLM} \end{bmatrix} \quad (2.4)$$

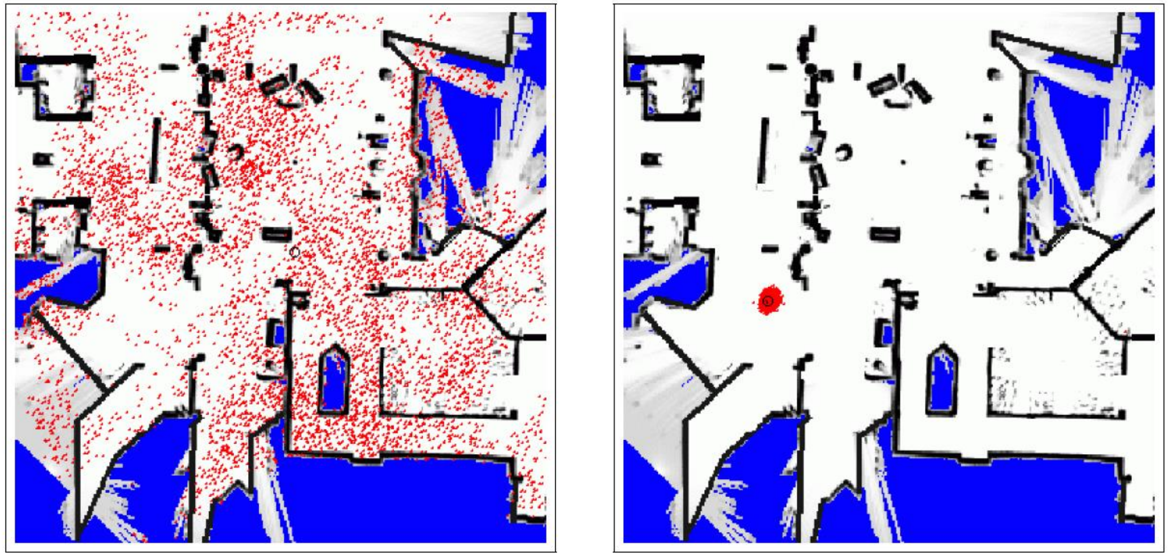
Introduction

EKF-SLAM

FastSLAM

Loop Closure

Particle Filter



[Montemerlo 2003]

- Finite set of samples
- Set represents the probability distribution
- Regions with high particle density → high probability
    1. Moving particles according to system model
    2. Weighting particles according to measurements
    3. Resampling particles
       →deleting particles with low weights
       →add new particles to the higher probable regions

FastSlam [Montemerlo 2003]

- Full SLAM
- Particle filter: path estimaton
- Extended kalman filter: landmark position estimation
- Low cost: O(IL), where I is the number of particles and L the number of landmarks

→Can be used in large environments with up to a million landmarks

FastSlam

Rao-Blackwellization

- Landmark positions and the robots path are treated independently

→ SLAM can be factorized into L+1 estimation problems

Robot position map observations control vector

$$p(x^k, m | z^k, u^k, n^k) = \underbrace{p(x^k | z^k, u^k, n^k)}_{\text{Path estimator}} \underbrace{\prod_l p(m_l | x^k, z^k, u^k, n^k)}_{\text{Landmark pose estimator}}$$

FastSlam

Path prediction via particel filter

- Each particel $s^{k,i} \in S_k$ represents a possible system state.
- Prediction
  - Next robot position in the path is predicted with the system model
- The new particel set is temporary

	Robot Pose	Landmark 1	Landmark 2		Landmark N
Particle 1:	$x \quad y \quad \theta$	$\mu_1 \; \Sigma_1$	$\mu_2 \; \Sigma_2$	...	$\mu_N \; \Sigma_N$
Particle 2:	$x \quad y \quad \theta$	$\mu_1 \; \Sigma_1$	$\mu_2 \; \Sigma_2$	...	$\mu_N \; \Sigma_N$
Particle M:	$x \quad y \quad \theta$	$\mu_1 \; \Sigma_1$	$\mu_2 \; \Sigma_2$	...	$\mu_N \; \Sigma_N$

[Montemerlo 2003]

FastSlam

Landmark location estimation via Kalman Filter

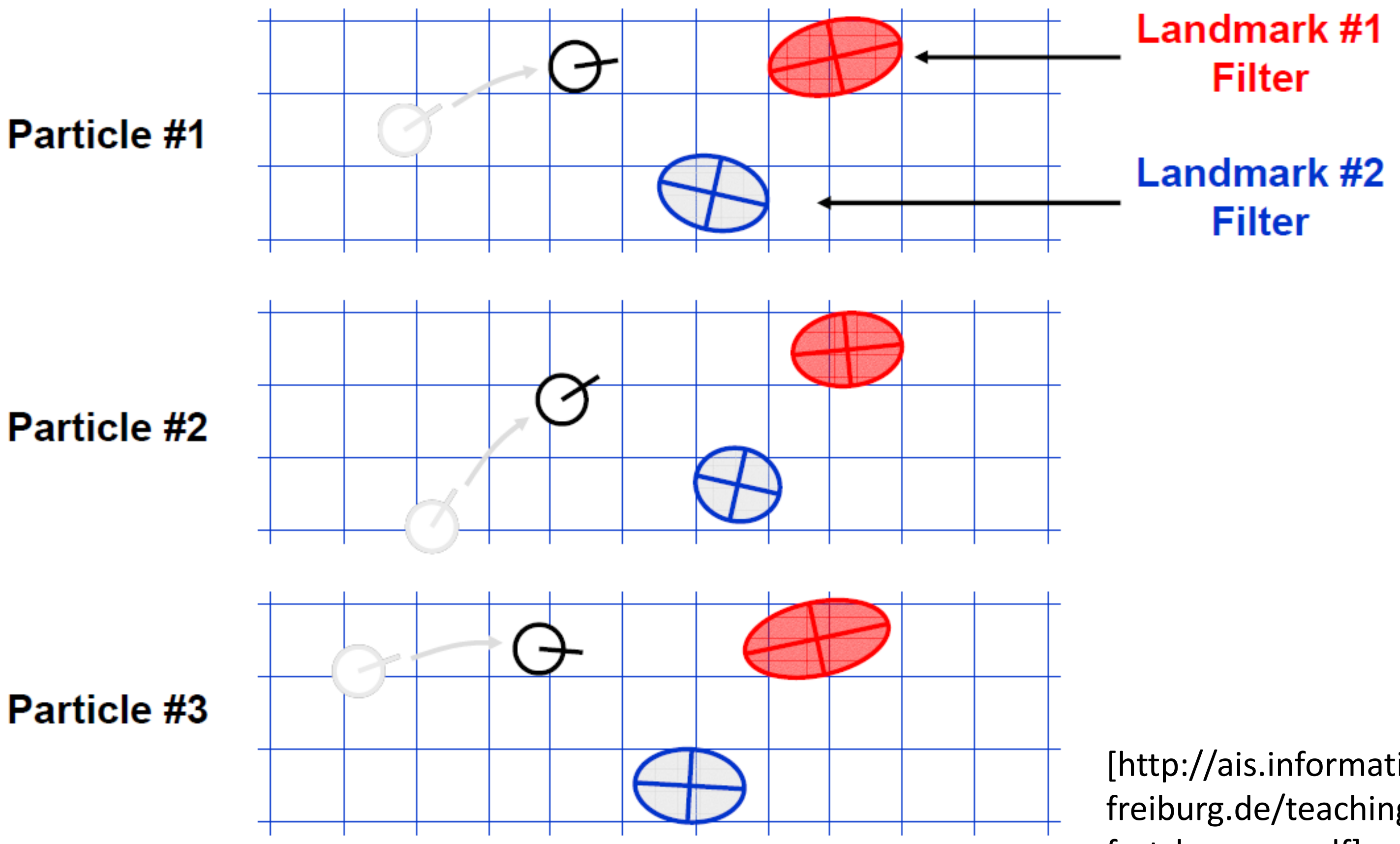
- Landmark map m will be estimated for each particle
- Extended kalman filter
  - Landmarks are treated independently, for every landmark a kalman filter exists. → IL kalman filters in total
  - No prediction step as the position of the landmarks is assumed as static.
  - Mean $\overline{m_l^i}$: 2-element vector
  - Covariance $P_{m_l}^i$: 2-by-2-matrix

FastSlam

weighting and resampling

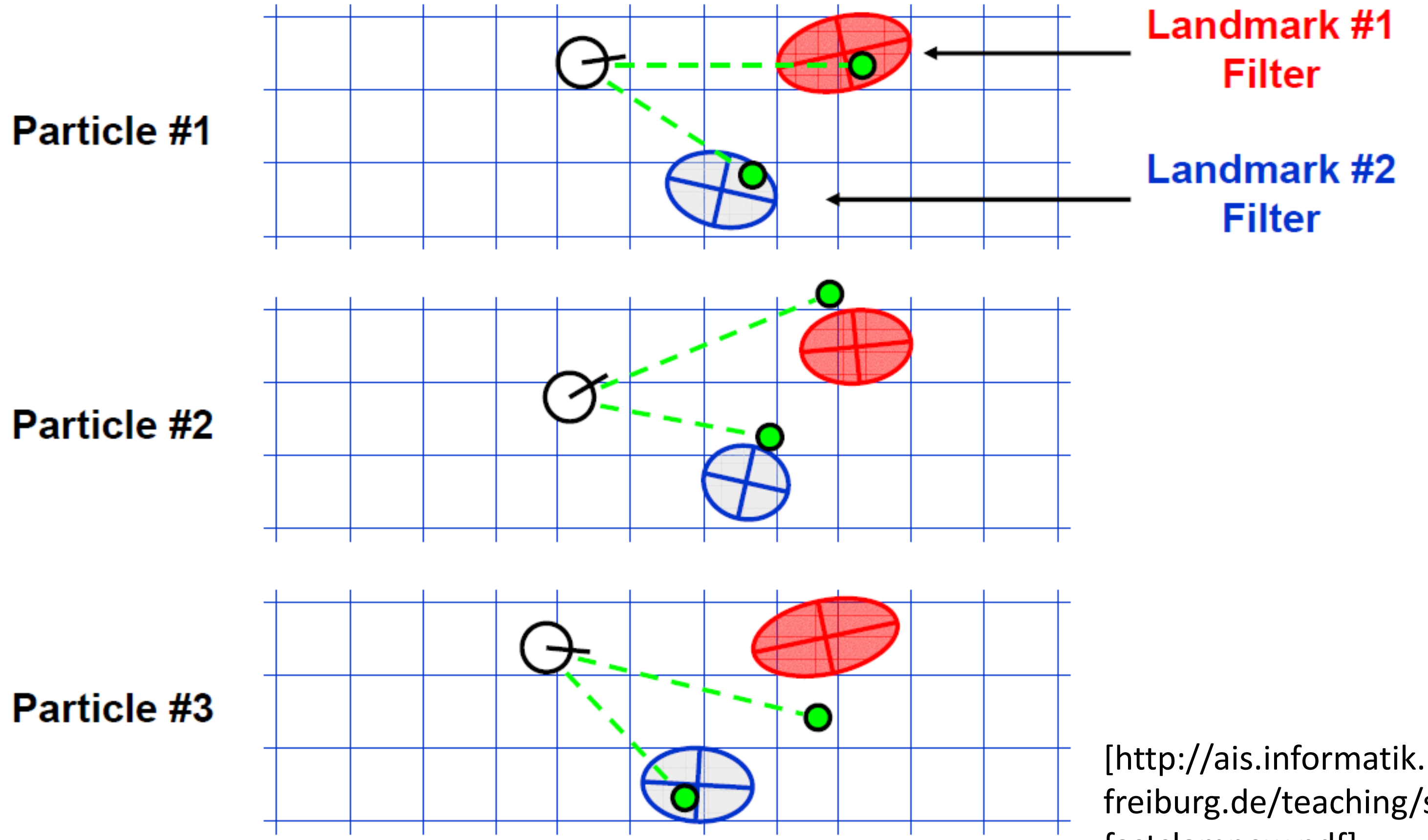
- After the landmark estimation, for each particle an importance factor w_k^i will be determined
  - Measurements
  - Landmark position
  - Temporary path
- Resampling according to importance factor
  - Sampling technique can be chosen variable
  - Most complex part of the procedure O(LI)

FastSlam
Visualization



[http://ais.informatik.uni-freiburg.de/teaching/ss14/robotics/slides/14-slam-fastslamnew.pdf]

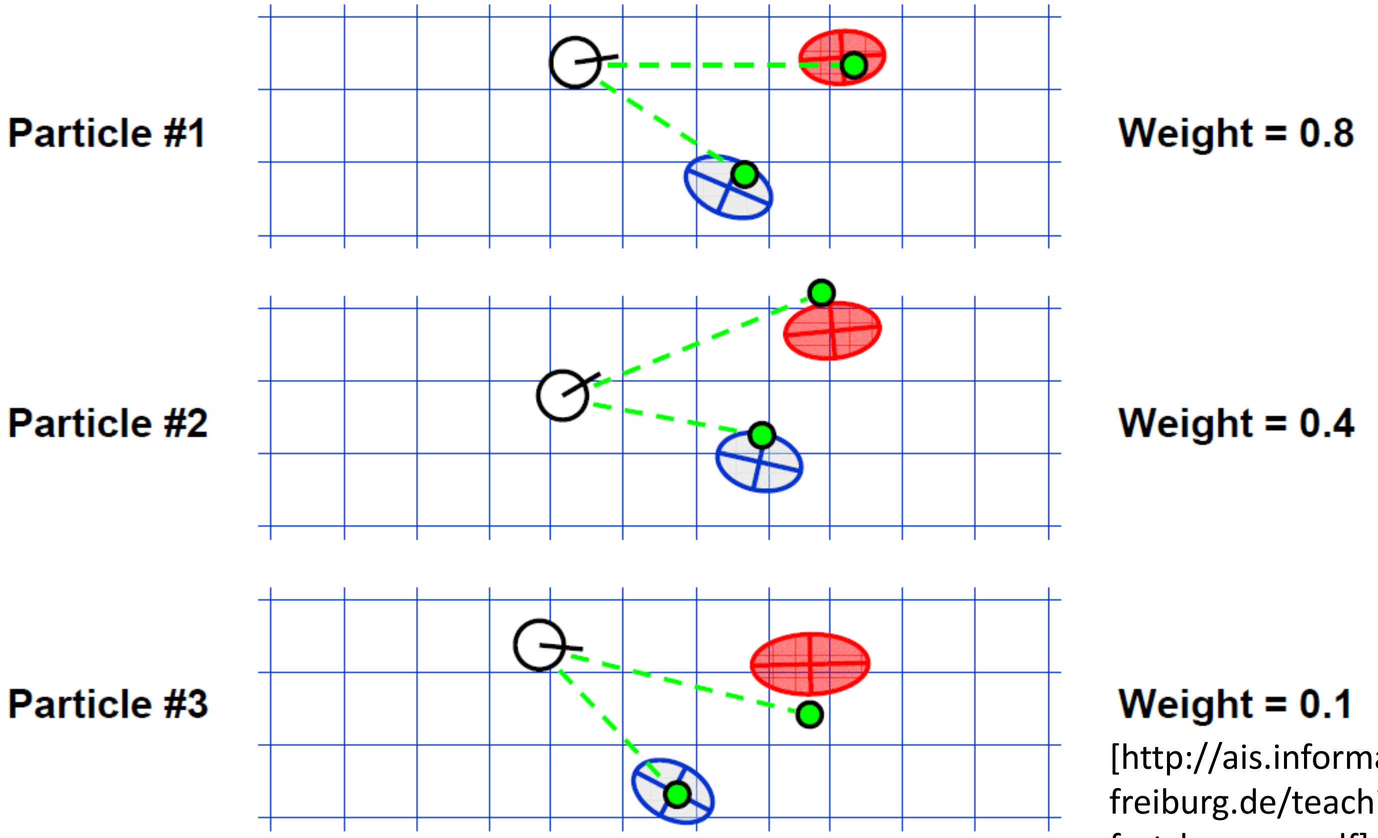
FastSlam
Visualization



[http://ais.informatik.uni-freiburg.de/teaching/ss14/robotics/slides/14-slam-fastslamnew.pdf]

FastSlam

Visualization



[http://ais.informatik.uni-freiburg.de/teaching/ss14/robotics/slides/14-slam-fastslamnew.pdf]

Quick Overview

EKF-SLAM

- Online SLAM
- $O(L^2)$

→for smaller areas

FastSLAM

- Full SLAM
- $O(IL)$ / $O(I\log(L))$

→ usable in larger areas

L: number of landmarks, I: number of particles

Introduction

EKF-SLAM

FastSLAM

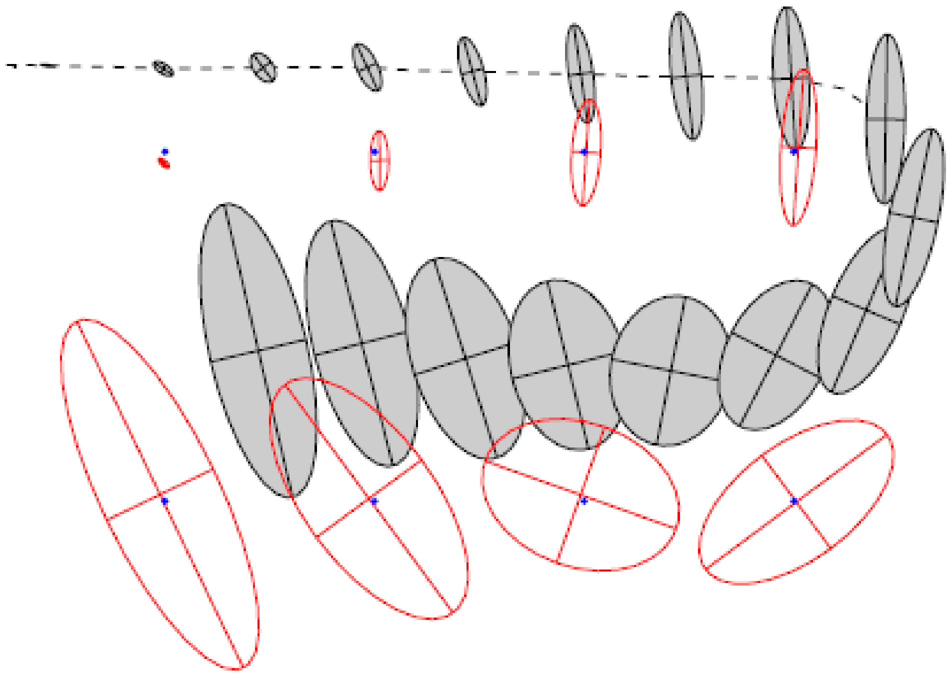
Loop Closure

Loop Closure

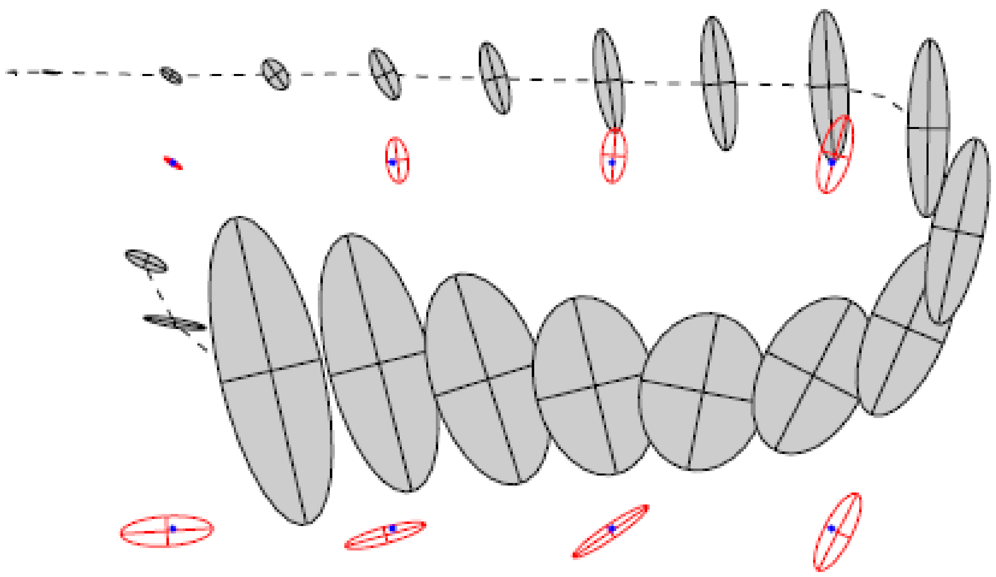
- Recognizing an already mapped area after having explored other regions
- Uncertainties will be reduced
- Robots pose/path and landmarks, all landmark position will be corrected and get more accurate
- Challenge: Reliable data association
  - →wrong loop closure lead to big errors

Loop Closure

Example

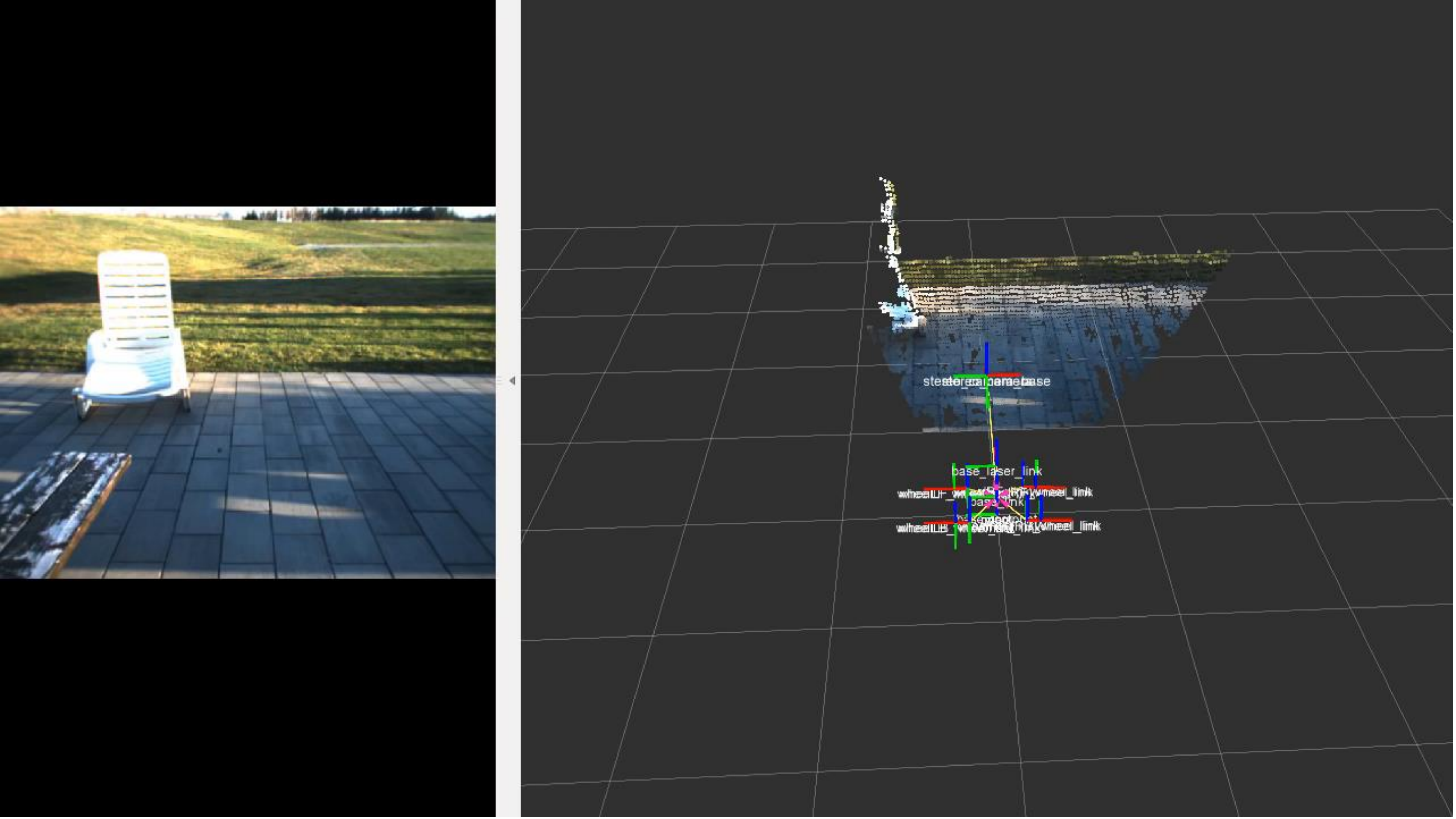


Before closing loop



After closing loop

[Montemerlo 2003]



[https://www.youtube.com/watch?v=qpTS7kg9J3A]

References

[Durrant-Whyte, Bailey, 2006] Hugh Durrant-Whyte, Tim Bailey. Simultaneous Localization and Mapping: Part 1. IEEE, 2006.

[Hidalgo, Bräunl, 2015] Franco Hodalgo, Thomas Bräunl. Review of Underwater SLAM Techniques. IEEE, 2015.

[Hertzberg, Lingemann, Nüchter, 2012] J. Hertzberg, K. Lingemann, A. Nüchter. Mobile Roboter – Eine Einführung aus der Sicht der Informatik. 1. Auflage. Springer-Verlag, 2012.

[Montemerlo 2003] M. Montemerlo. FastSLAM: A Factored Soltution to the Simultaneous Localization and Mapping Problem With Unknown Data Association. 2003.

[Solà 2014] J. Solà. Simultaneous localization and mapping with extended Kalman filter – A very quick guide... with Matlab code!. 2014.