Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

https://www.space.com/perseverance-rover-mars-2020-mission

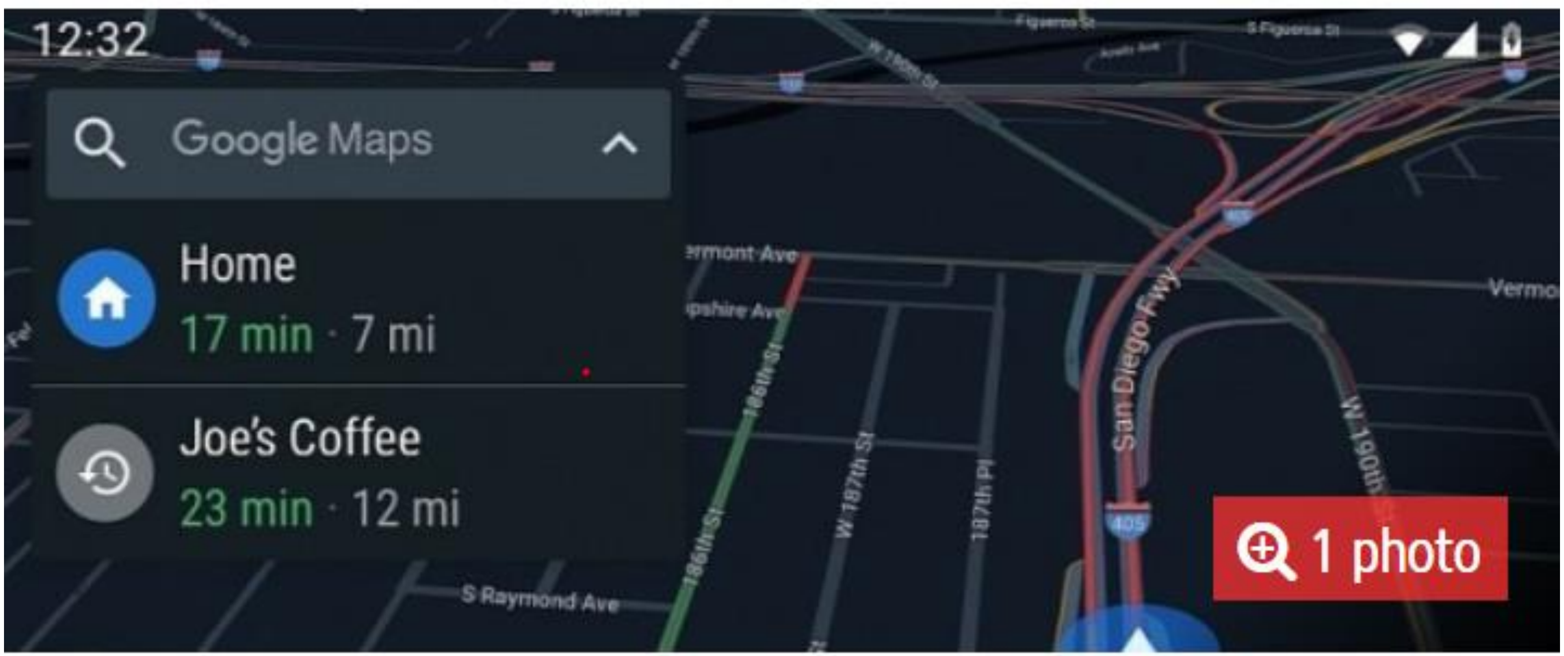
Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

Google Maps Getting Users' Locations All Wrong, Showing Them in Another Country

Home > News > Technology

11 Oct 2020, 7:42 UTC · by **Bogdan Popa**

When it works, Google Maps is a super-advanced tool, especially when it comes to navigation, both on the head unit in a car and on the phone.



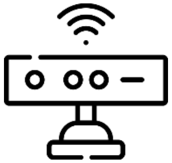
https://www.autoevolution.com/news/google-maps-getting-users-locations-all-wrong-showing-them-in-another-country-149978.html

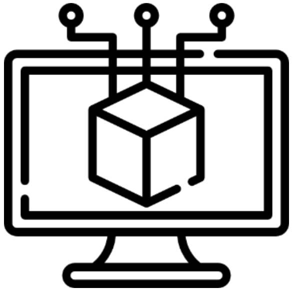
Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

Autonomous Driving Software Engineering

Prof. Dr.-Ing. Markus Lienkamp

Phillip Karle, M. Sc.





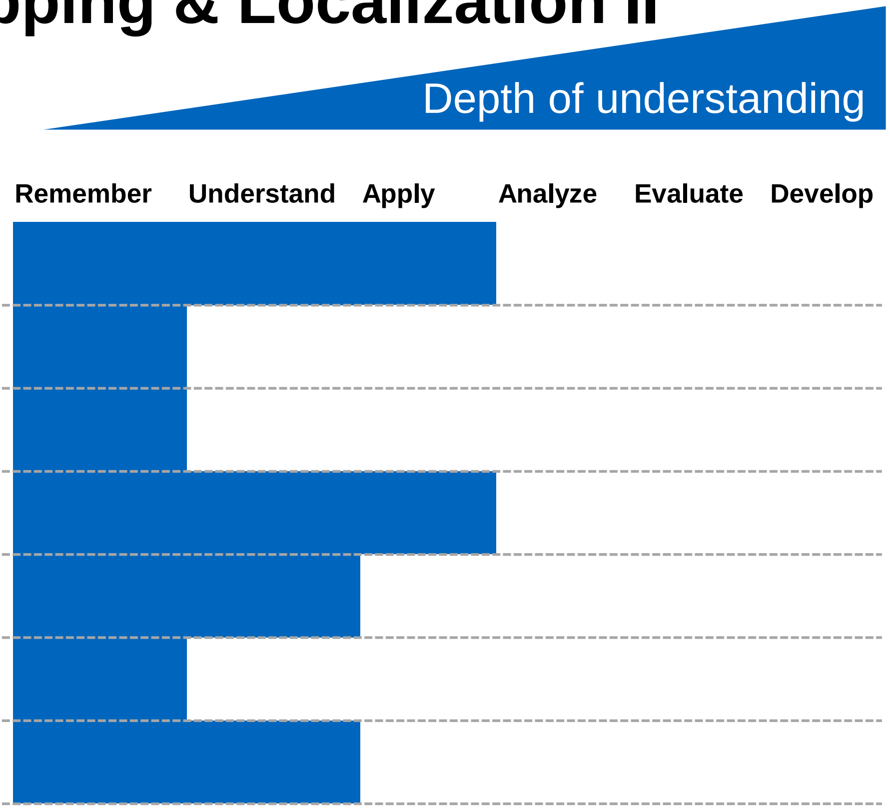
Lecture Overview

Lecture – 90min	Practice – 45min
1 Introduction: Autonomous Driving Karle	**1 Practice** Karle
2 Perception I: Localization & Mapping I Sauerbeck	**2 Practice** Sauerbeck
3 Perception II: Localization & Mapping II Sauerbeck	**3 Practice** Sauerbeck
4 Perception III: Detection Huch	**4 Practice** Huch
5 Prediction Karle	**5 Practice** Karle
6 Planning I: Global Planning Trauth	**6 Practice** Trauth
7 Planning II: Local Planning Ögretmen	**7 Practice** Ögretmen
8 Control 15.06.2021 – Wischnewski	**8 Practice** Wischnewski
9 Safety Assessment Stahl	**9 Practice** Stahl
10 Teleoperated Driving Feiler	**10 Practice** Feiler
11 End-to-End Betz	**11 Practice** Betz
12 From Driver to Passenger Fank	**12 Practice** Karle

Objectives for Lecture 3: Mapping & Localization II

After the lecture you are able to…

- … know why which sensors are used for localization, analyze upsides and downsides and how to combine them
- … remember different approaches of SLAM on mathematical level
- ... remember the different approaches of SLAM on sensor level (LiDAR/camera/radar)
- … utilize existing SLAM tools and know how to set them up
- … understand the basic SLAM problem and analyze an exemplary
- ... remember the different solutions to SLAM problem and how they are approached
- … understand the need and concepts of sensor fusion for SLAM



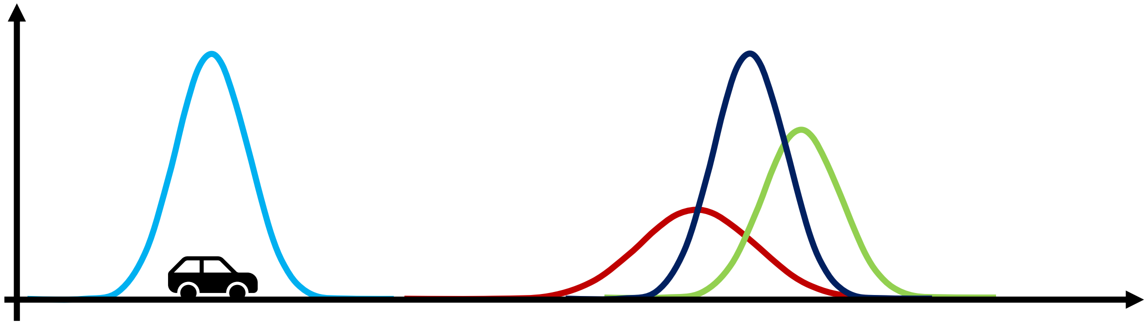
Mapping & Localization II
Prof. Dr. Markus Lienkamp

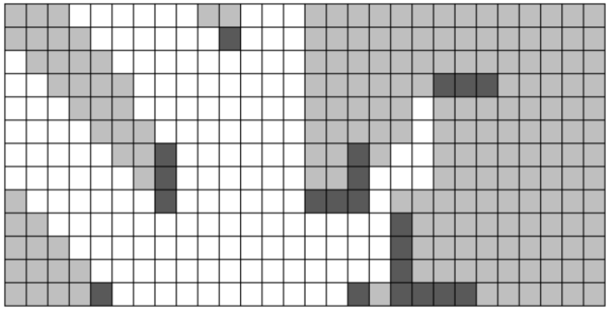
Florian Sauerbeck, M. Sc.

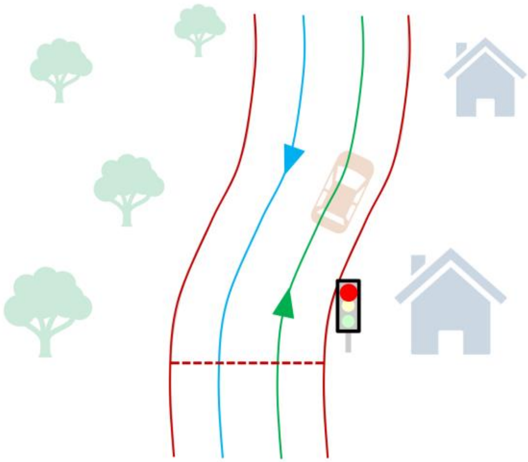
Agenda

1. **Previously …**
2. The SLAM Problem
3. The SLAM Algorithm
4. SLAM Paradigms
   1. EKF SLAM
   2. Particle Filter SLAM
   3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary

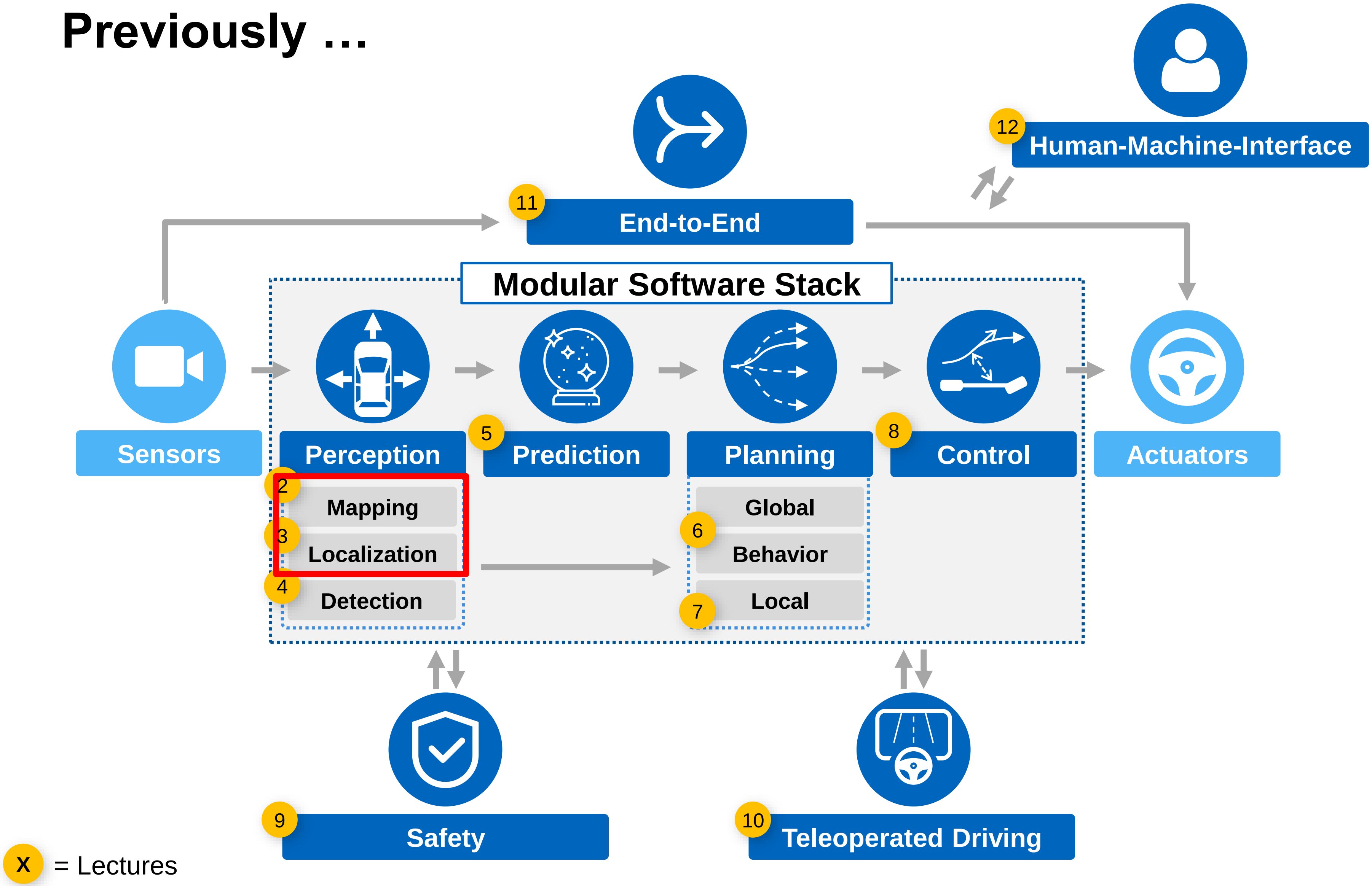
Previously …







Previously …



12
Human-Machine-Interface
11
End-to-End
Modular Software Stack
Sensors
Perception
5
Prediction
Planning
8
Control
Actuators
2
Mapping
3
Localization
4
Detection
Global
6
Behavior
Local
7
9
Safety
10
Teleoperated Driving
X
= Lectures

Previously …

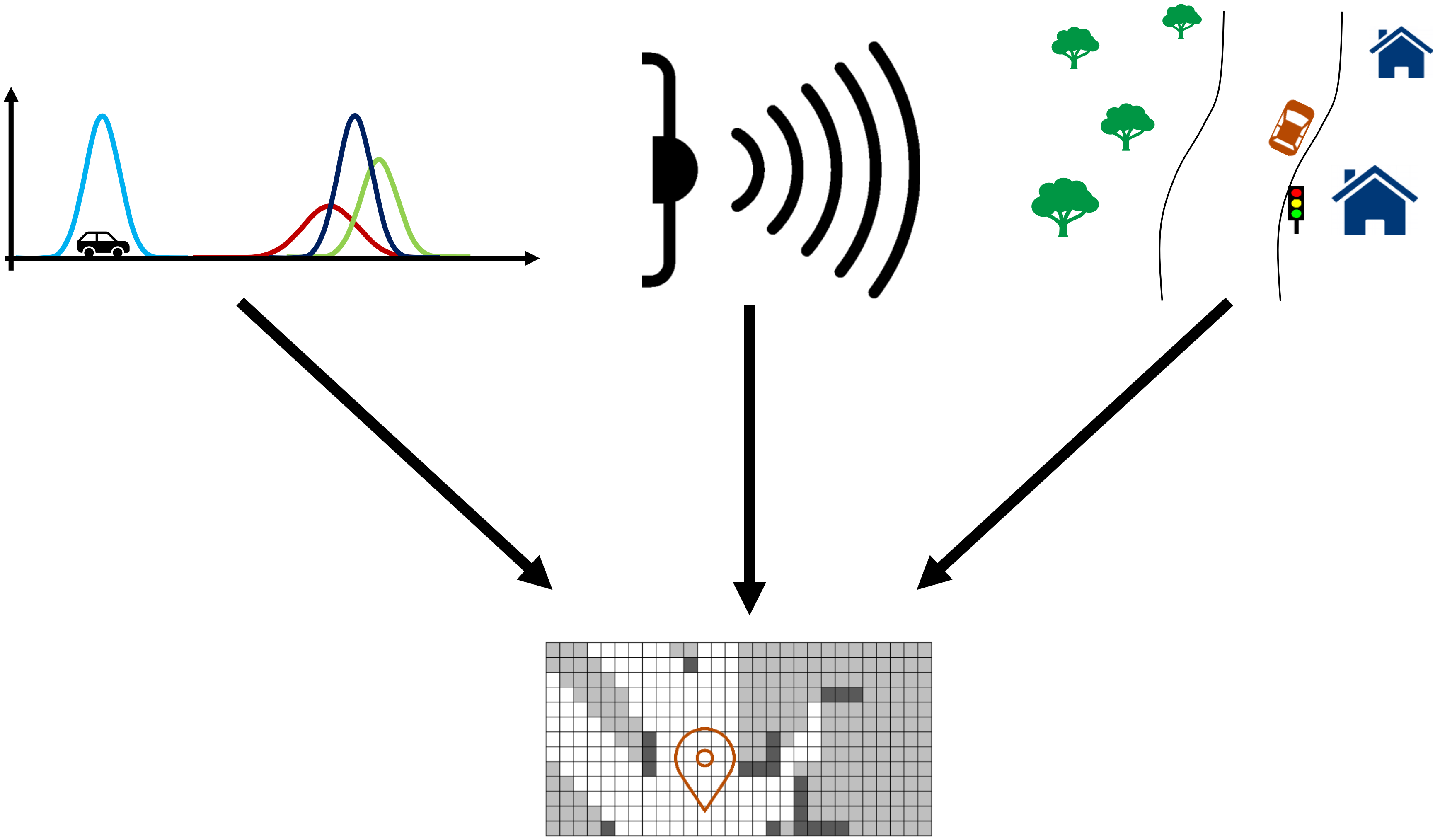
Mapping and localization are crucial for autonomous driving

Sensor measurements must be handled and combined as gaussian distributions

Bayesian filters give us tools to do this

Different map representations for different algorithms and applications

… Today



… Today

How can we use environmental information for localization and mapping?

What was there first: The pose or the map?

How do different algorithms tackle the localization and mapping problem?

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. **The SLAM Problem**
3. The SLAM Algorithm
4. SLAM Paradigms
    1. EKF SLAM
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary

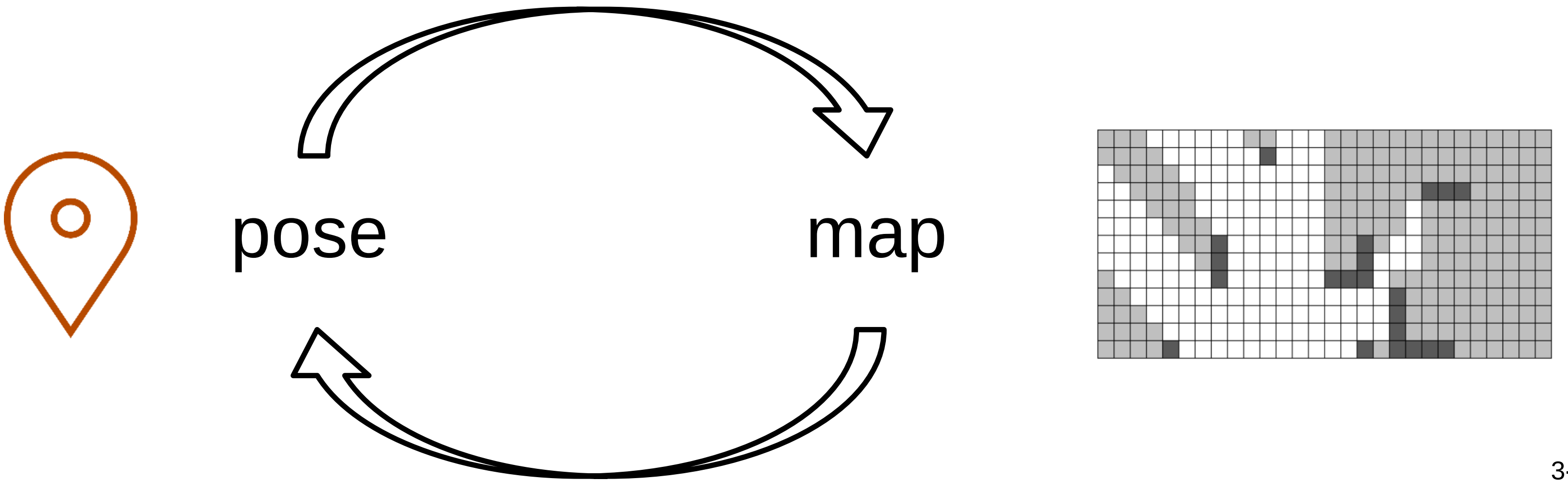

The SLAM problem

SLAM is a chicken-or-egg-problem

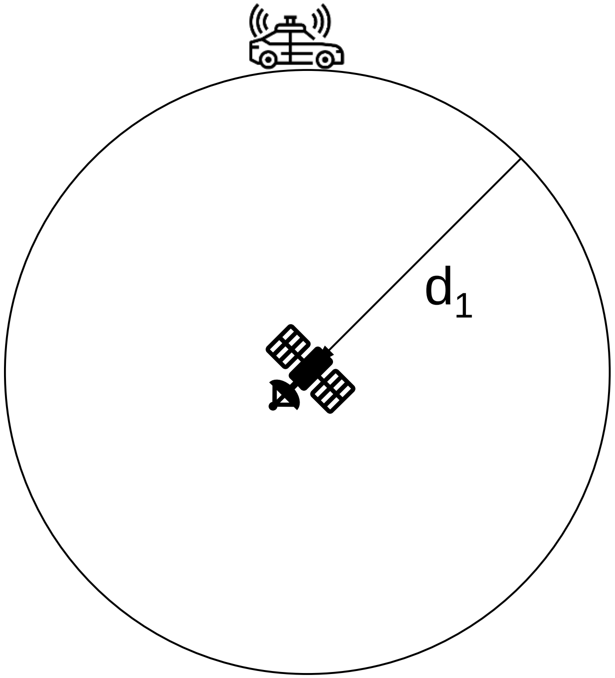
➙ The position is needed for map generation

➙ A map is needed for estimating the position

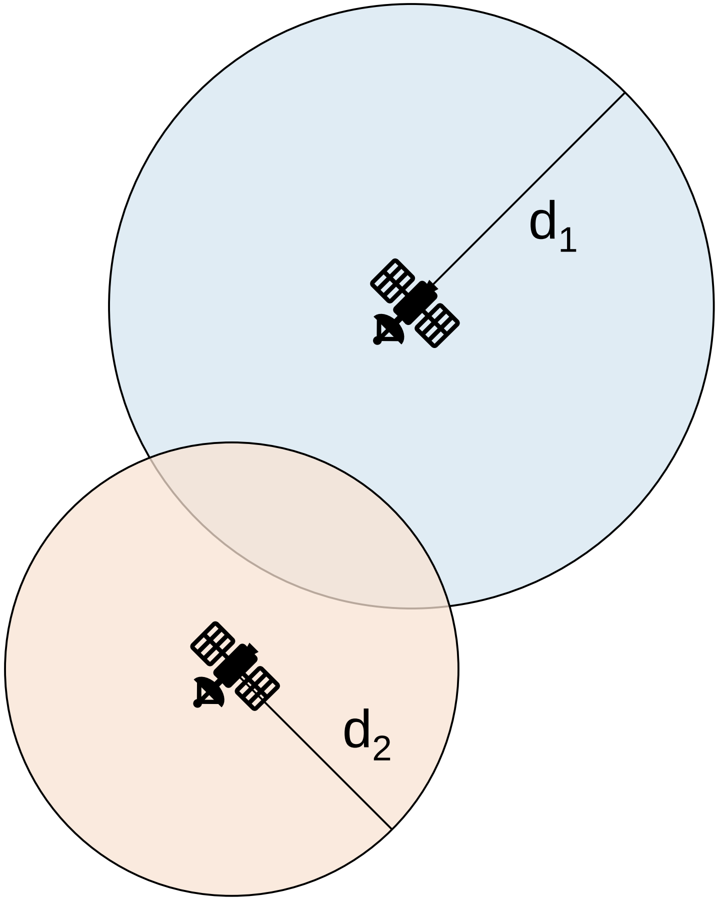
Simultaneous **L**ocalization **A**nd **M**apping



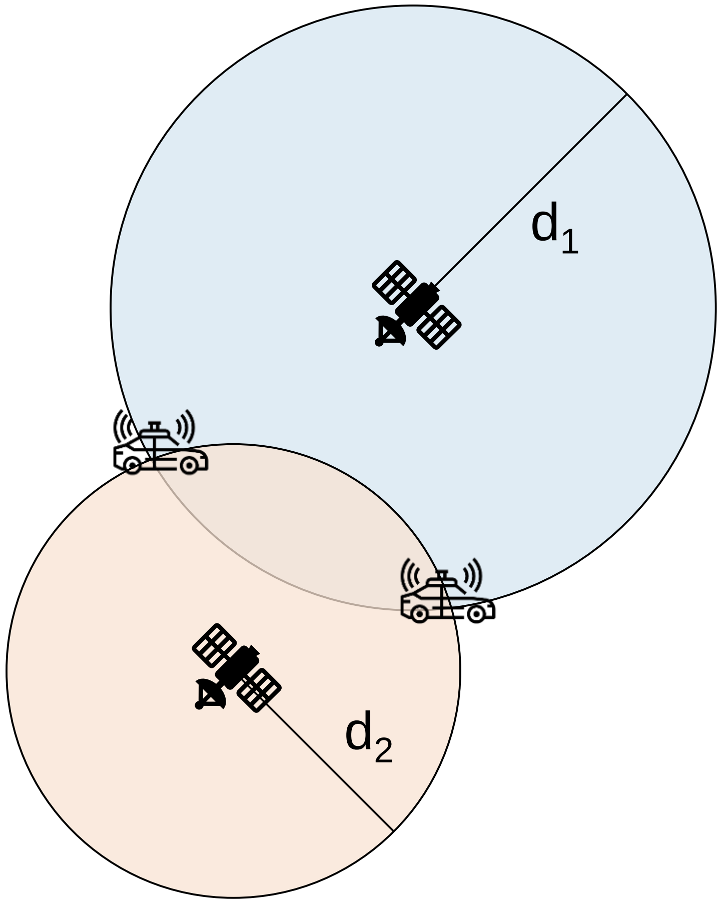
The SLAM problem: Digression



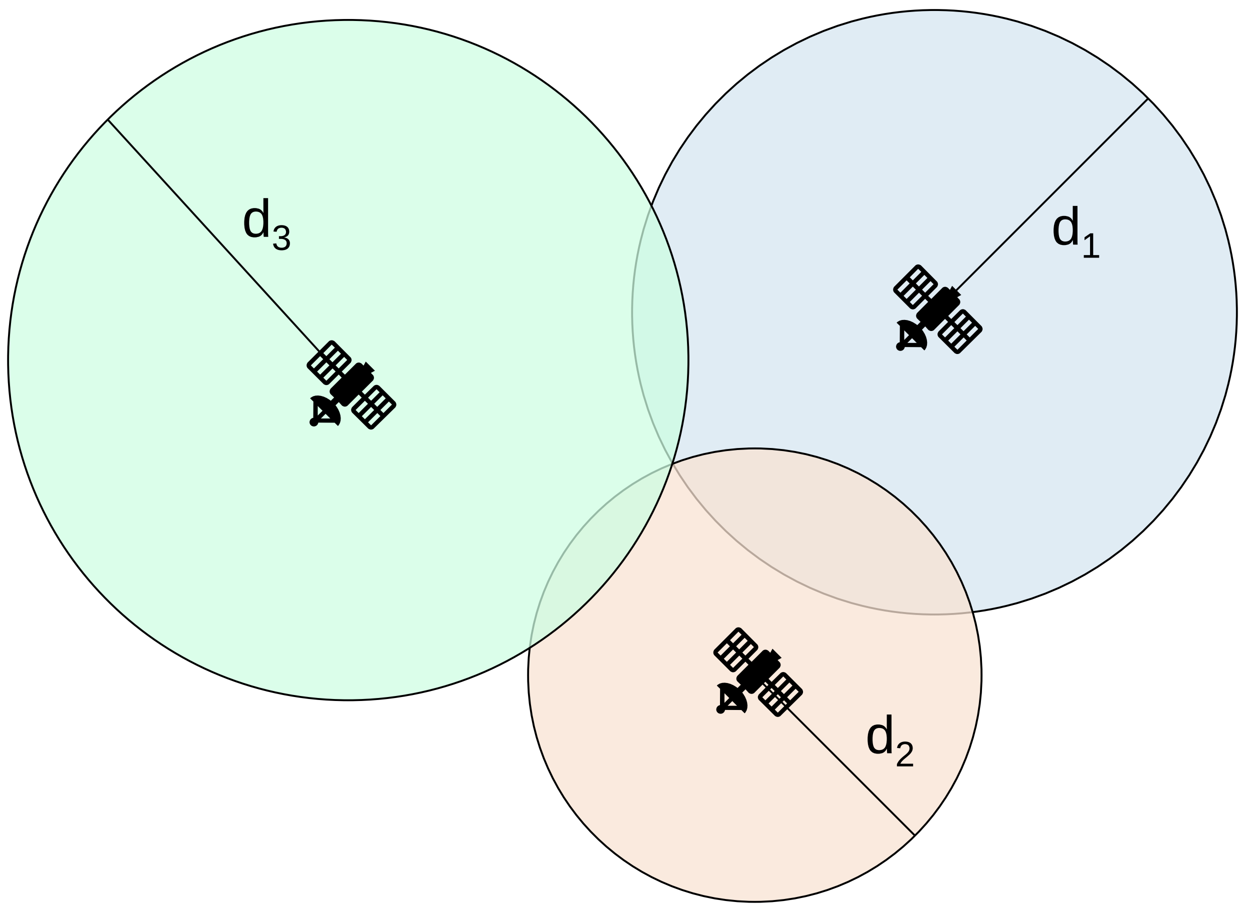
The SLAM problem: Digression



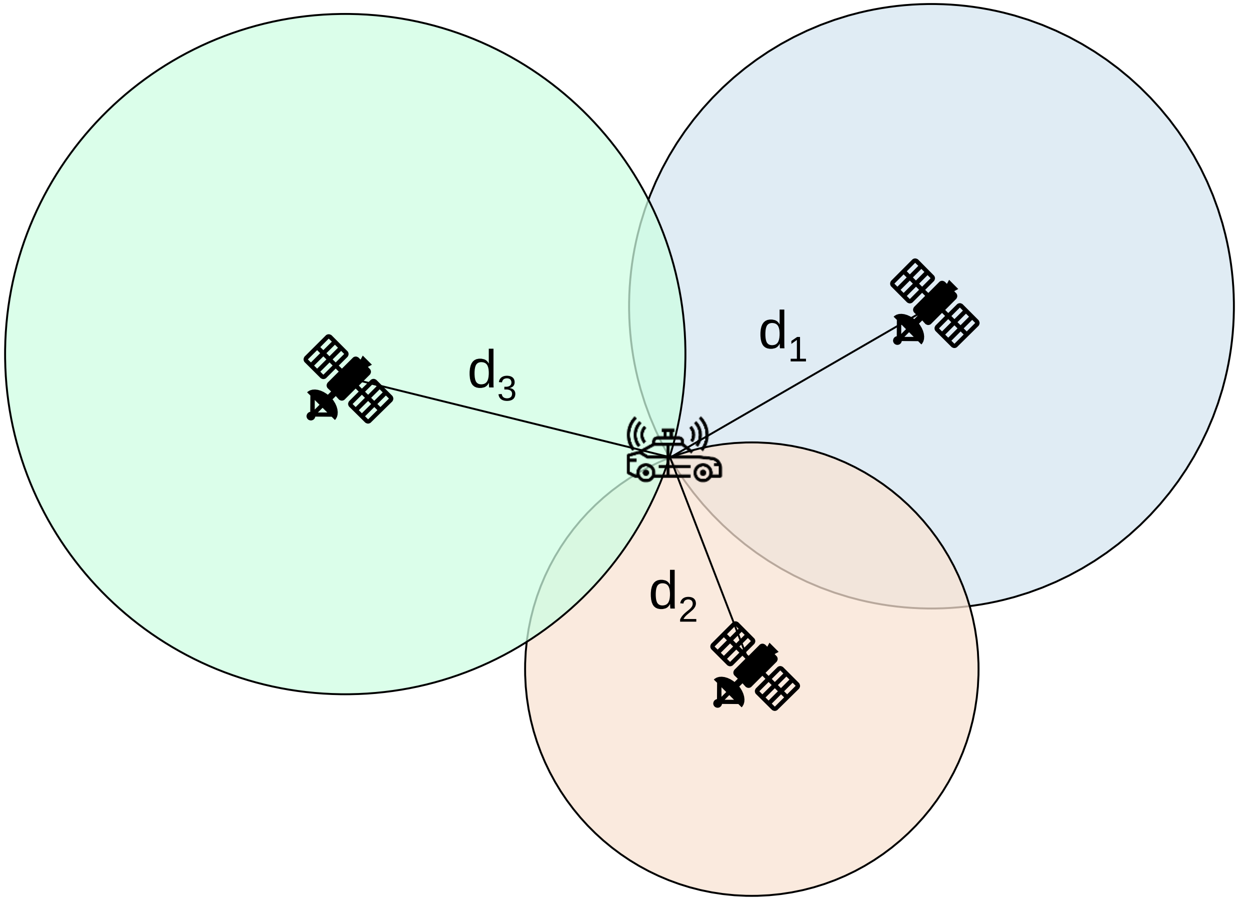
The SLAM problem: Digression



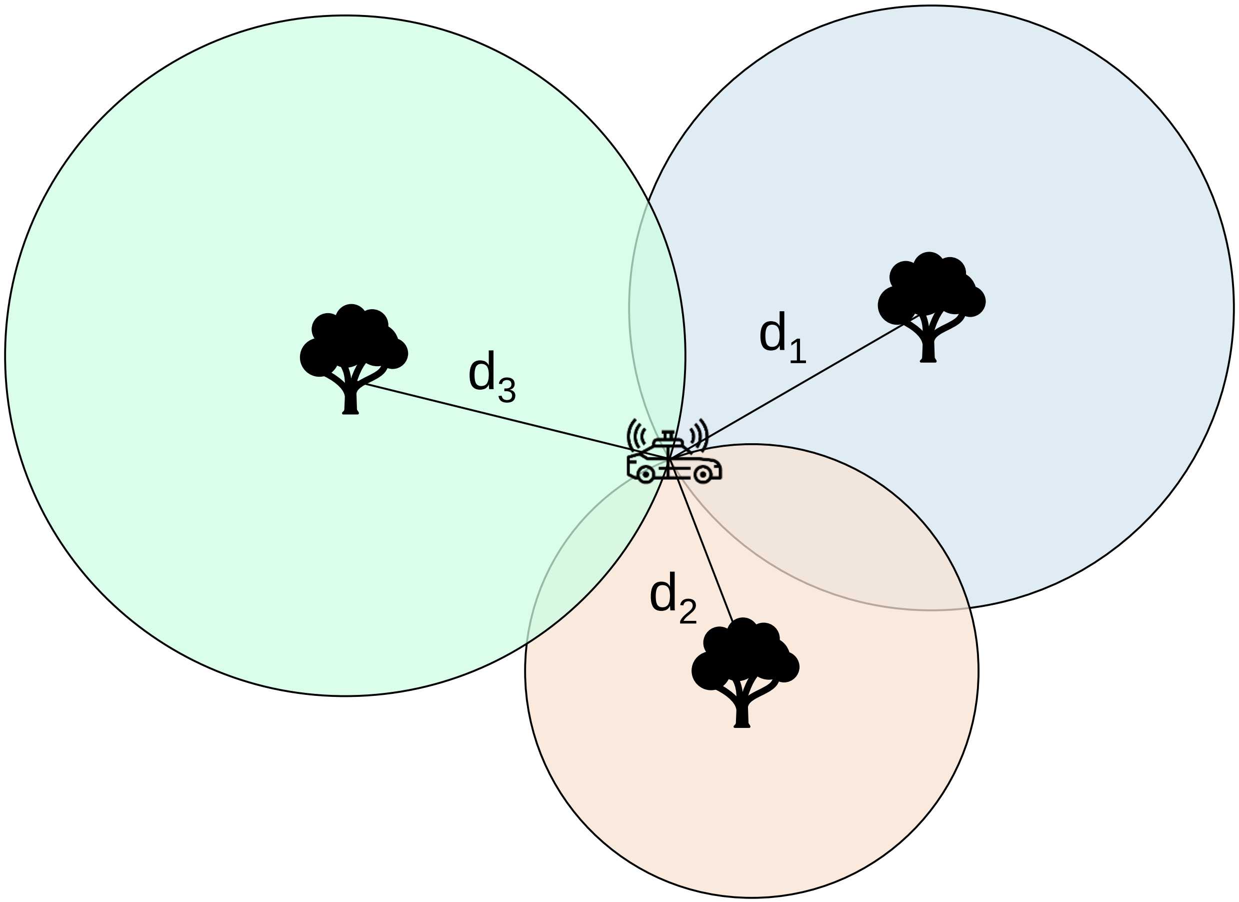
The SLAM problem: Digression



The SLAM problem: Digression



The SLAM problem – From GNSS to SLAM



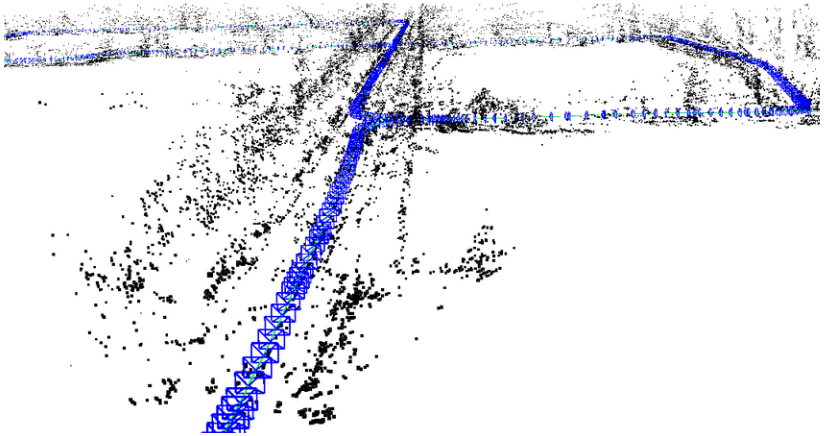
The SLAM problem

Definition:

Feature	A clear-cut attribute in the data (camera or LiDAR) that can be extracted by algorithms
Landmark	Points from the environment that are recognized in different frames

The SLAM problem

Feature **vs** **Landmark**



https://www.researchgate.net/publication/320180254_Ultra-Wideband_Aided_Fast_Localization_and_Mapping_System/figures?lo=1

The SLAM problem

Given:

Ego controls $U_T = \{u_1, u_2, u_3, \dots, u_\mathrm{T}\}$

accelerator pedal position, steering wheel angle, etc.

Measurements $Z_T = \{z_1, z_2, \mathrm{z}_3, \dots, z_\mathrm{T}\}$

IMU, GNSS, Camera, LiDAR, etc.

Wanted:

Path $\mathrm{X}_T = \{x_1, x_2, x_3, \dots, x_\mathrm{T}\}$

Map $M_T = \{m_1, m_2, m_3, \dots, m_T\}$

The SLAM problem

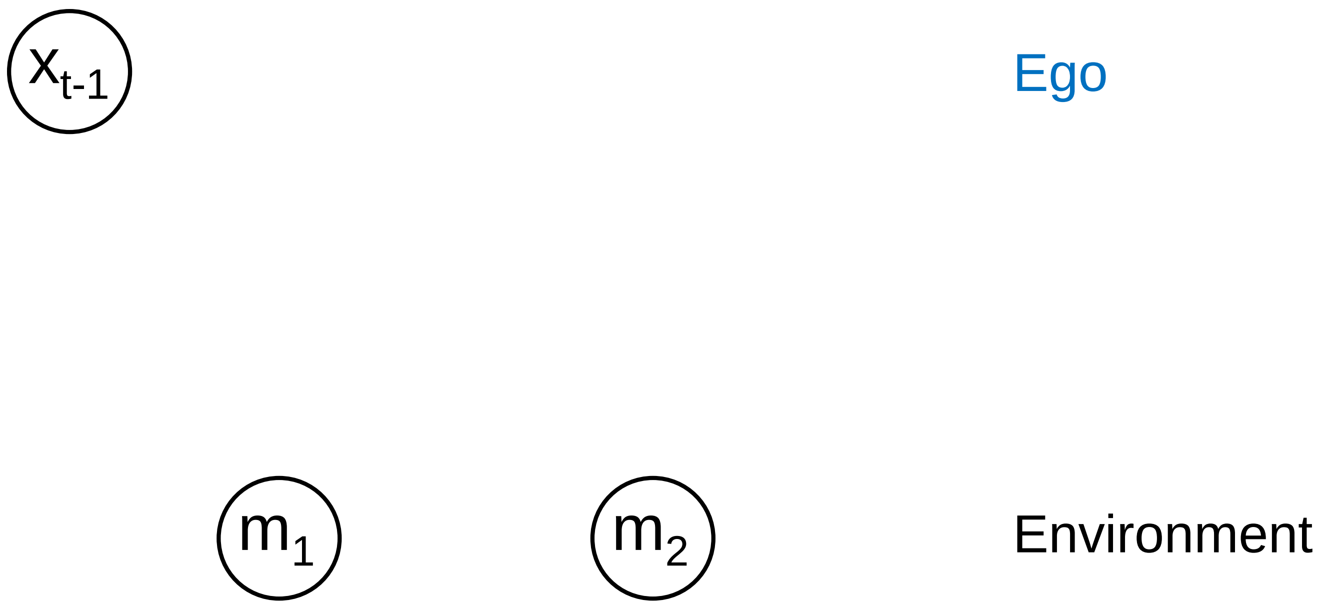
Given:

Ego controls	$U_T = \{u_1, u_2, u_3, \dots, u_T\}$
Measurements	$Z_T = \{z_1, z_2, \mathrm{z}_3, \dots, z_T\}$

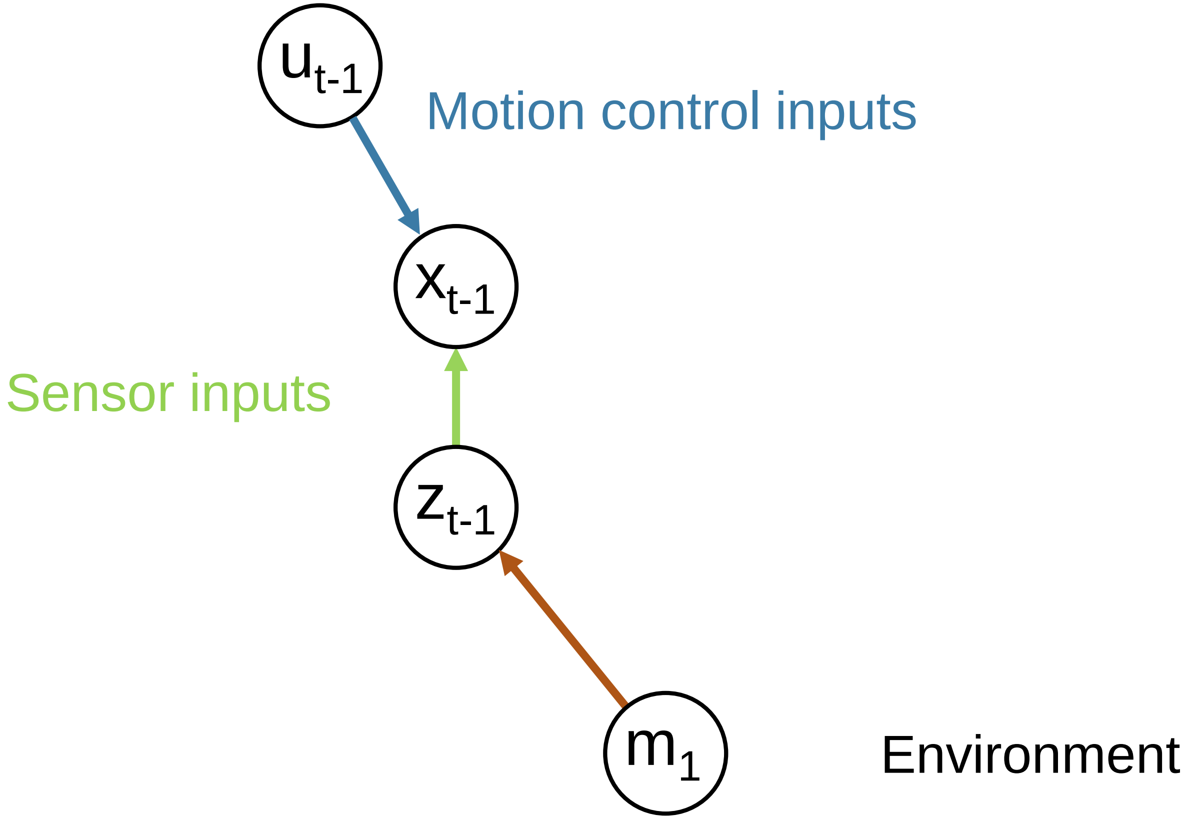
Wanted:

Path	$\mathrm{X}_T = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots, \mathrm{x}_T\}$	Bayesian Filter
Map	$M_T = \{m_1, m_2, \mathrm{m}_3, \dots, \mathrm{m}_T\}$	?

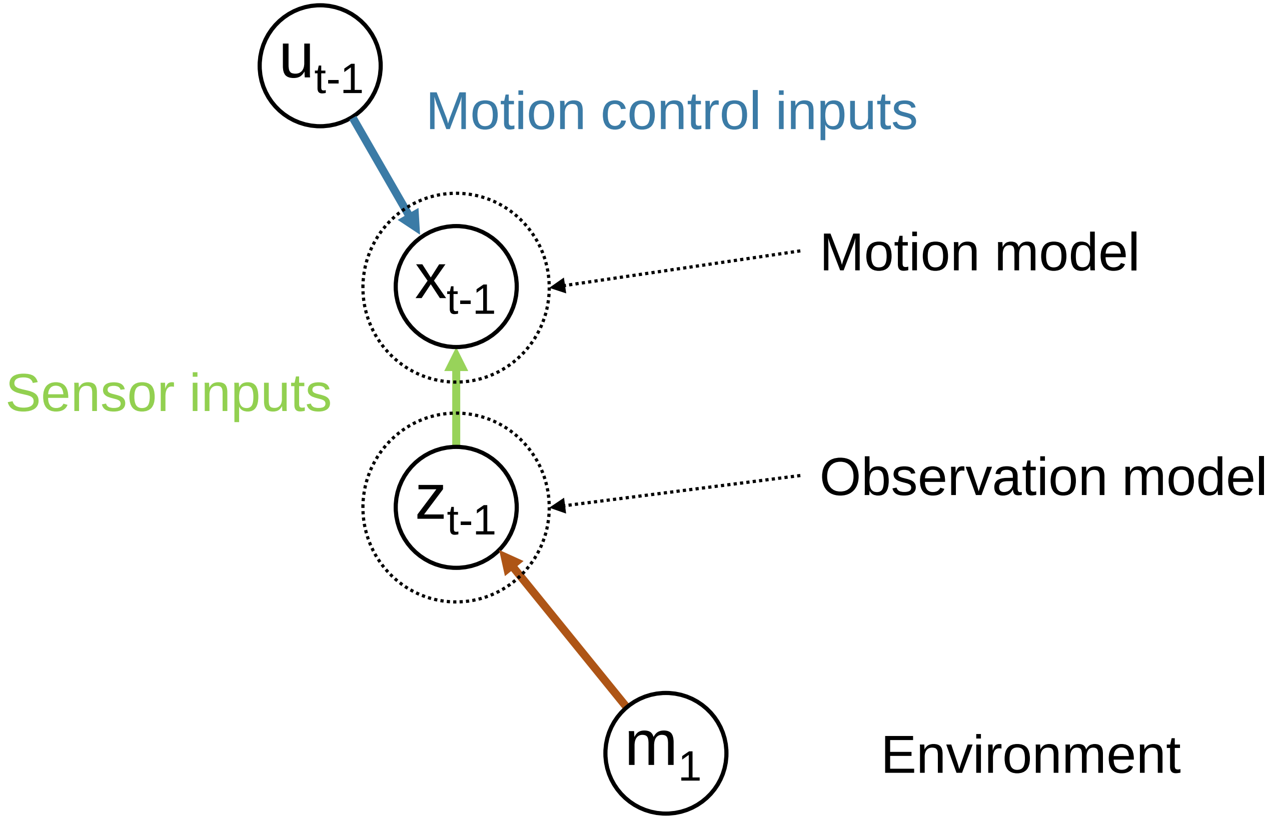
The SLAM problem



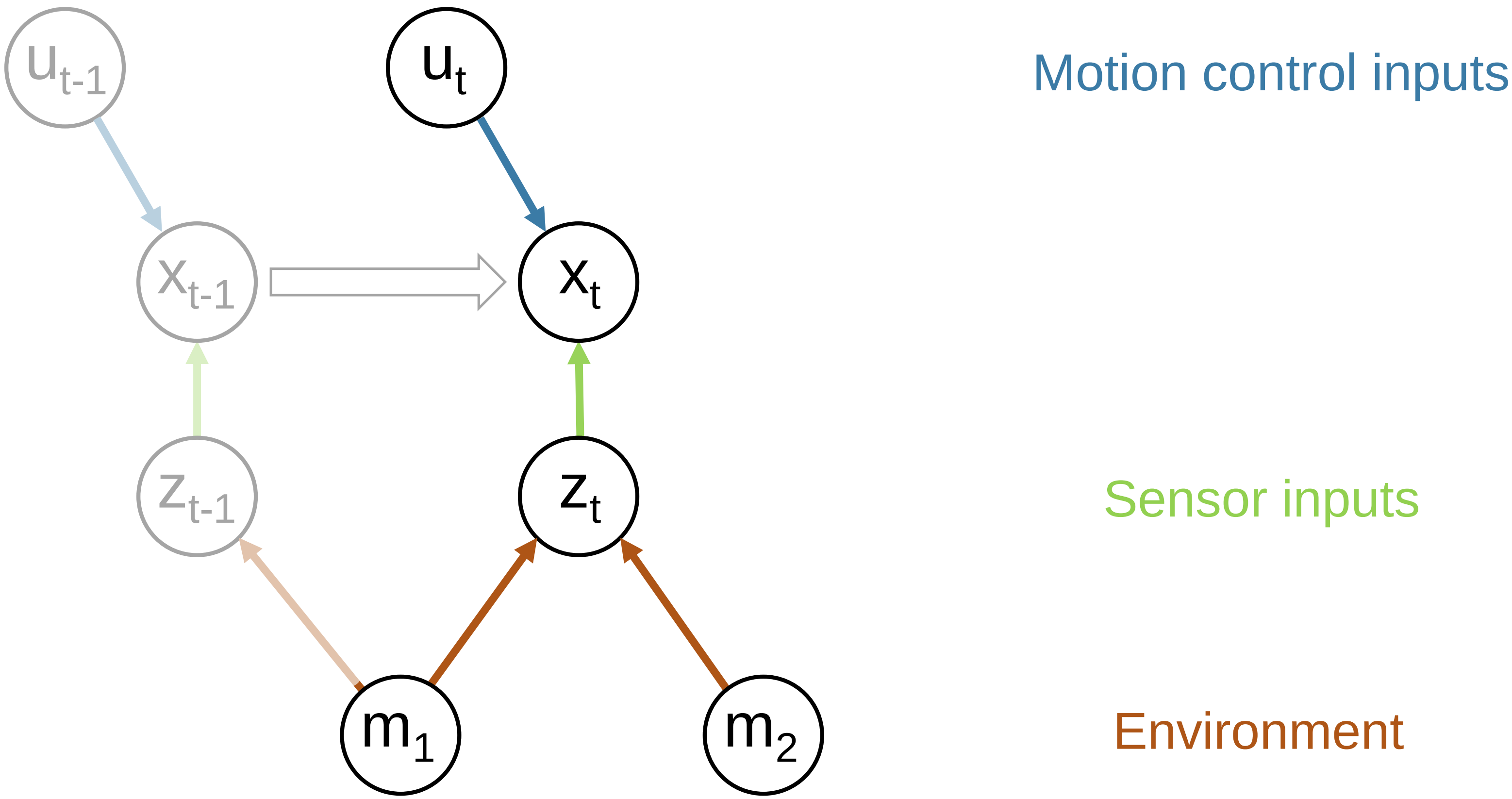
The SLAM problem



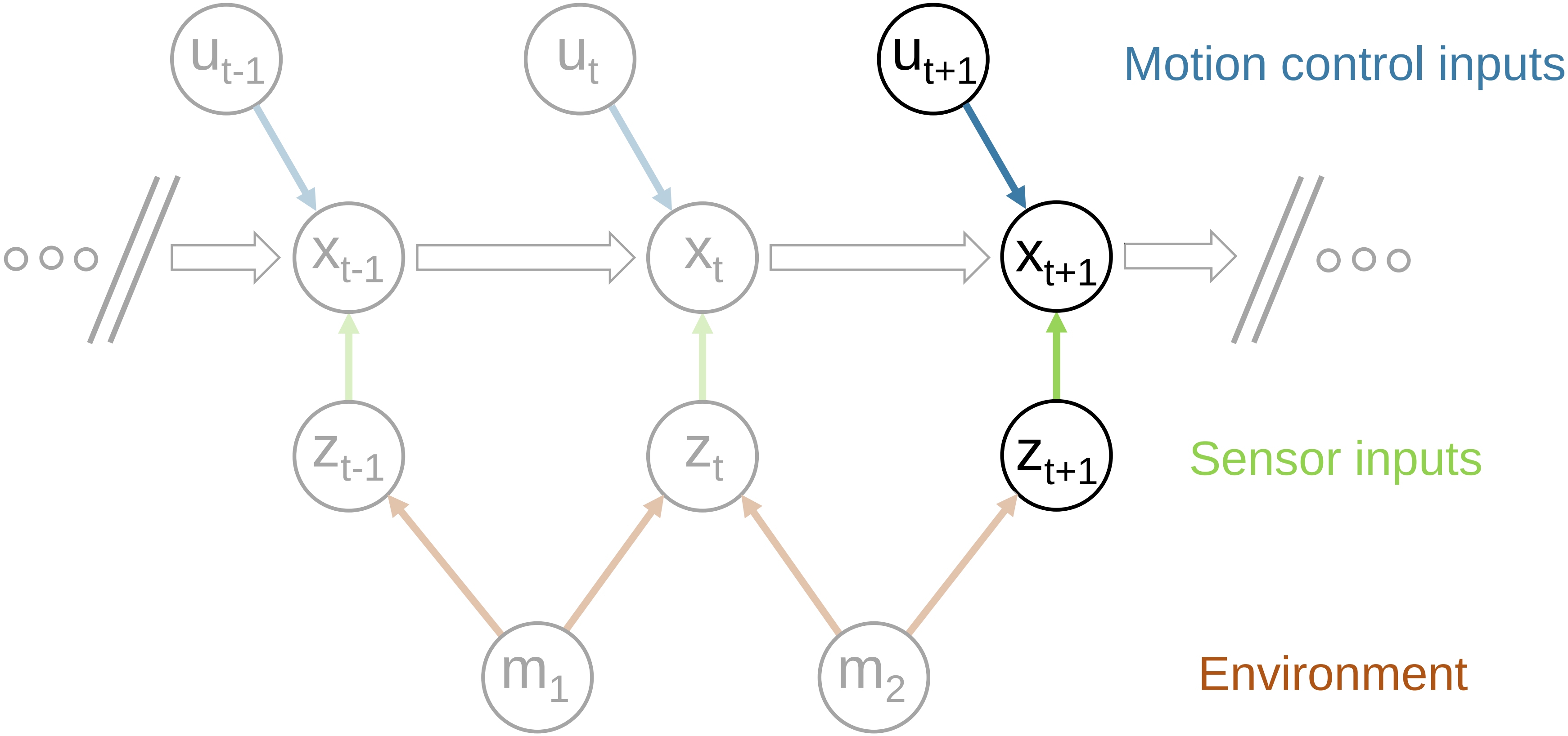
The SLAM problem



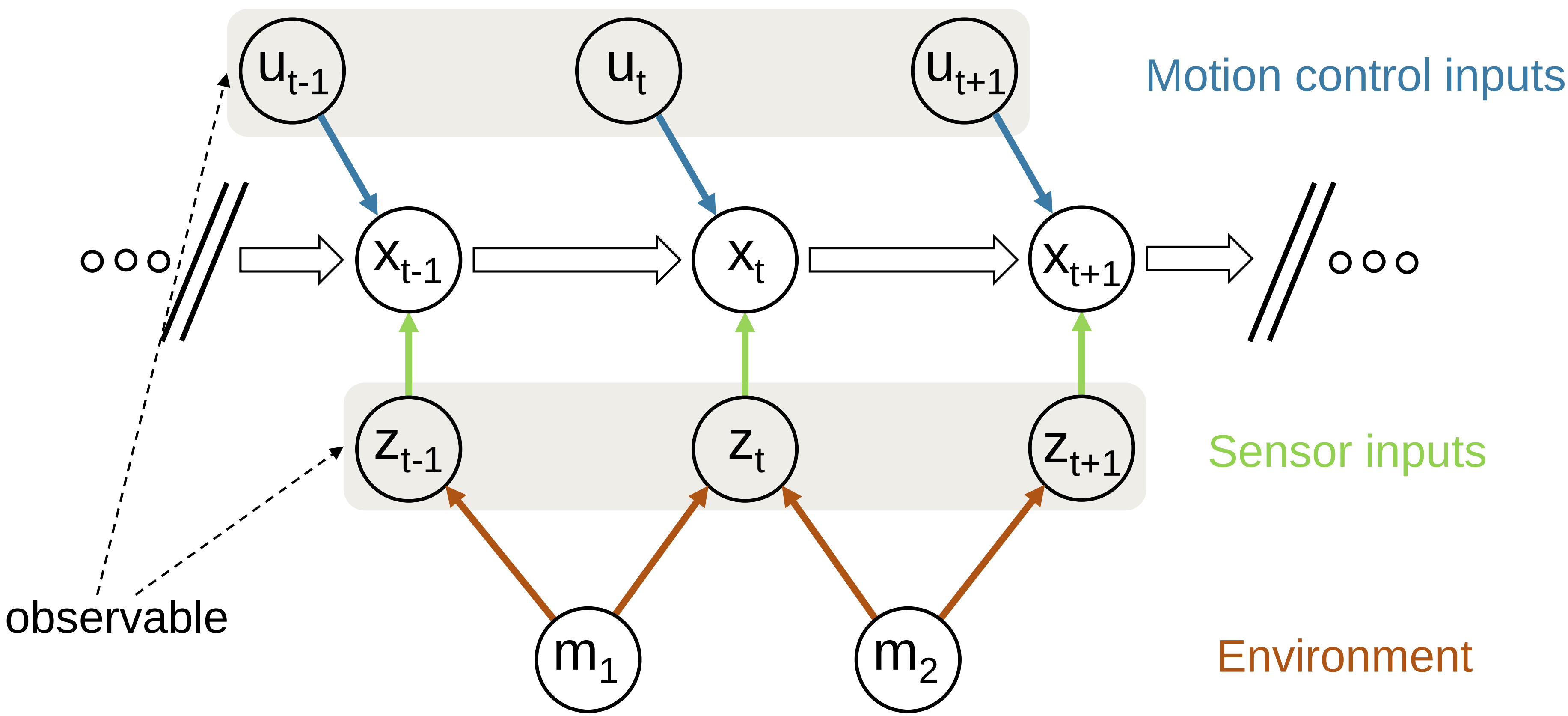
The SLAM problem



The SLAM problem



The SLAM problem



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

The SLAM problem

Two versions of the SLAM problem

Online SLAM:

Seeking the current pose of the robot

Offline SLAM:

Seeking the whole path of the robot

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. **The SLAM Algorithm**
4. SLAM Paradigms
    1. EKF SLAM
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary

The SLAM problem

SLAM problem can be seen as two separate problems:

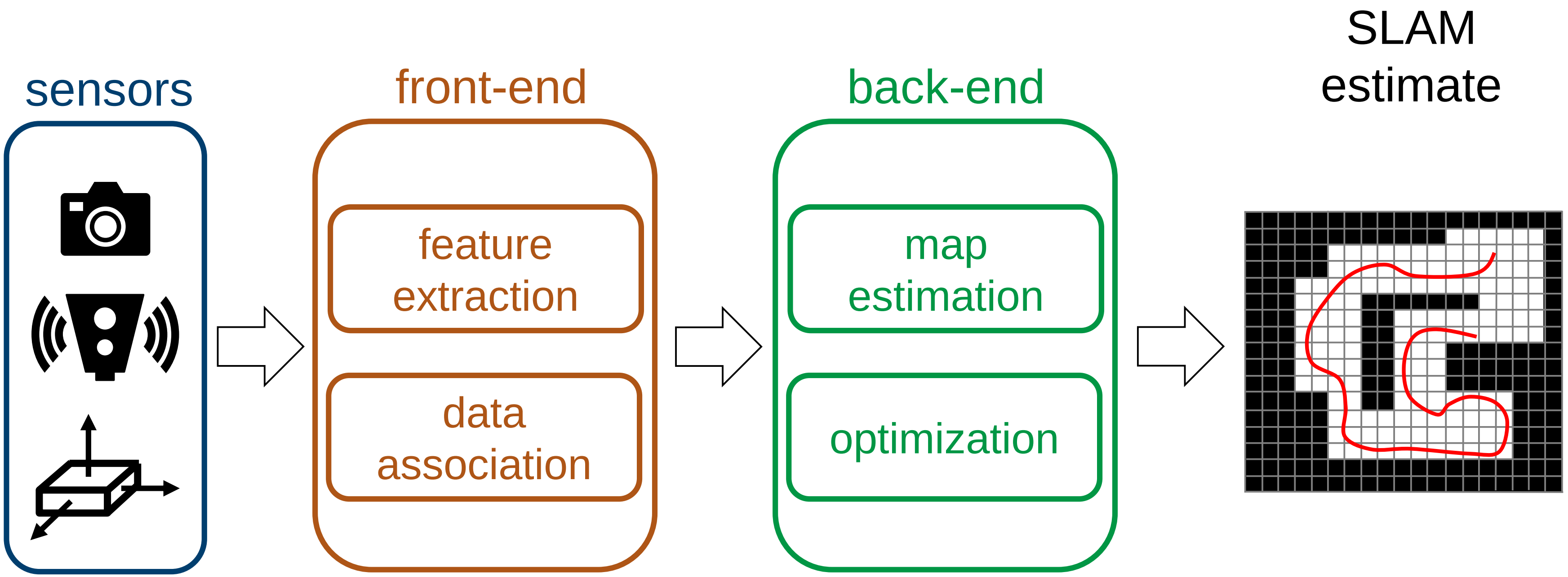
How to get the relation between ego and environment?

→ **Front-End**

How to combine all those relations to a single path and map?

→ **Back-End**

The SLAM algorithm



Exact definitions of front- and back-end depend on algorithm

The SLAM algorithm

Distinctions of SLAM

Online – Offline

Volumetric – Feature-Based

Topological – Metric

Static – Dynamic

Active – Passive

Single-Robot – Multi-Robot

Any-Time – Any-Space

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. **SLAM Paradigms**
    1. EKF SLAM
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary

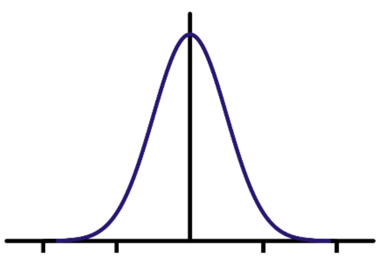
The SLAM algorithm

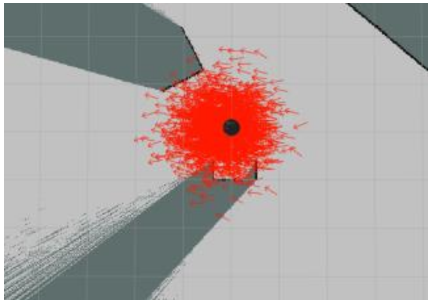
Three Main SLAM Paradigms

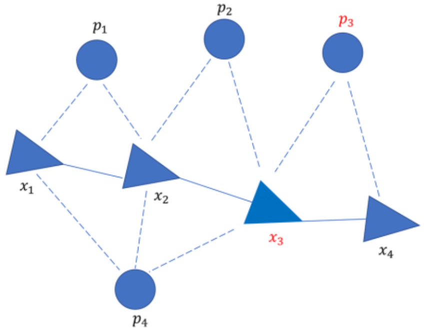
EKF

Particle

Graph-based







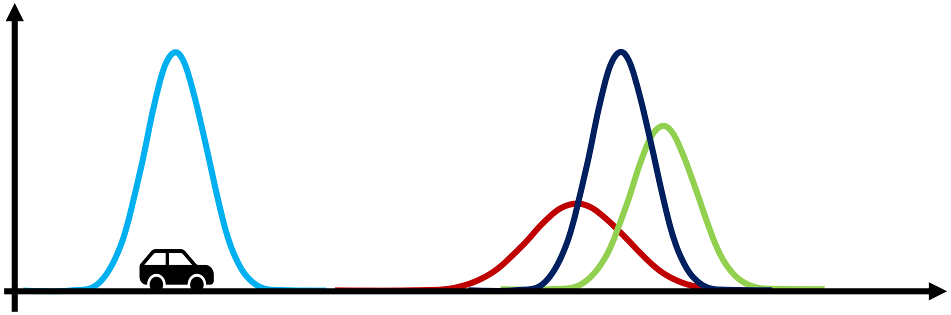
Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. **SLAM Paradigms**
    1. **EKF SLAM**
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary

EKF SLAM



Introduced in the 80s

Widespread use until today

Limited computational resources

Remember Extended Kalman Filter from previous lecture

Assumption: Known correspondence between environmental perception and ego-vehicle (features)

Robot estimate represented by a multivariate Gaussian:

$$p(x_t, m | z_t, u_t) = N(\underbrace{\mu_t}_{\text{mean}}, \underbrace{\Sigma_t}_{\text{covariance}})$$

EKF SLAM

EKF Filter Cycle:

State prediction

Measurement prediction

Measurement update

Data association

State update

vehicle state

$$\underbrace{\begin{pmatrix} x \\ m \end{pmatrix}}_{\mu} \underbrace{\begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{m_{mm}} \end{pmatrix}}_{\Sigma}$$

map

EKF SLAM

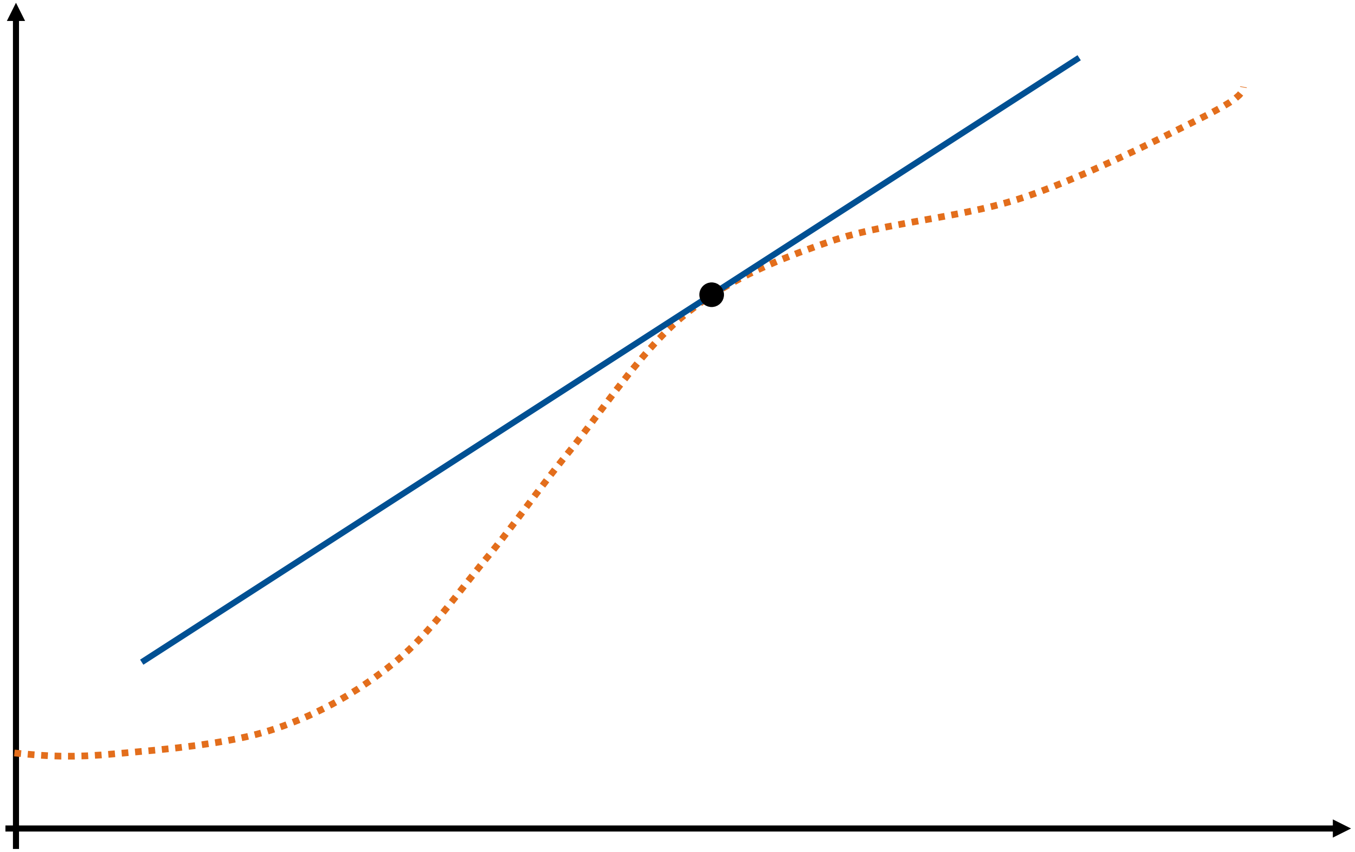
Σ is the covariance matrix.

On the main diagonal are the variances (uncertainty of the state and uncertainty of the map).

The rest of the matrix represents the covariances.

EKF SLAM

Motion function and measurement function are **Taylor-linearized**



EKF SLAM

Motion function and measurement function are **Taylor-linearized**

Uncertainties of landmark- and ego-positions are growing over time

When already used landmarks are re-perceived all uncertainties collapse → **Loop closure**

If **data association** is unknown (as usual) a probability-based data association has to be applied

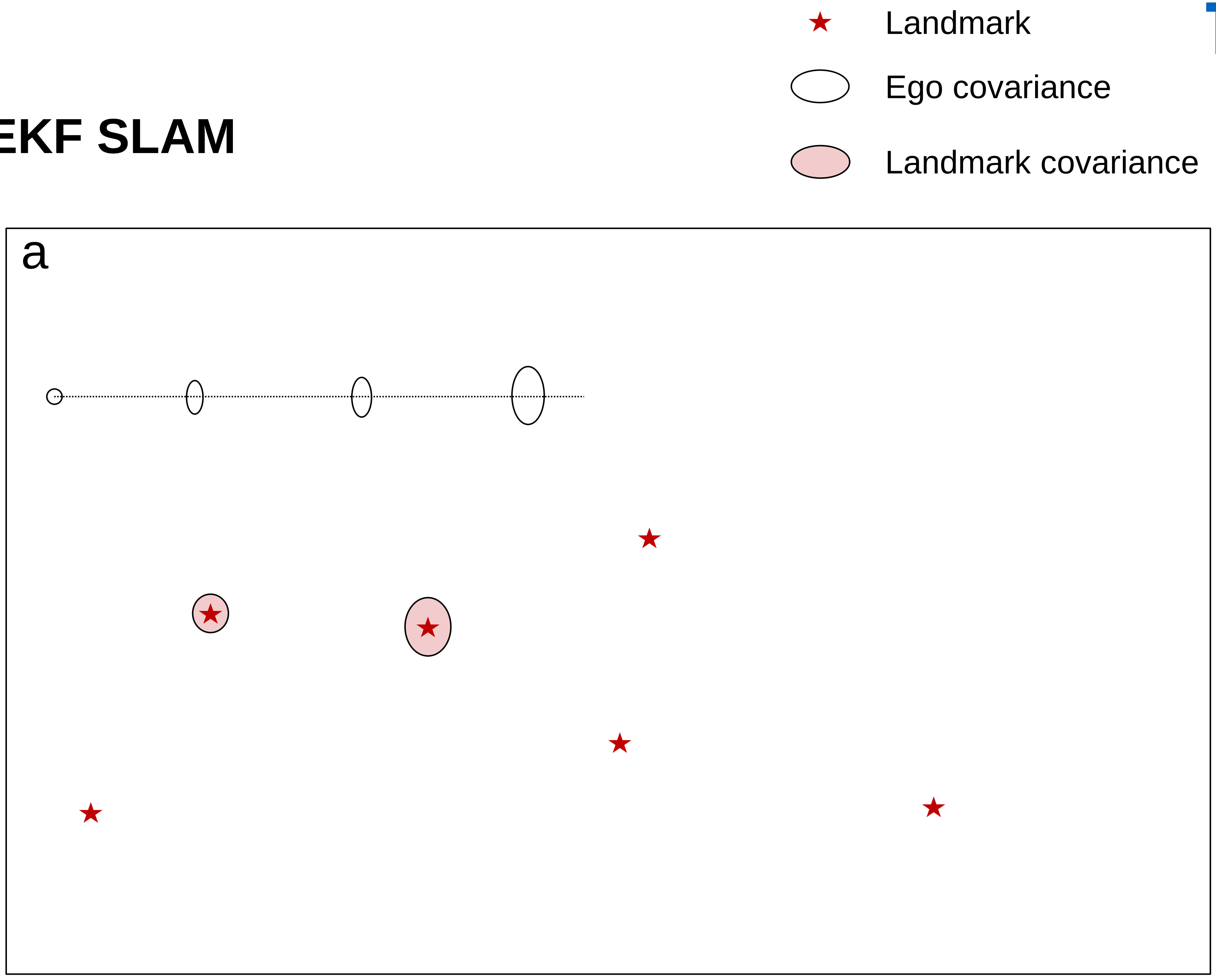
EKF SLAM

Example

Assumptions:

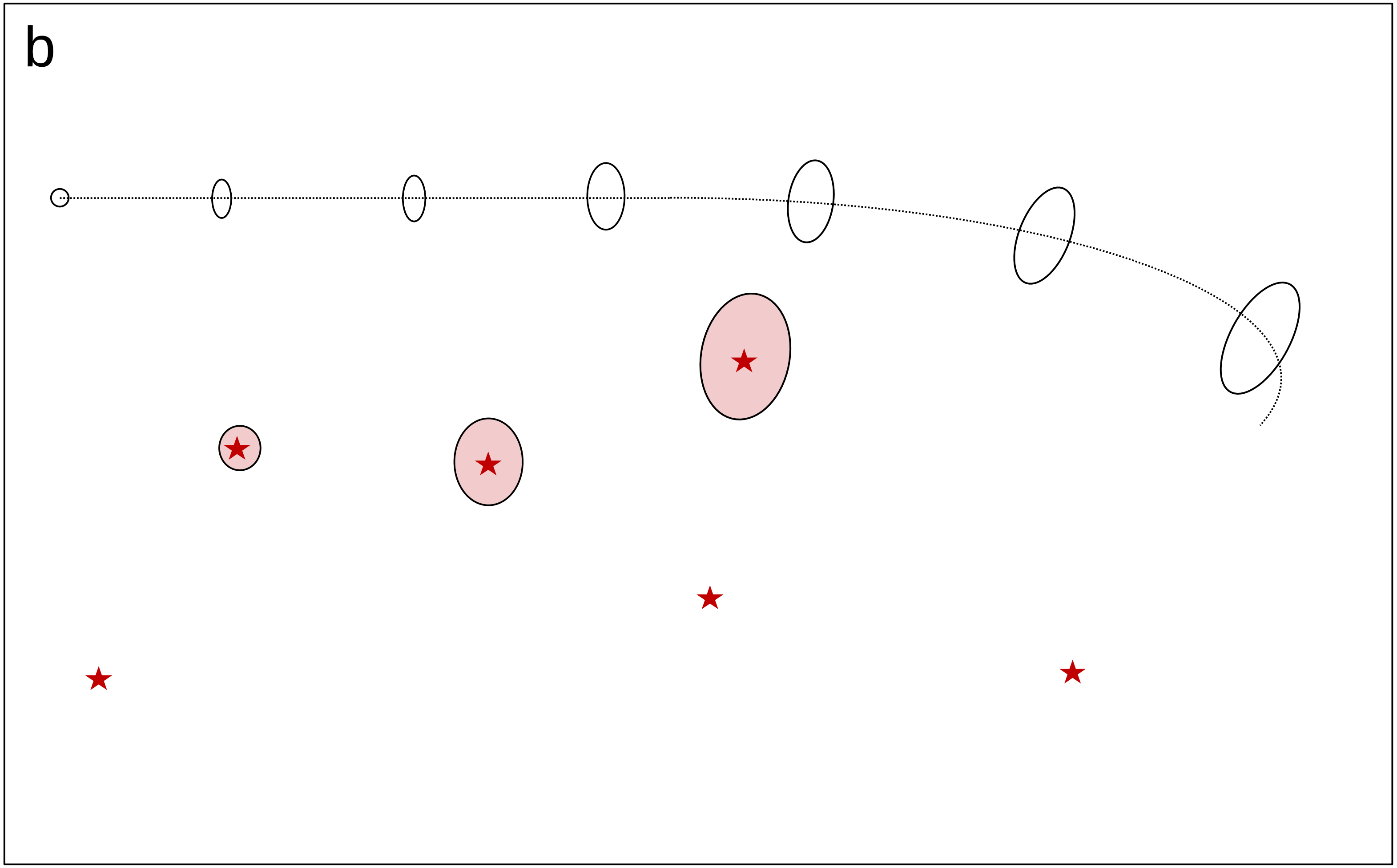
- Moving robot through virtual environment
- Known data association
- All landmarks are fully perceivable (distance + angle)

EKF SLAM

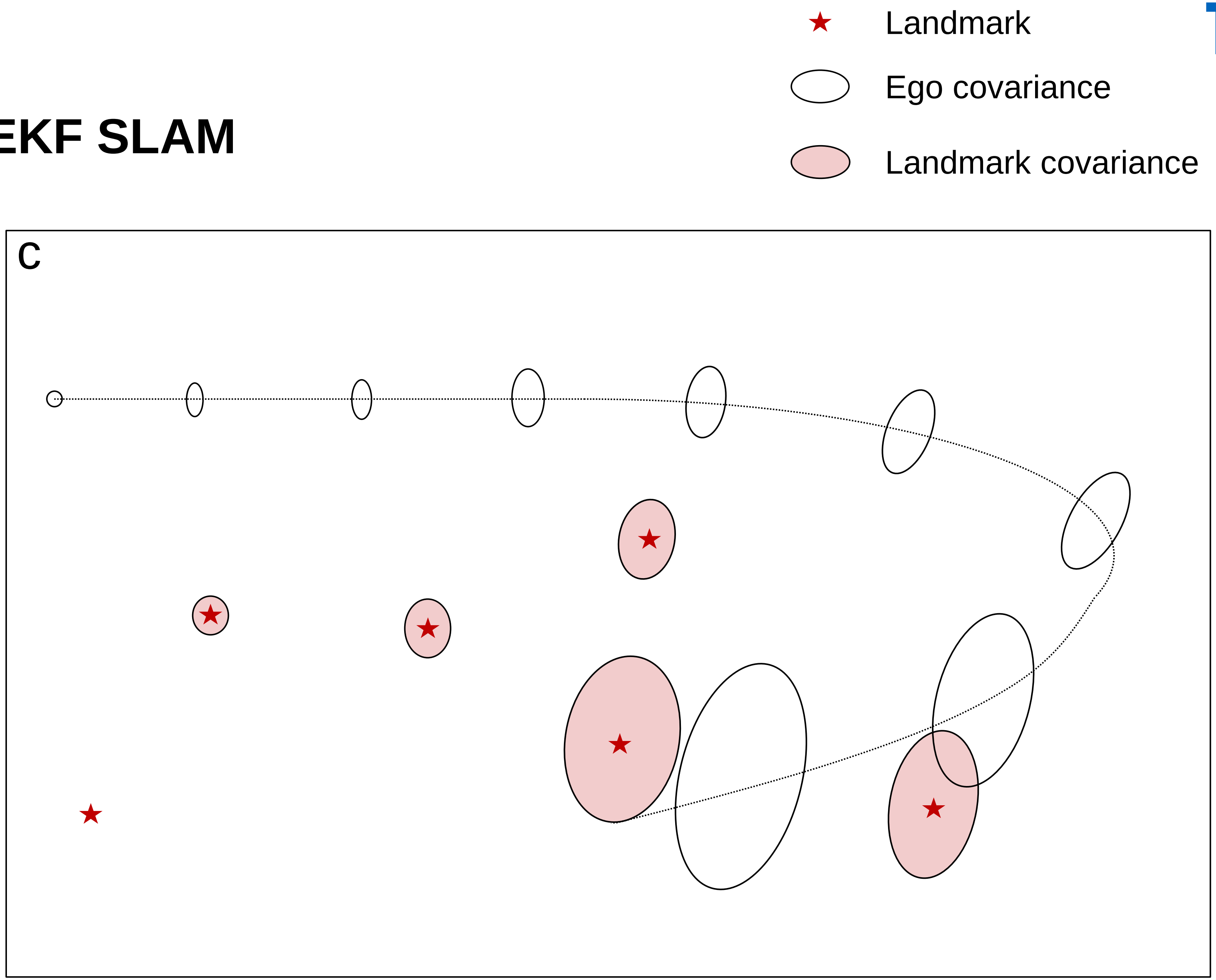


EKF SLAM

★ Landmark
Ego covariance
Landmark covariance



EKF SLAM



EKF SLAM

If the map, represented by the sum of all landmarks, is known and we want to use the EKF for localization, the uncertainties won't grow over time as long as we still perceive landmarks.

This is because we always have detected landmarks as global reference. When we have to estimate both, path and map, the uncertainties grow as we are missing this reference.

EKF SLAM

Problem

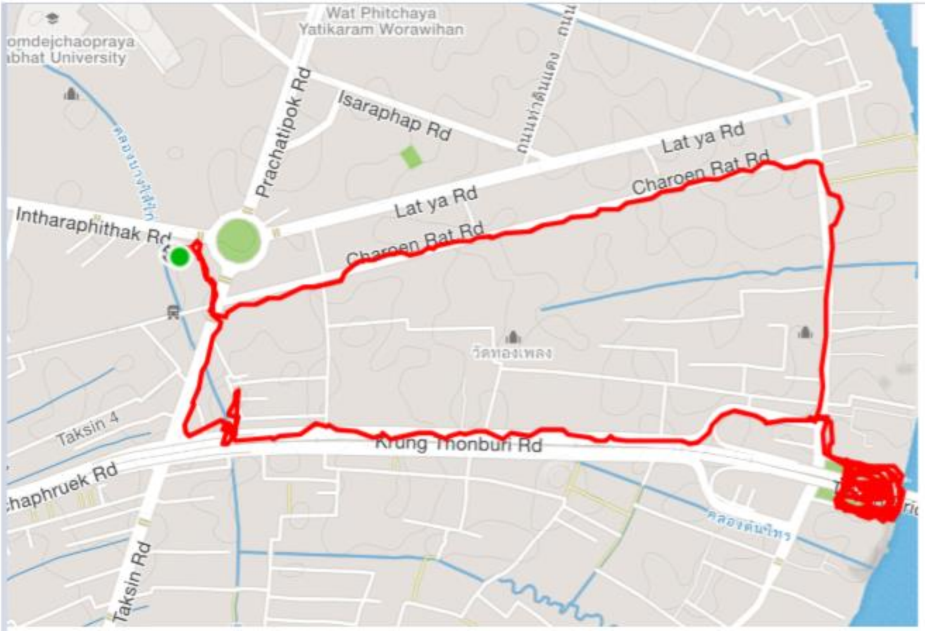
Uncertainties keep increasing

Pose estimate drifts away

Solution

“Global” reference is needed

- Maps
- GNSS
- **Loop closure**



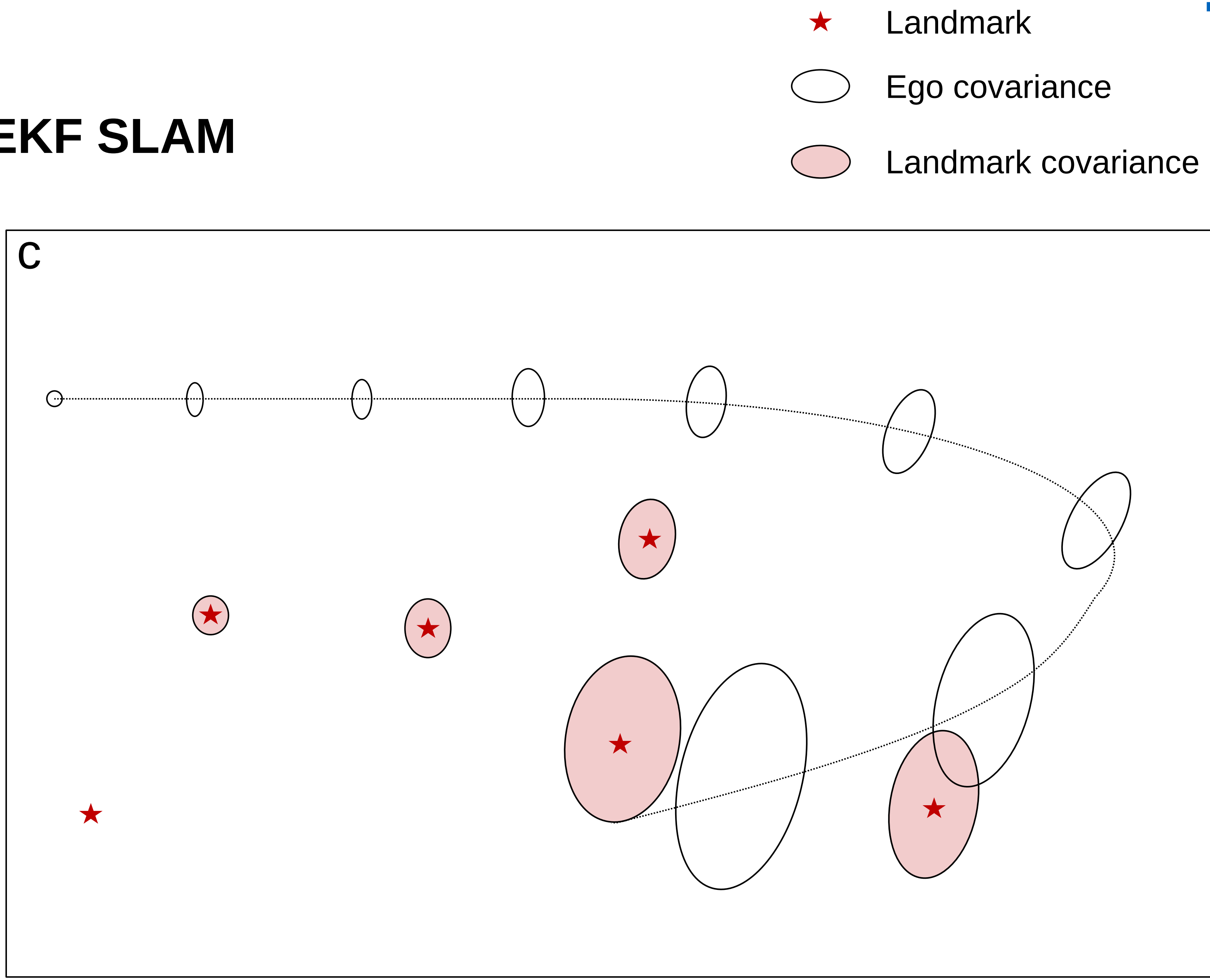
https://towardsdatascience.com/how-tracking-apps-analyse-your-gps-data-a-hands-on-tutorial-in-python-756d4db6715d

EKF SLAM

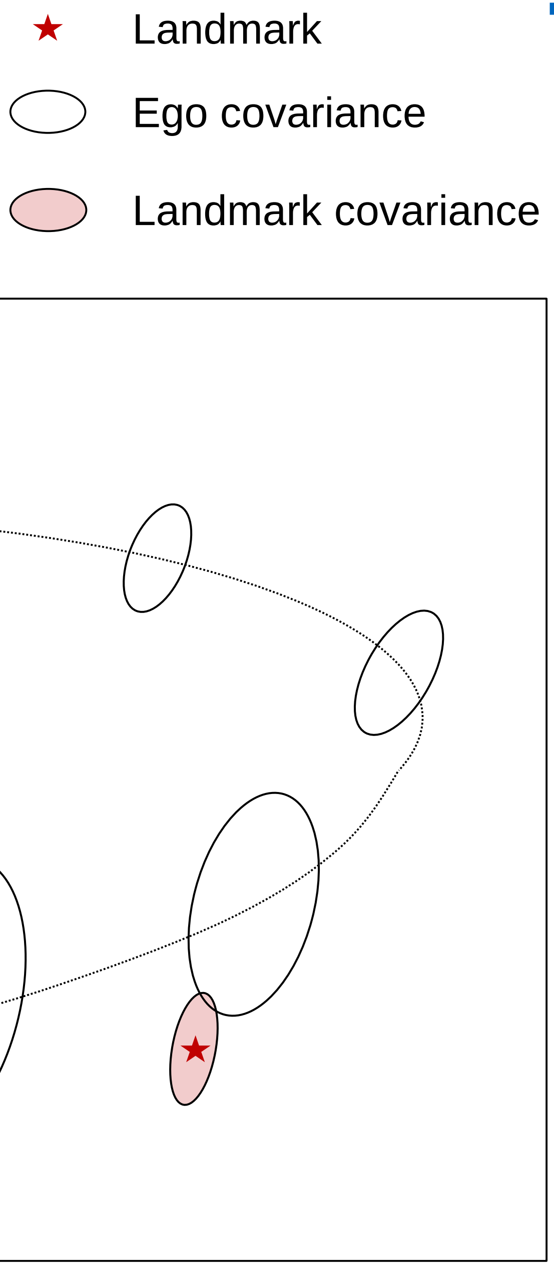
If we want to drive autonomously and have a high level navigation, we don’t care too much about the global drift and uncertainty.

More important is the local localization within the perceivable environment. Relative uncertainties to the direct surroundings don’t grow as we are steadily detecting the environment with the same sensors and thus uncertainties.

EKF SLAM

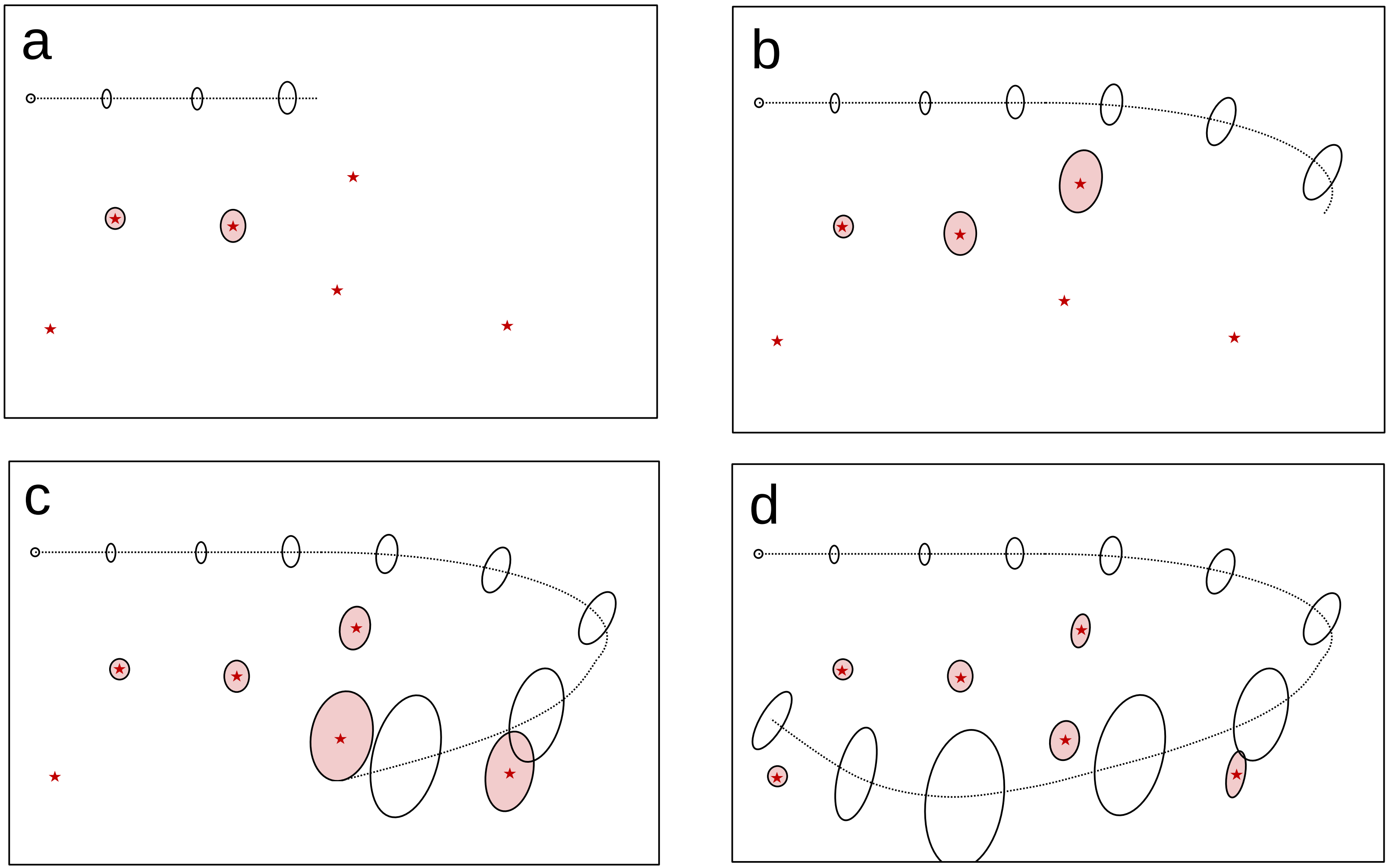


EKF SLAM



EKF SLAM

● Landmark
Ego covariance
Landmark covariance



EKF SLAM

Pros and Cons

+ Easy applicable
+ Well known method
+ Computationally cheap for small maps

- Only applicable for small non-linearities
- Highly dependent on parametrization
- Computationally unworkable for big maps (Covariance scales quadratic)

Other methods of applying Kalman Filters for nonlinear problems:

- Unscented (UKF)
- Extended Information (EIF)

Mapping & Localization II
Prof. Dr. Markus Lienkamp

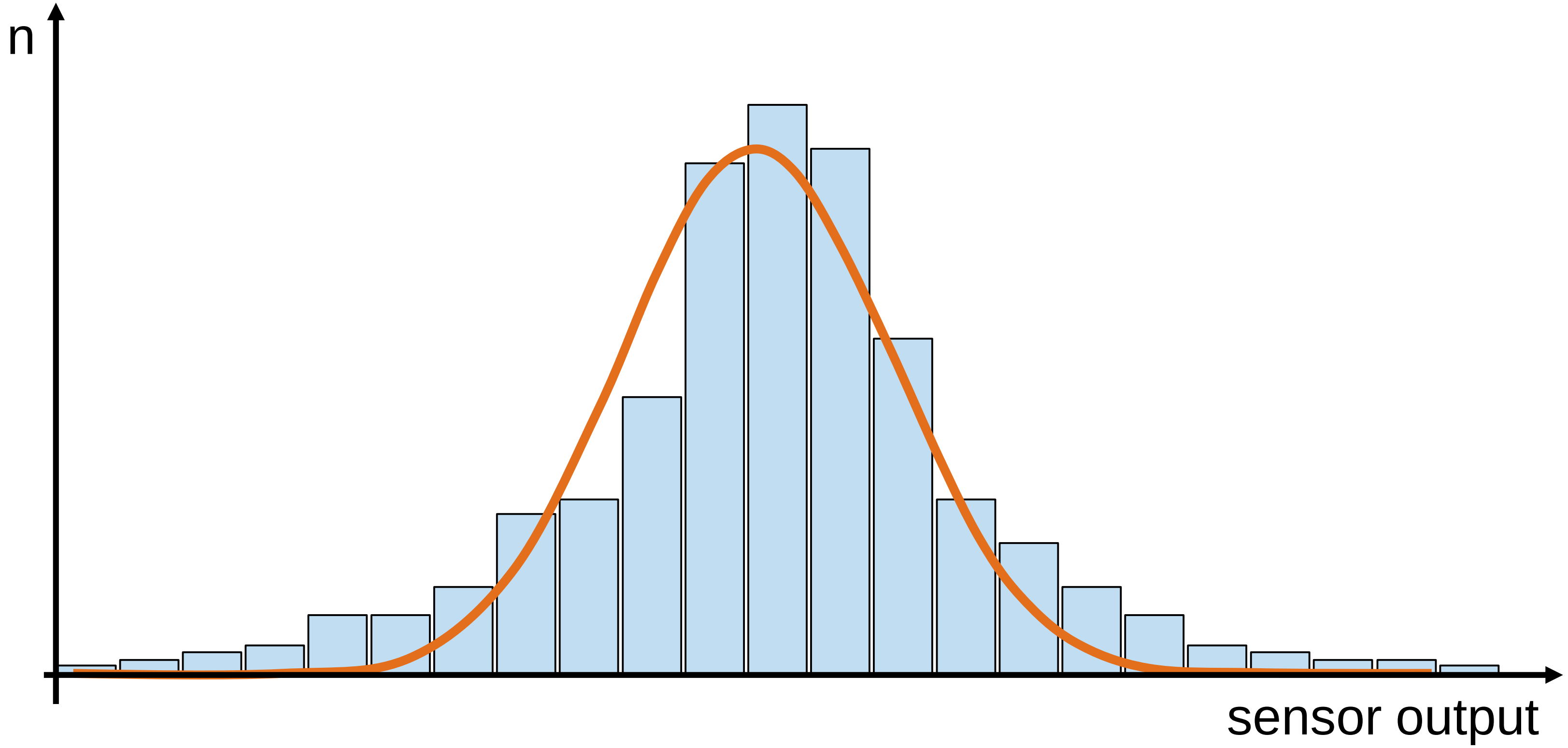
Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. **SLAM Paradigms**
    1. EKF SLAM
    2. **Particle Filter SLAM**
    3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. Summary


Particle Filter SLAM

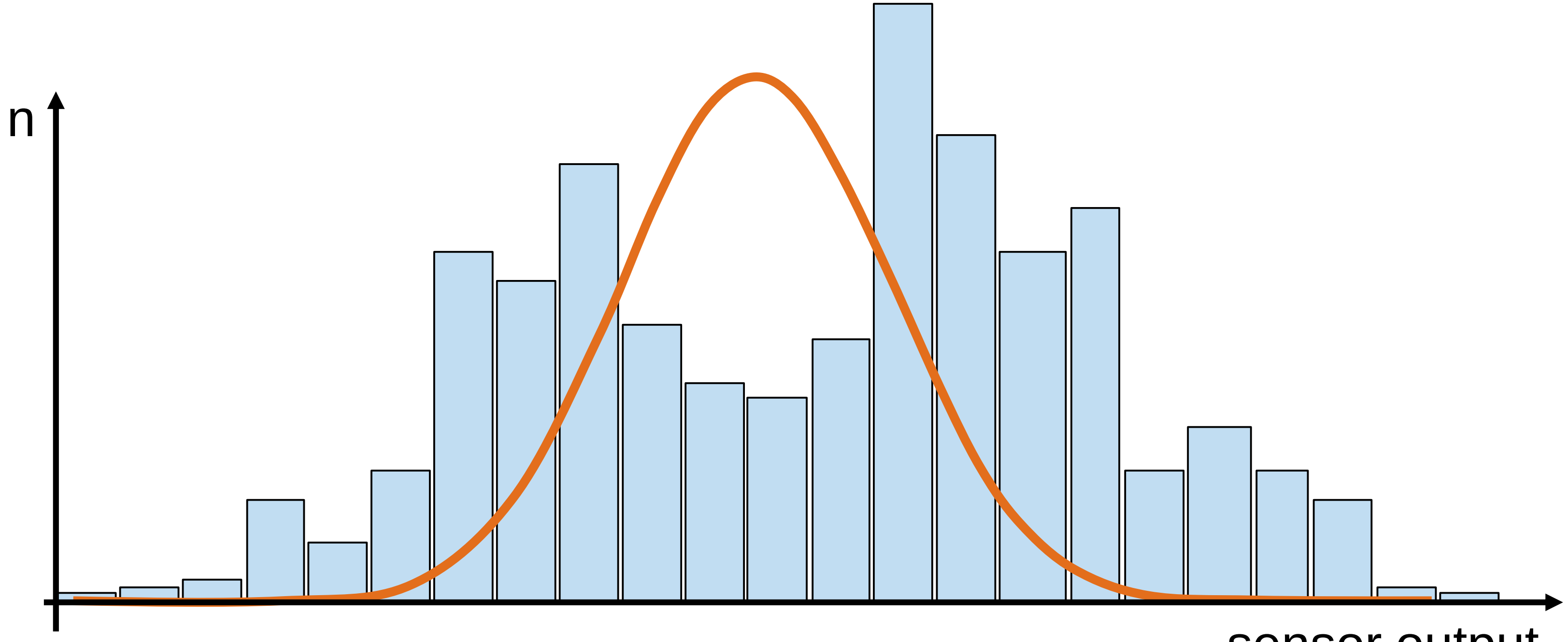
Motivation



Particle Filter SLAM

Motivation

What if sensors can't be assumed Gaussian?



Particle Filter SLAM

Each posterior represented by a set of particles

Each particle can be seen as a single guess of the robot state

Many particles needed, scale exponentially with dimensions

→ only applicable for **low dimensions**

Making it applicable: Rao-Blackwellization

→ **fastSLAM**

For any point in time, fastSLAM has K particles with path $x_t^{[k]}$ and N Gaussians with mean $\mu_t^{[n]}$ and variance $\Sigma_t^{[n]}$ (one for each landmark)

Particle Filter SLAM

Each posterior represented by a set of particles

Each particle can be seen as a single guess of the robot state

Many particles needed, scale exponentially with dimensions

→ only applicable for **low dimensions**

For any point in time, Particle filter SLAM has K particles with path $x_t^{[k]}$ and N Gaussians with mean $\mu_t^{[n]}$ and variance $\Sigma_t^{[n]}$ (one for each landmark)

Particle Filter SLAM

In theory, we could also model the uncertainty distribution of the landmarks as particles.

In practice this won’t be possible as we would have to model K state particles and for each of this particle N landmark particles.

Particle Filter SLAM

Basic Algorithm

1. Sample particle set based on proposal distribution

2. Compute the importance weight of single particles

3. Resample particles based on calculated weights

Particle Filter SLAM

Short insight into Rao-Blackwellization

Path is assumed known by predictions

Factorization to exploit dependencies between variables:

$$p(a, b) = p(b|a)p(a)$$

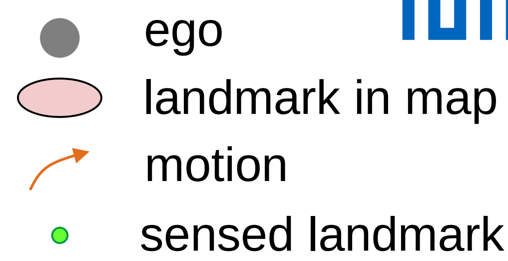
poses map observations movements

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:\mathrm{t}}) = p(x_{0:t} \mid z_{1:t}, u_{1:\mathrm{t}})\; p(l_{1:M} \mid x_{0:t}, z_{1:\mathrm{t}})$$

path posterior map posterior

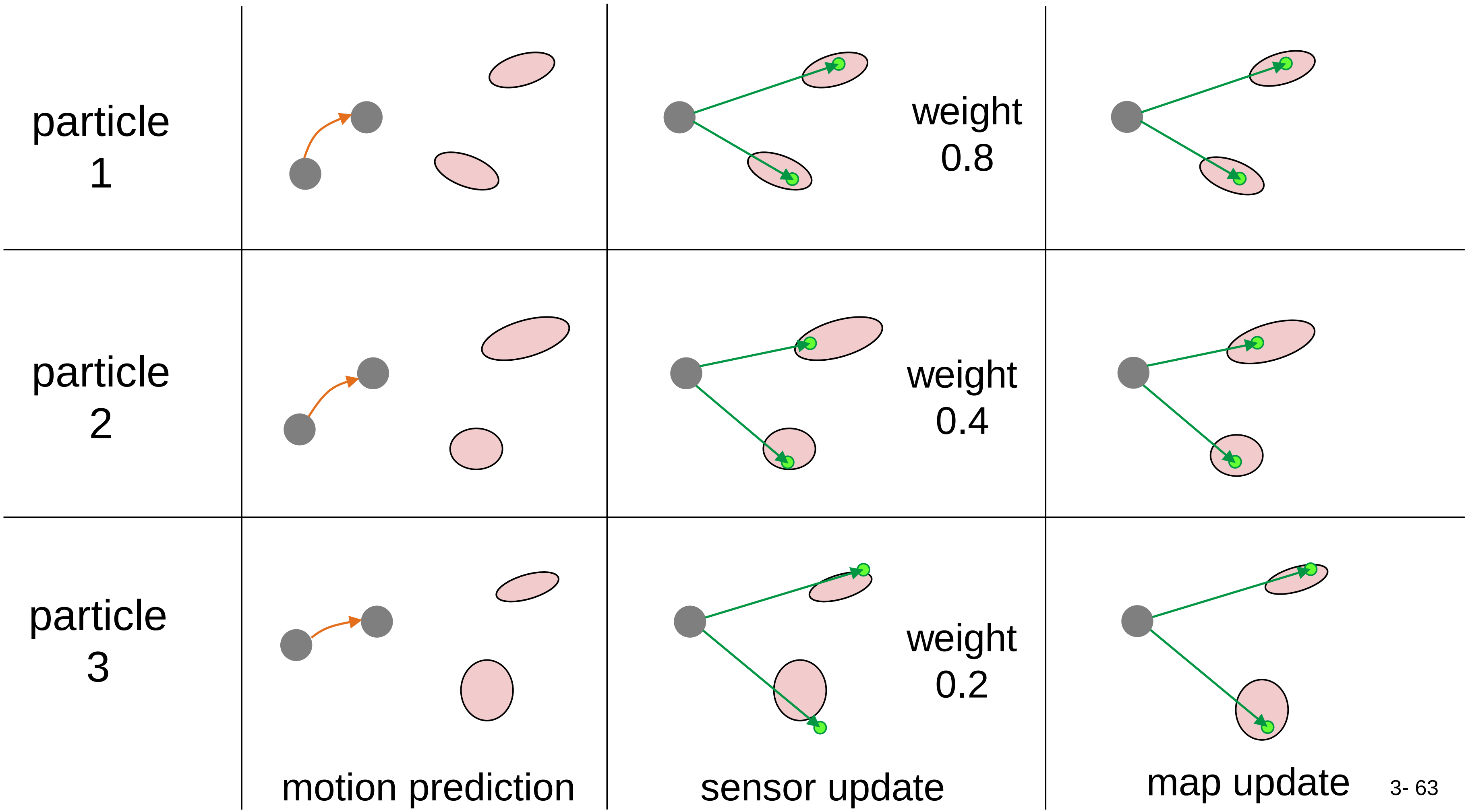
If you are interested in the details of the Rao-Blackwellization the Uni Freiburg Courses give you a good starting point:

http://ais.informatik.uni-freiburg.de/teaching/ws12/mapping/pdf/slam20-frontends-4.pdf



Particle Filter SLAM

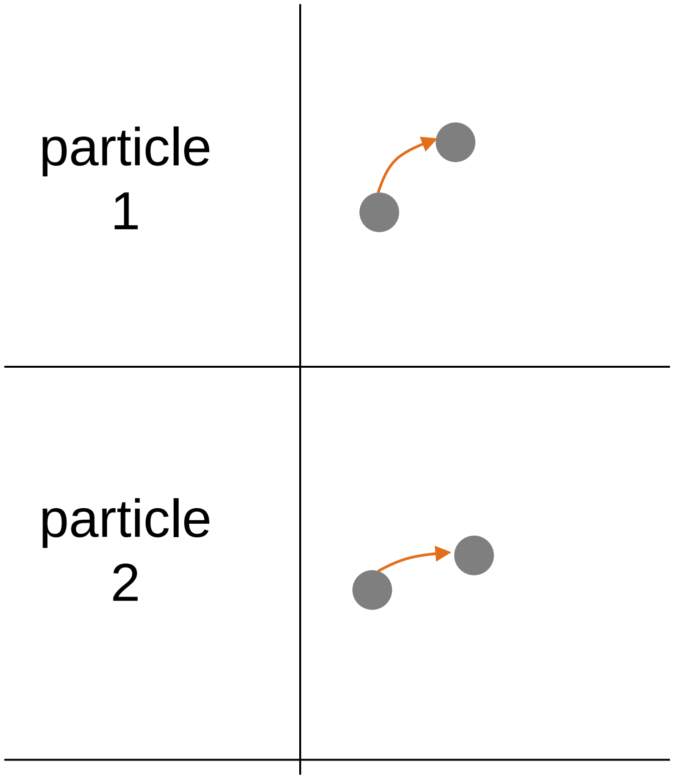
fastSLAM algorithm



Particle Filter SLAM

ego
landmark in map
motion
sensed landmark

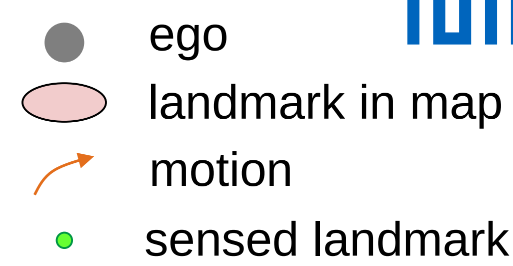
fastSLAM algorithm



Motion prediction

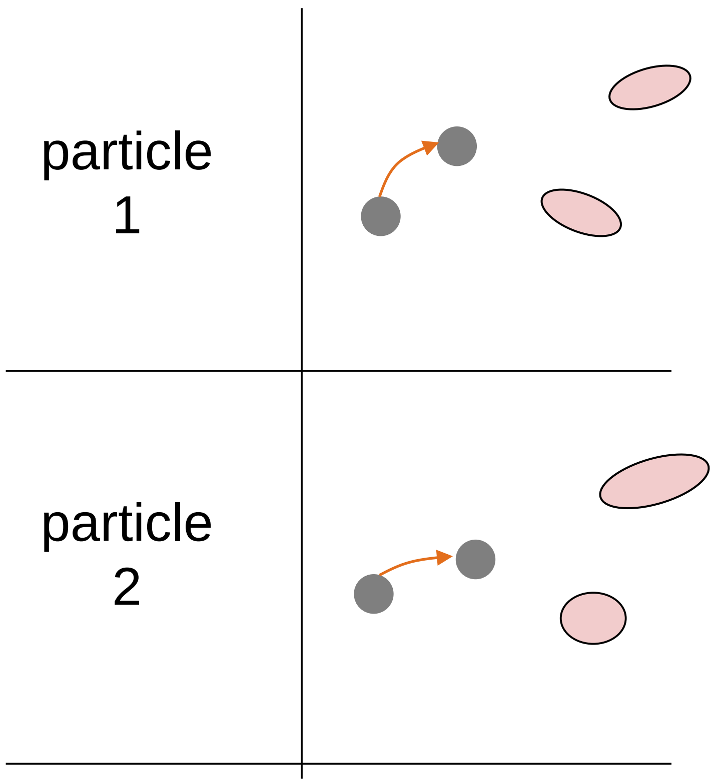
Ego motion predicted through controls and motion model

Different particles have different predictions



Particle Filter SLAM

fastSLAM algorithm

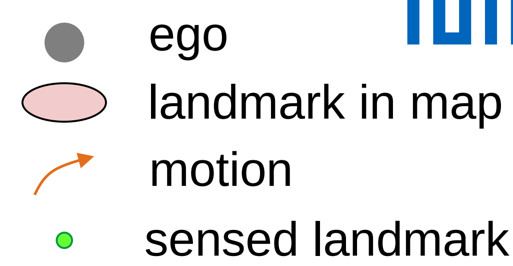


motion prediction

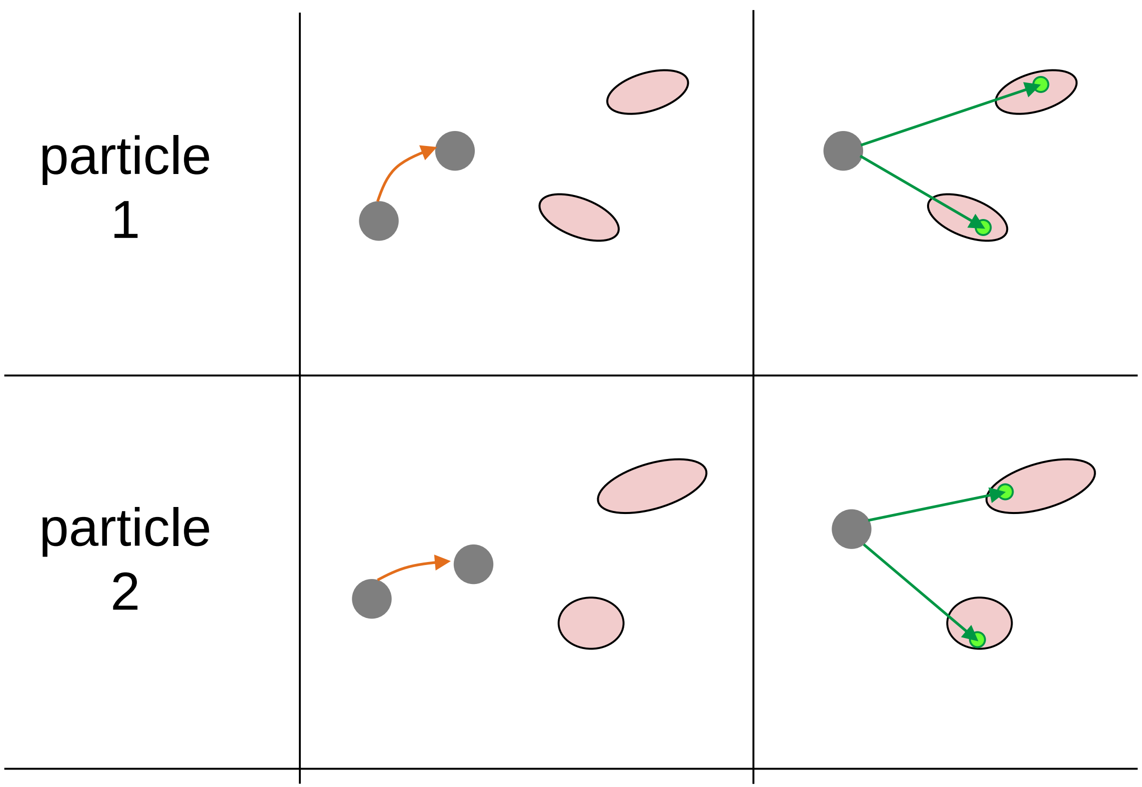
Landmark prediction

Based on updated position, measurements are predicted

Particle Filter SLAM



fastSLAM algorithm



motion prediction

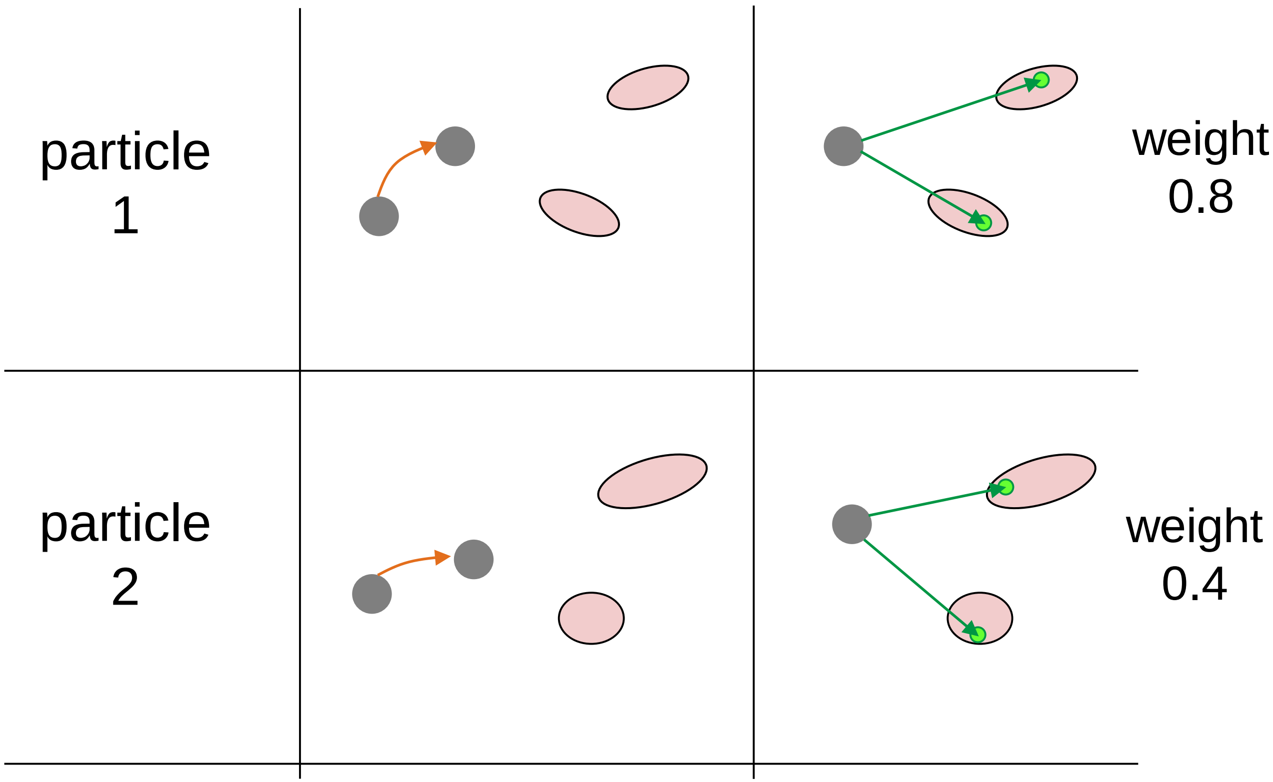
Sensor update

New sensor measurements

Particle Filter SLAM

ego
landmark in map
motion
sensed landmark

fastSLAM algorithm



motion prediction

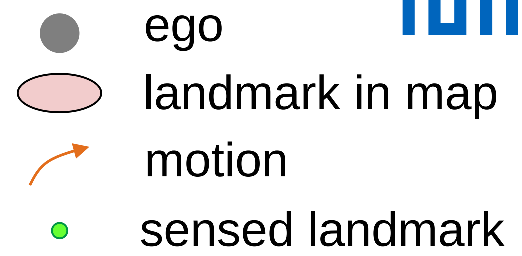
Sensor update

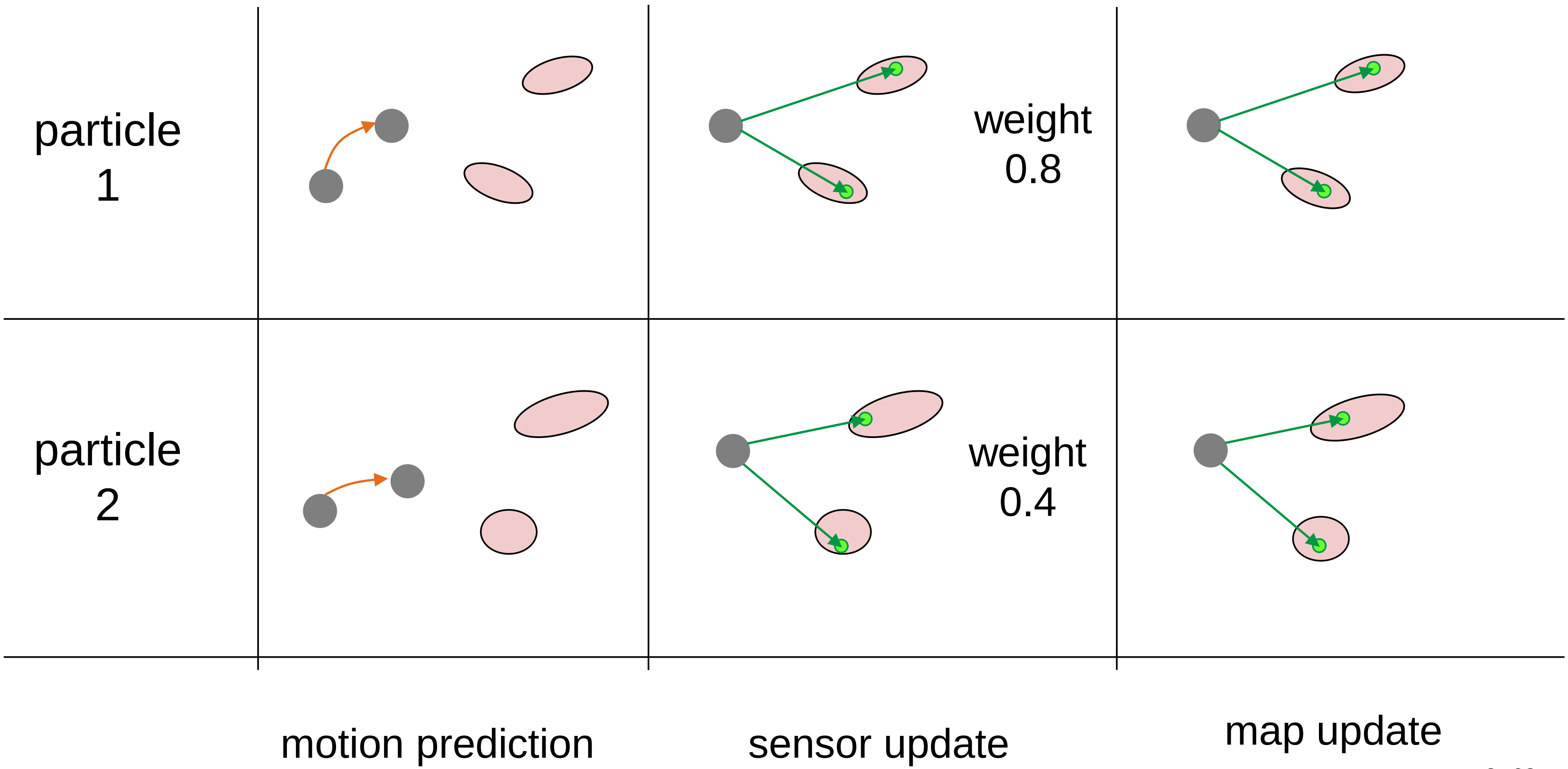
Comparison of predicted and actual measurements

→ Weighting particles

Particle Filter SLAM

fastSLAM algorithm





Particle Filter SLAM

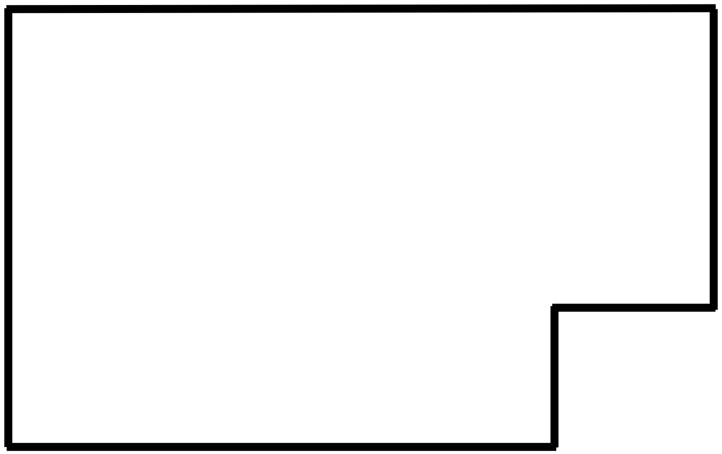
Data Association

Which observation senses which landmark in the map?

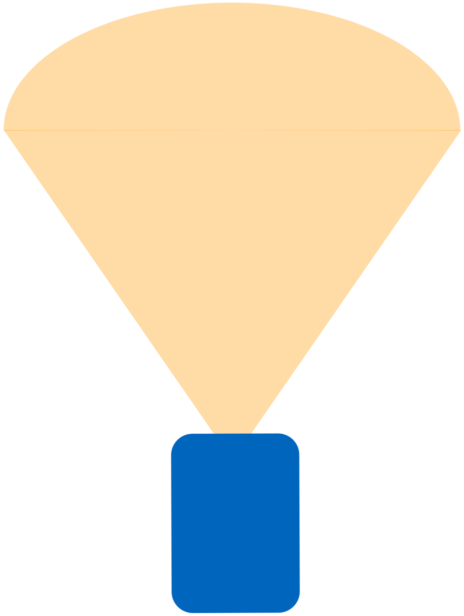
Per-particle approach:

- Association for each particle and landmark
- Based on probability
- Same observations can be differently associated for different particles

Particle Filter SLAM

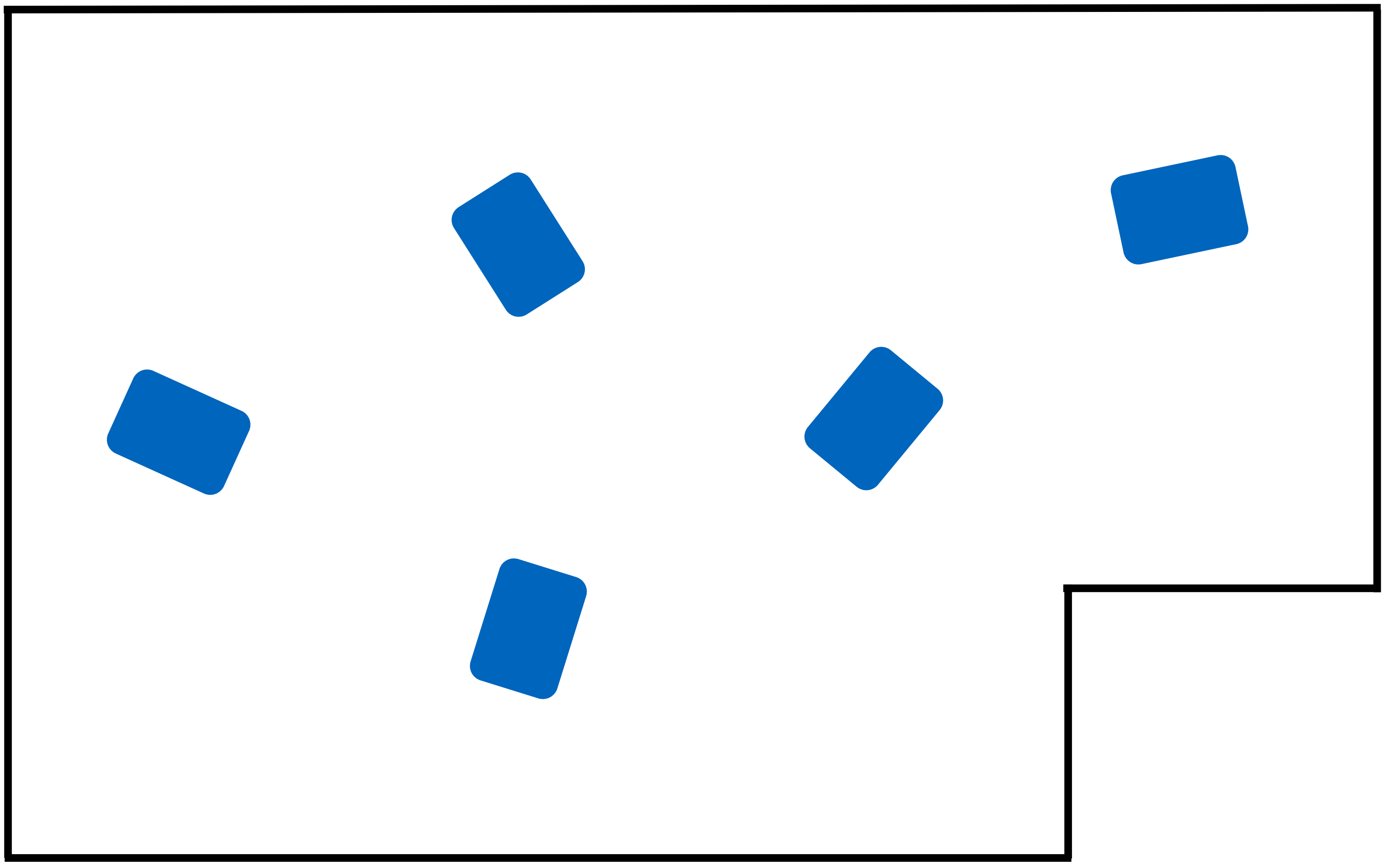


Environment



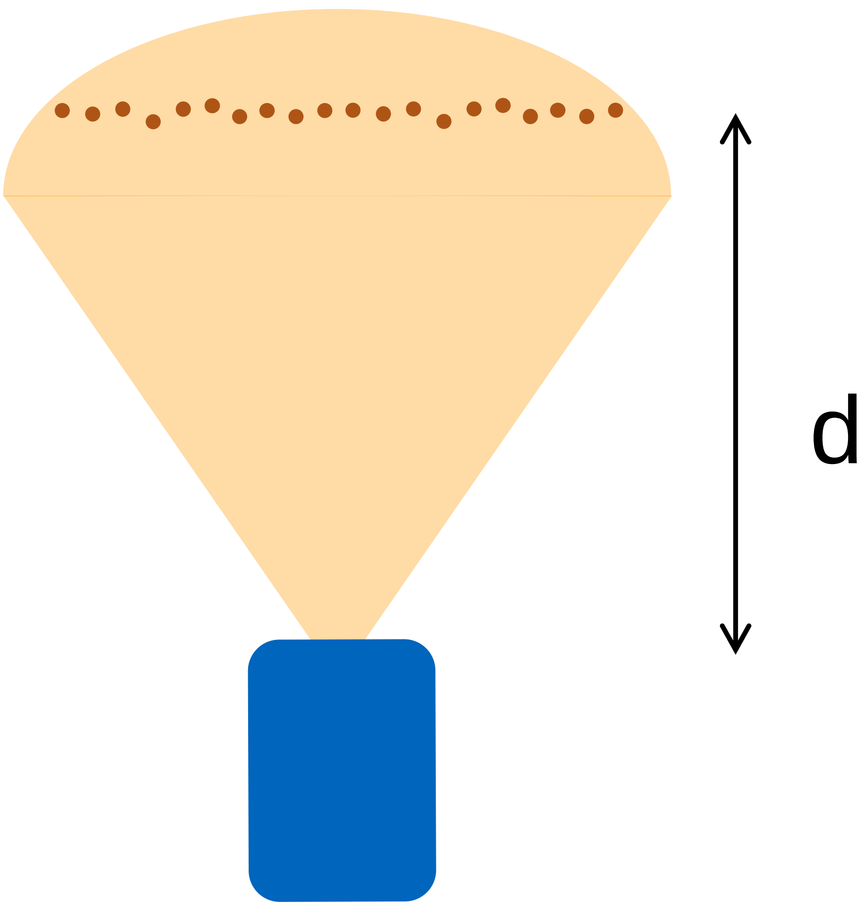
Robot

Particle Filter SLAM

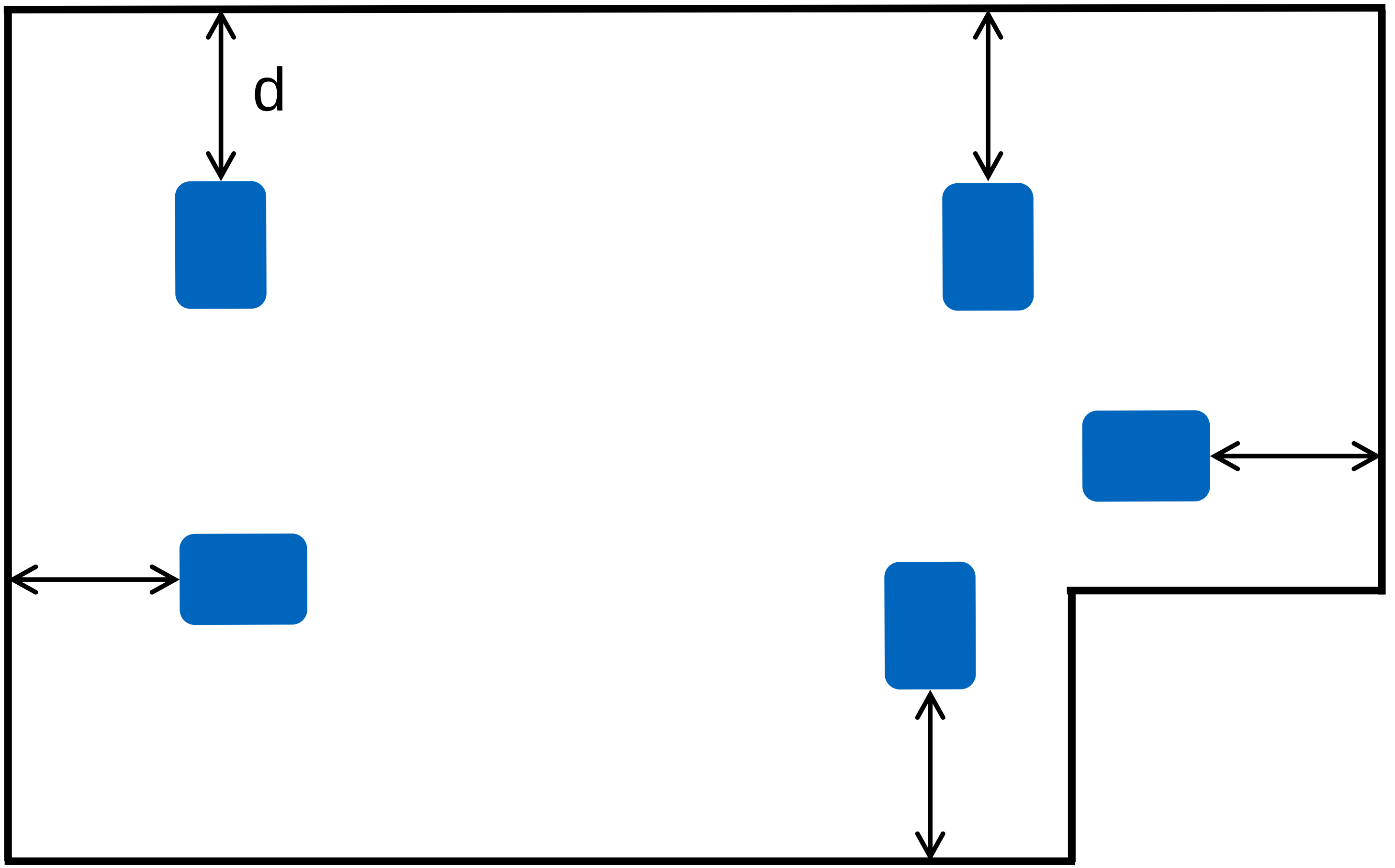


Equally distributed

Particle Filter SLAM

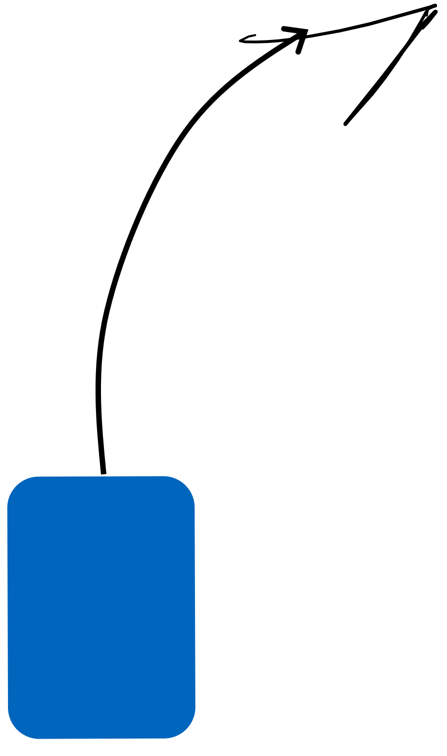


Particle Filter SLAM

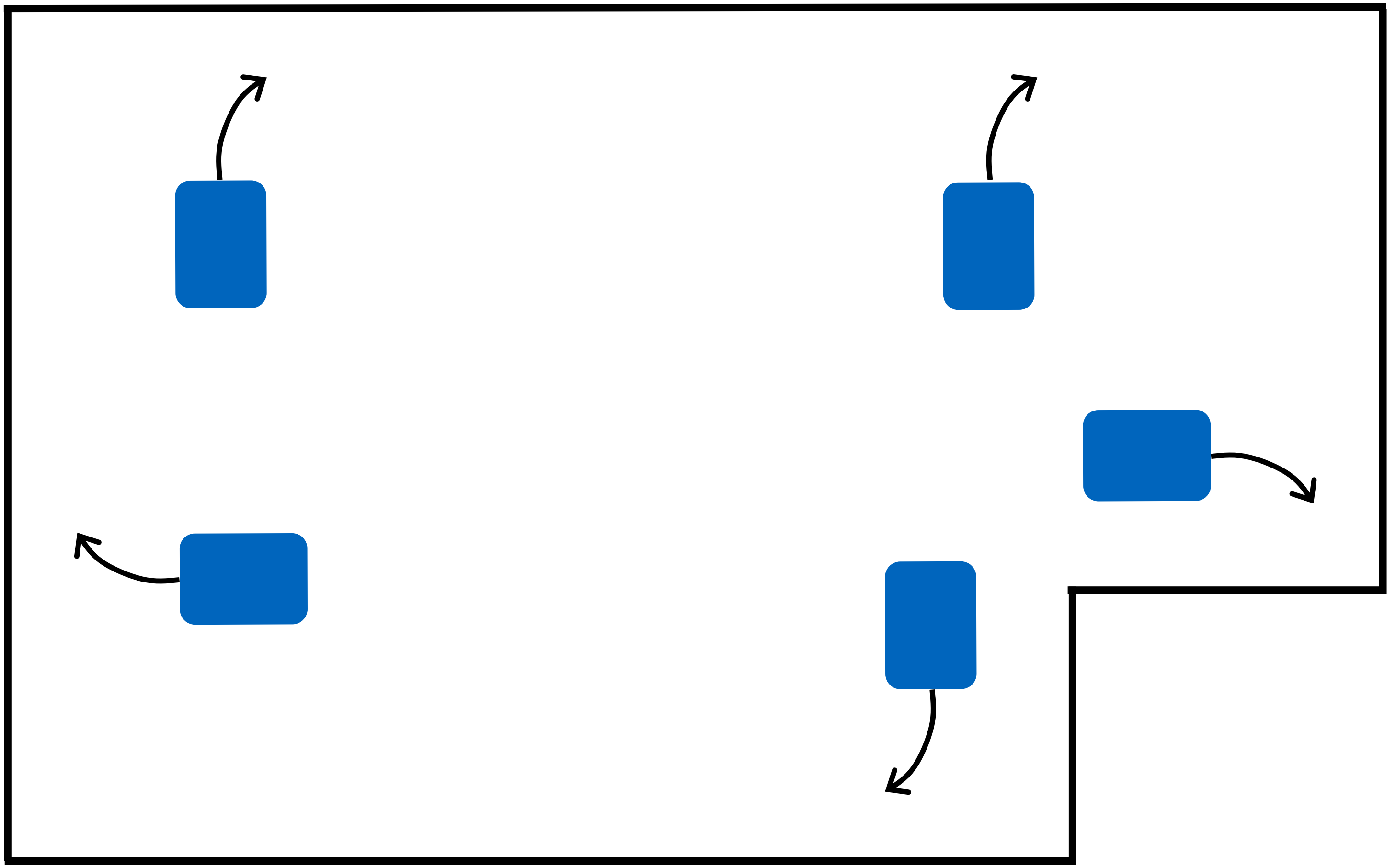


Particle Filter SLAM

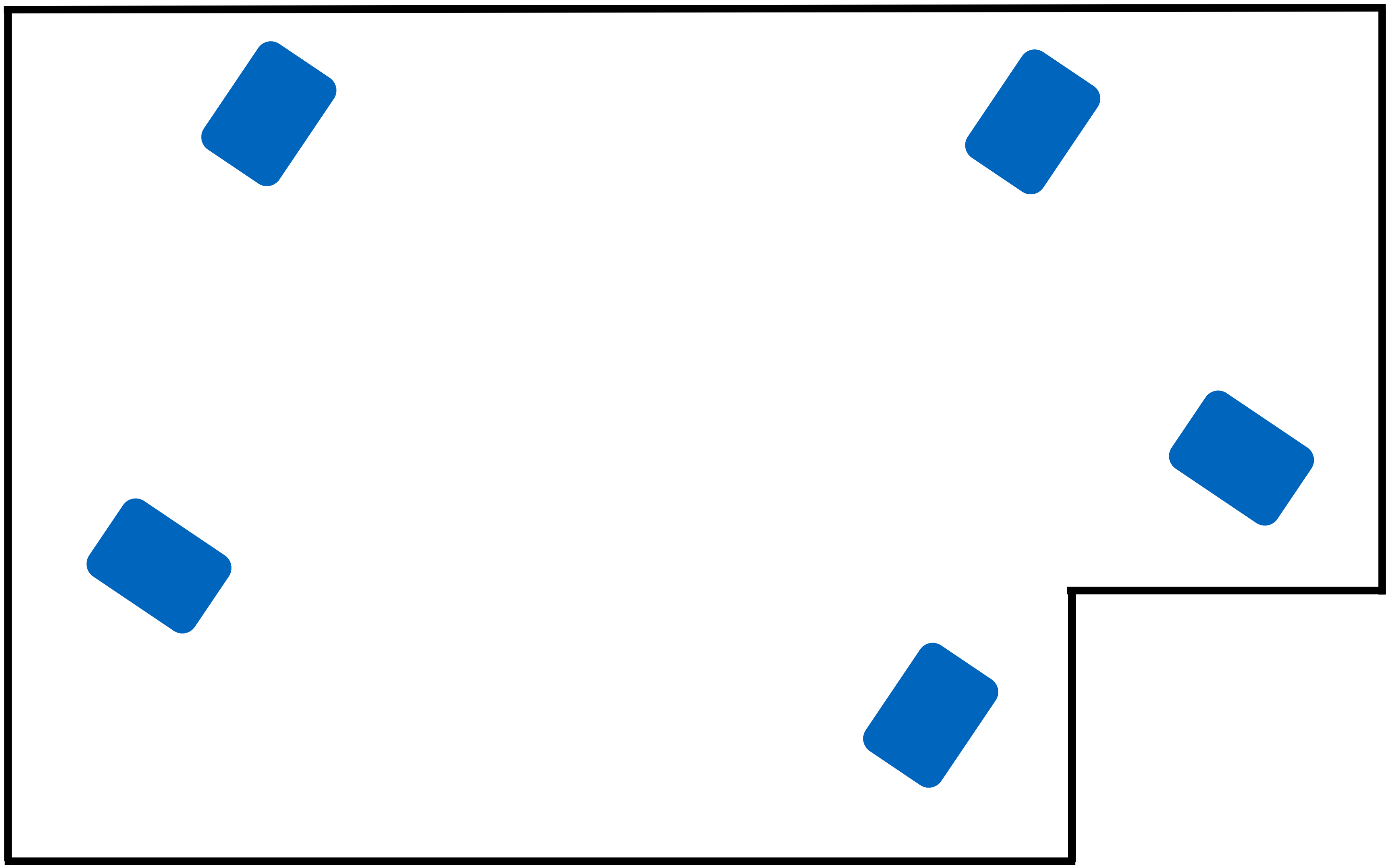
Odometry



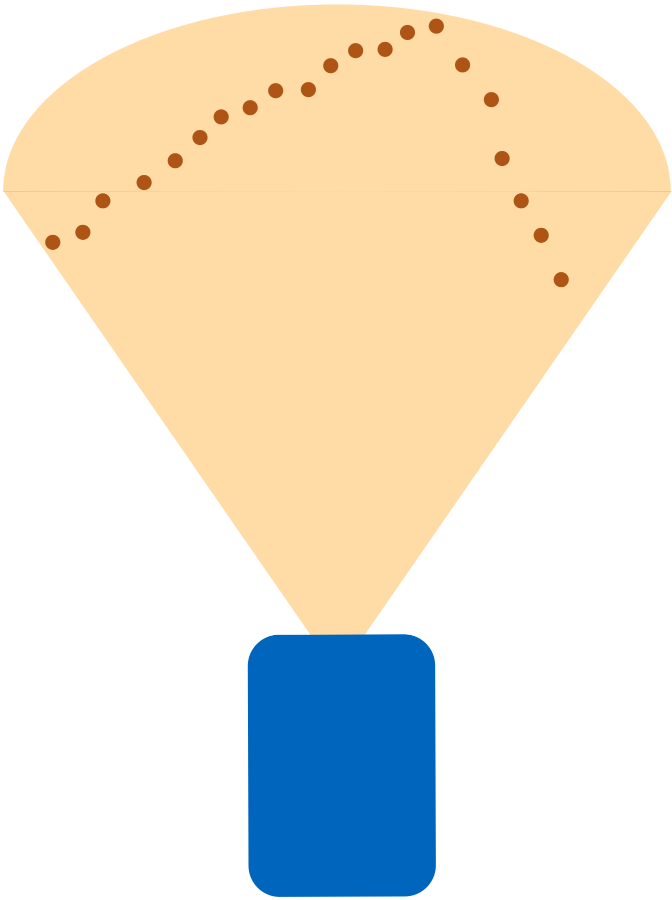
Particle Filter SLAM



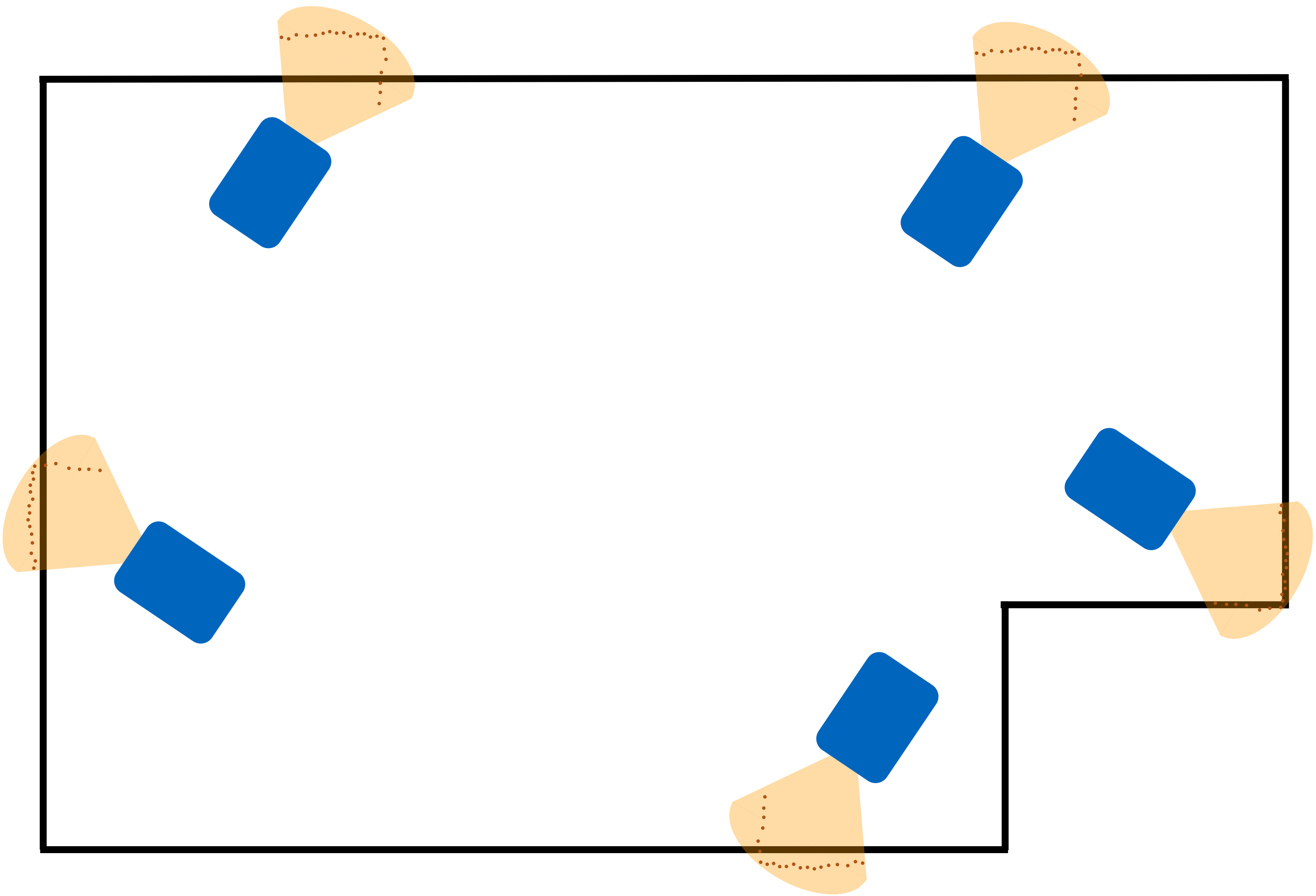
Particle Filter SLAM



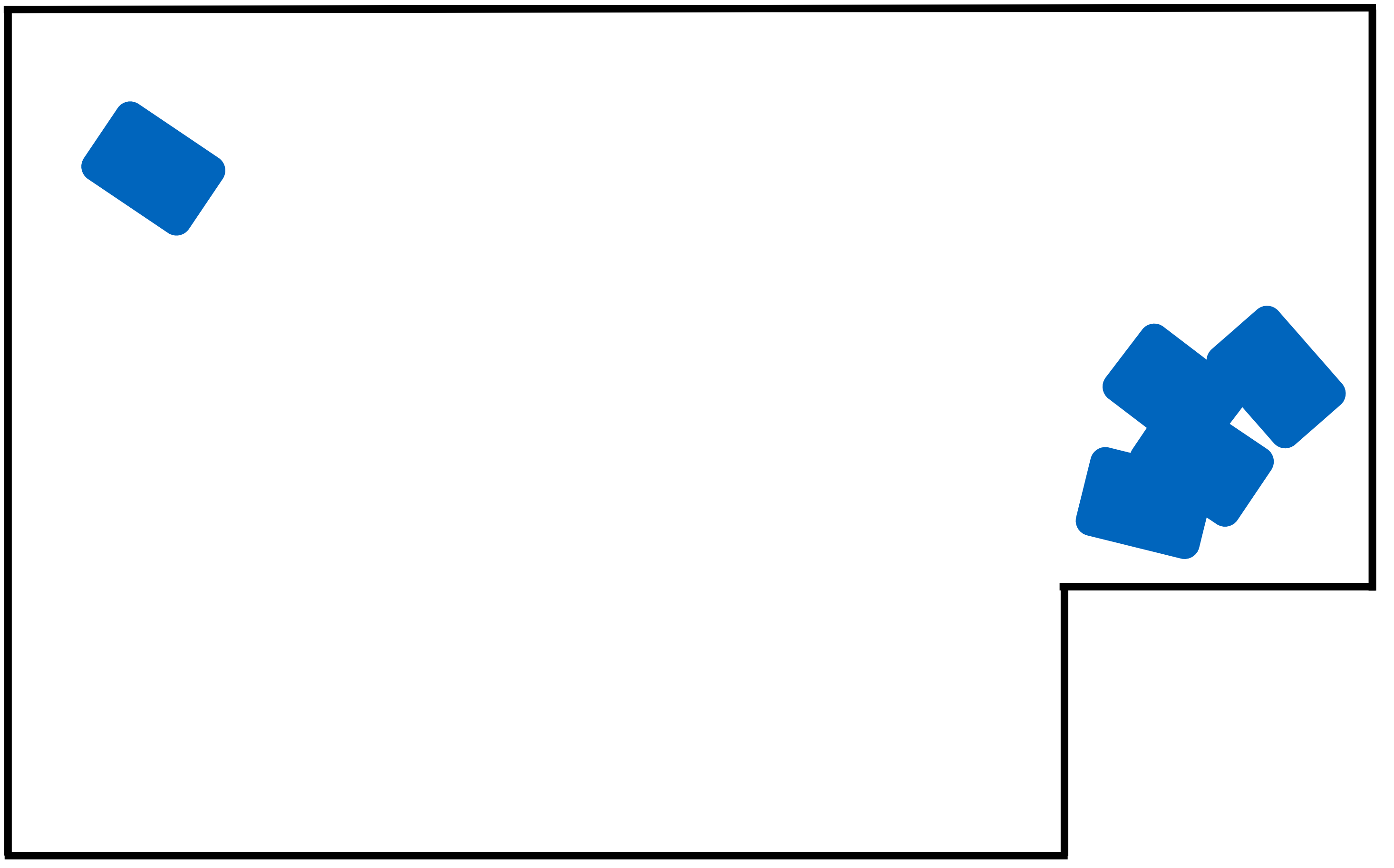
Particle Filter SLAM



Particle Filter SLAM



Particle Filter SLAM



Resampling

Particle Filter SLAM

Particle Filter SLAM - Pros and Cons

+ Non-Gaussian distributions can be modelled
+ Well scaling (1 million+ features)
+ Robust
+ Handling of nonlinearities

- Determination of optimal particle size
- Particle deprivation
- Only applicable for low dimensional spaces

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

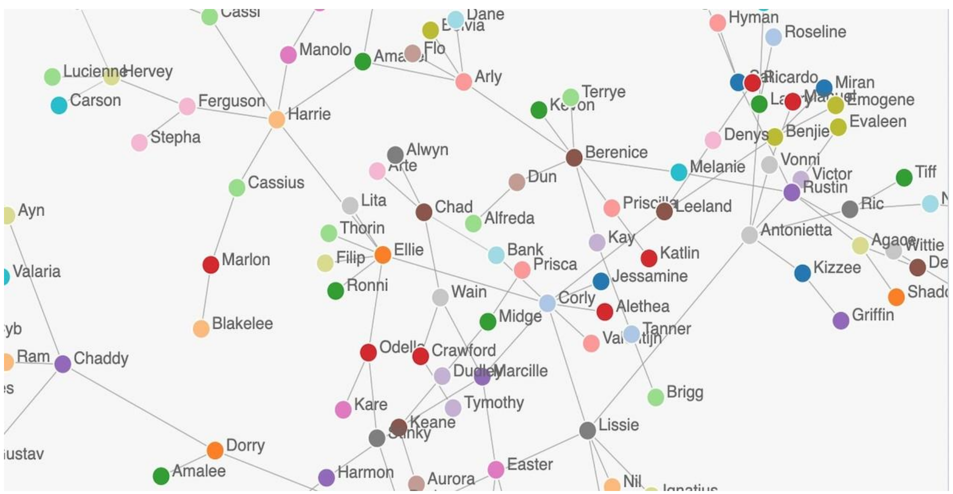
1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. **SLAM Paradigms**
    1. EKF SLAM
    2. Particle Filter SLAM
    3. **Graph-Based SLAM**
5. Front-Ends
6. Overview of existing solutions
7. Summary

Graph-Based SLAM

What is a graph?

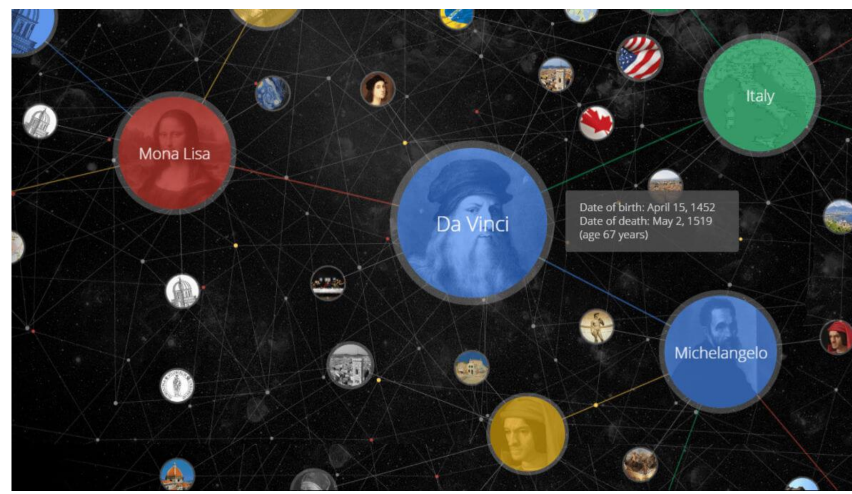
A model to represent objects (nodes) and relationships between them (edges)

Social graph



https://www.linkedin.com/pulse/build-social-network-javascript-graphs-fabian-hinsenkamp/

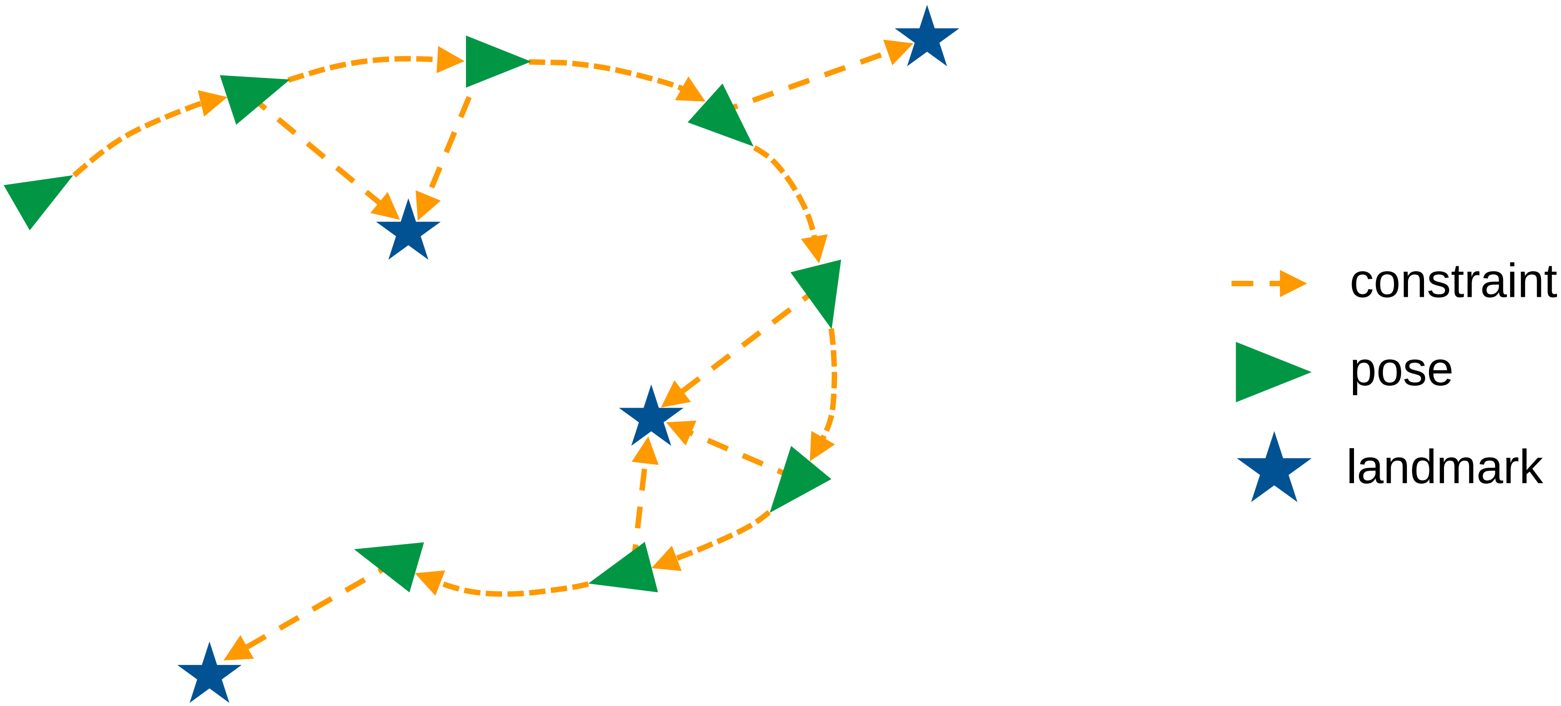
Google knowledge graph



https://searchengineland.com/leveraging-wikidata-gain-google-knowledge-graph-result-219706

Graph-Based SLAM

Exemplary Graph

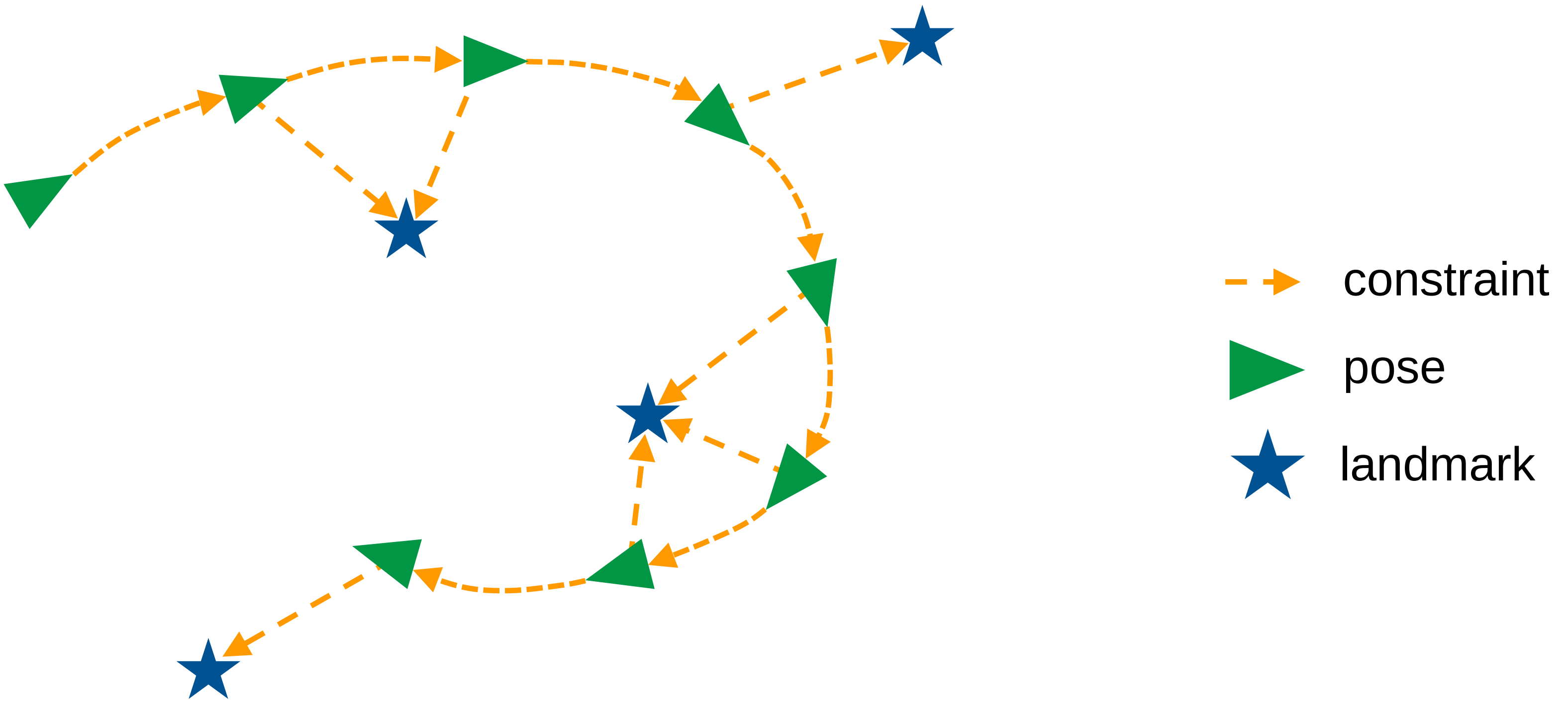


Nodes can be landmarks or ego poses

Edges can be landmark observations or motion observations (odometry)

Graph-Based SLAM

Exemplary Graph

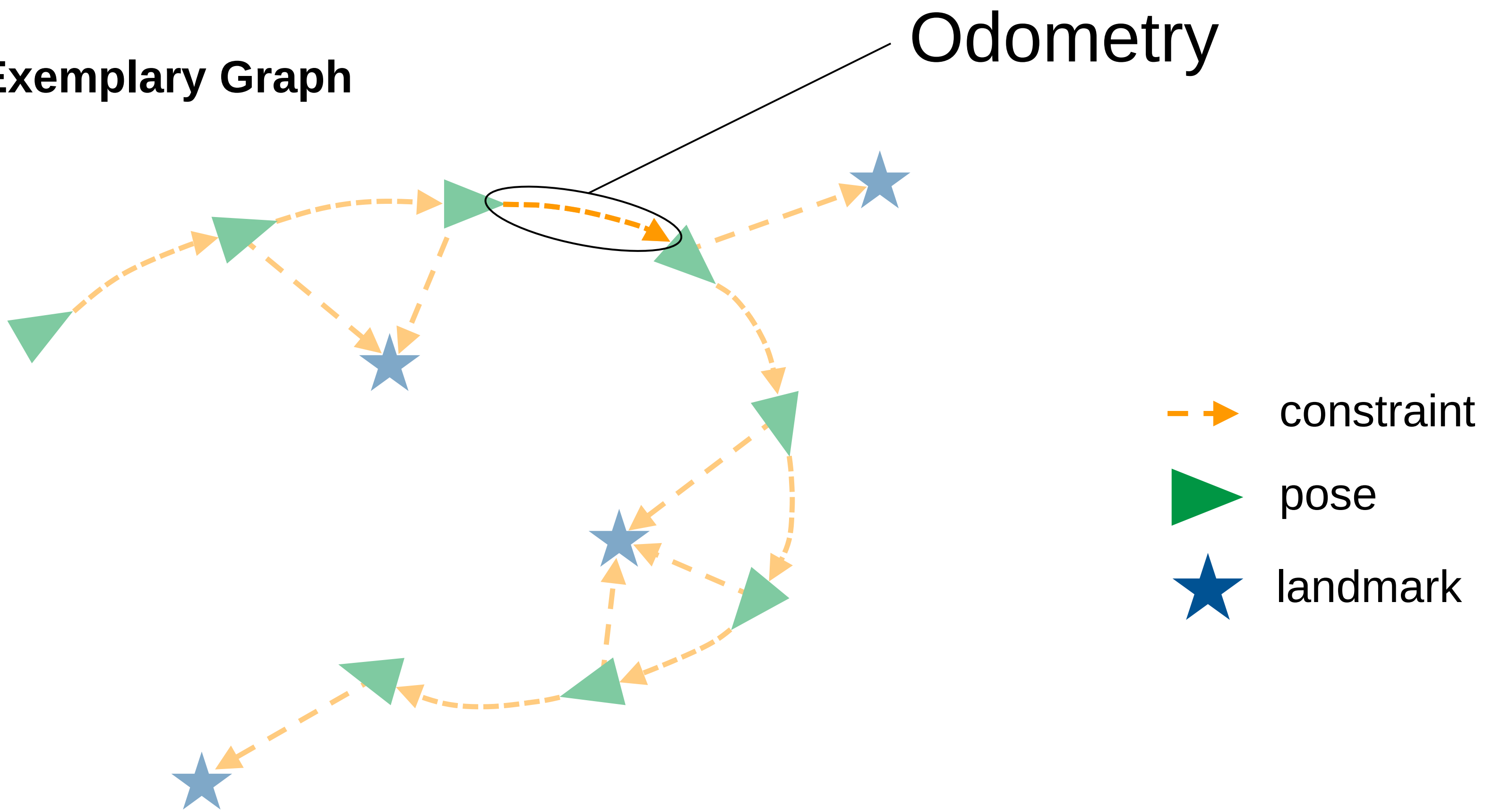


Nodes can be landmarks or ego poses

Edges can be landmark observations or motion observations (odometry)

Graph-Based SLAM

Exemplary Graph

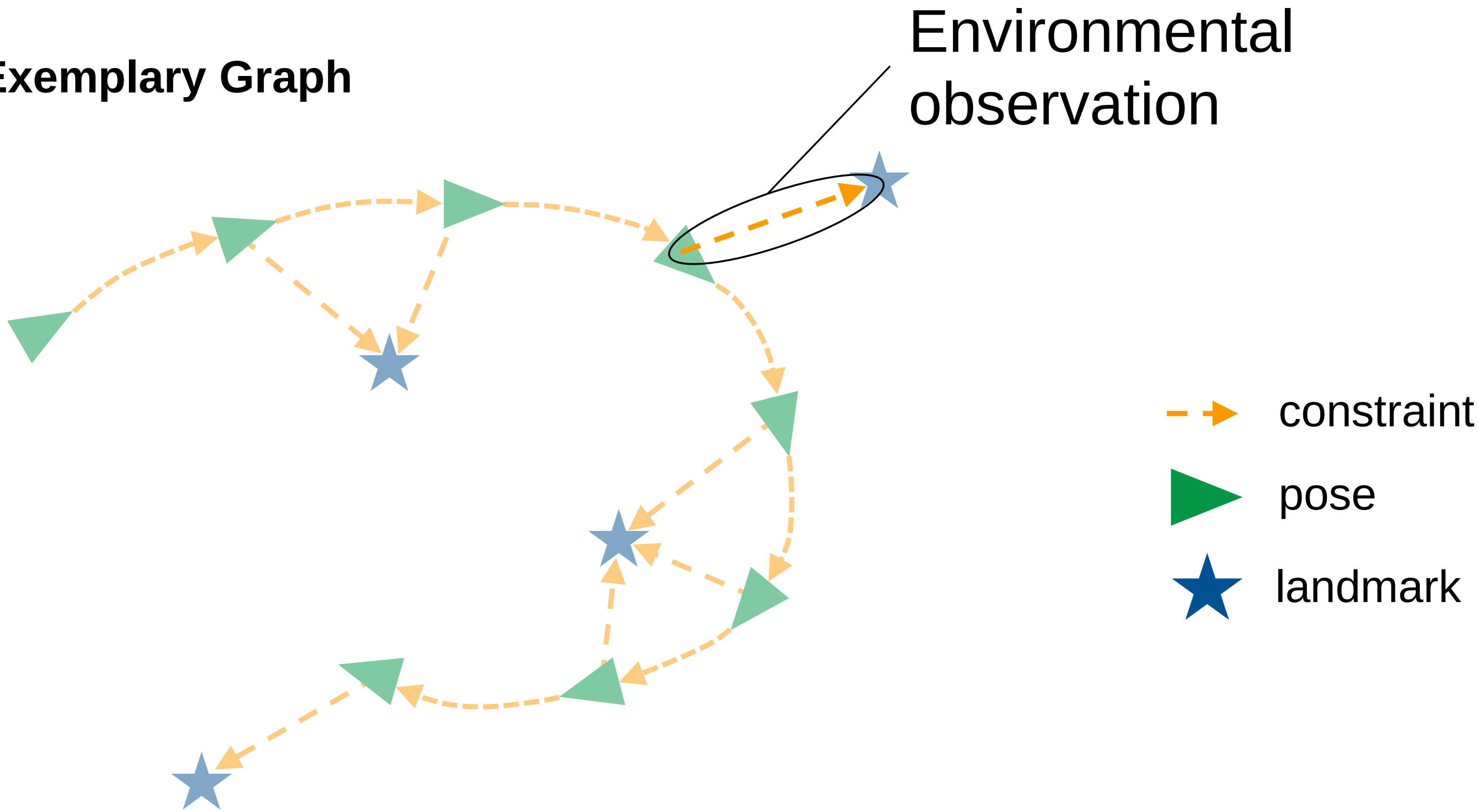


Nodes can be landmarks or ego poses

Edges can be landmark observations or motion observations (odometry)

Graph-Based SLAM

Exemplary Graph



Nodes can be landmarks or ego poses

Edges can be landmark observations or motion observations (odometry)

Graph-Based SLAM

Algorithm

Poses over time connected through constraints

Constraints are inherently uncertain

Observations of previous areas also generate constraints

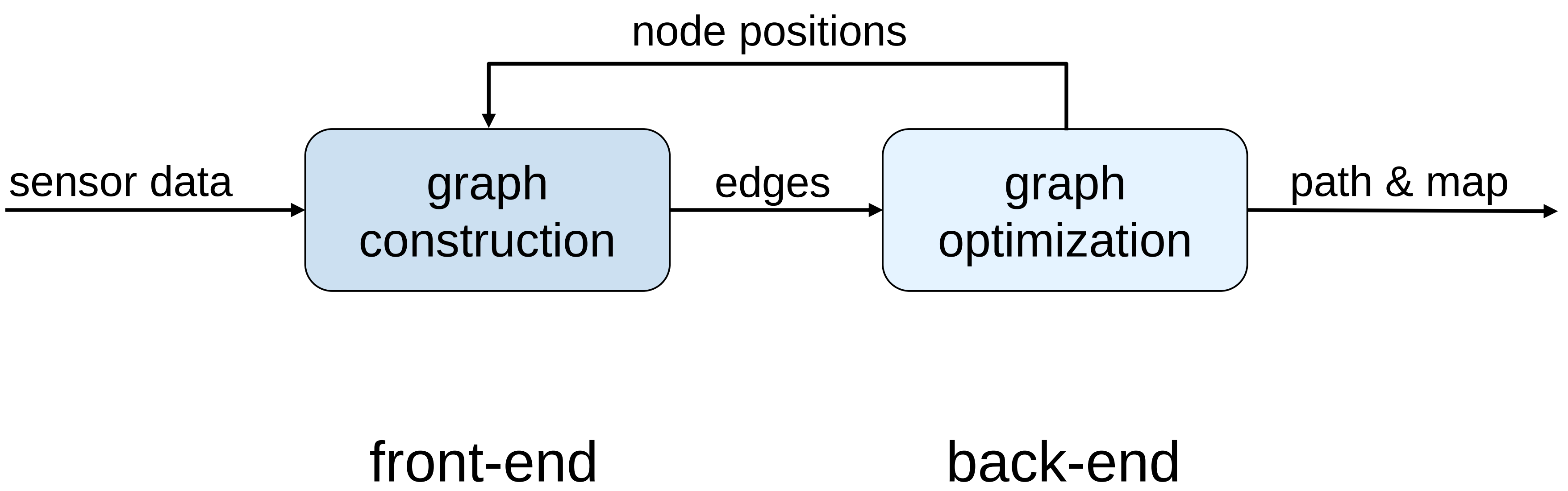
Node: pose of the ego and sensed landmarks during mapping

Edge: spatial constraint between nodes

Graph-Based SLAM: Build graph that minimizes errors by constraints

Graph-Based SLAM

Algorithm



Graph-Based SLAM

Exemplary Graph

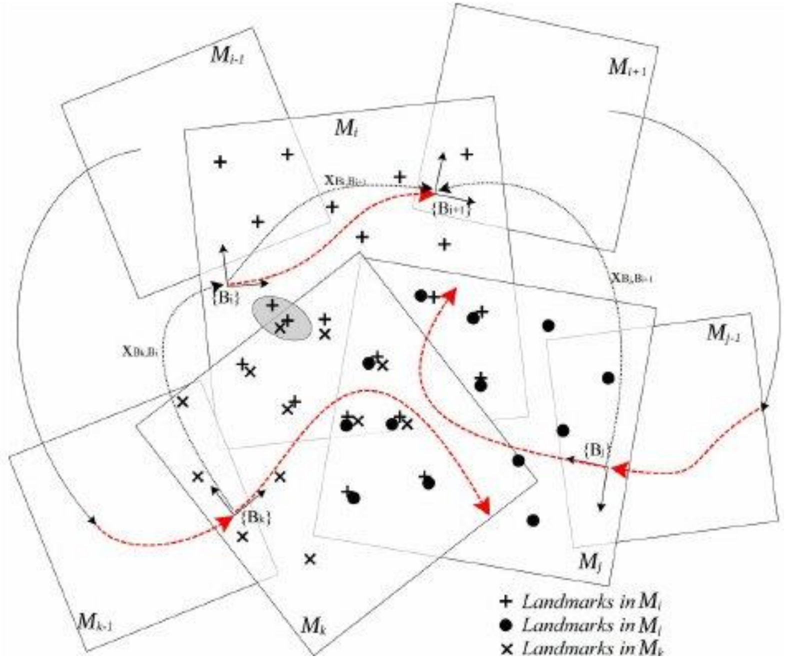
Problem:

Graph might become huge

Calculation times

Compute resources

→ **Hierarchical approach!**



https://www.researchgate.net/publication/221216146_Independent_Local_Mapping_for_Large-Scale_SLAM/figures?lo=1

Graph-Based SLAM

Hierarchical approach

To limit computation time for optimization and loop-closure search submaps are used

Submaps combine to one global map

Optimization takes place on two levels:

- Optimization within submap
- Optimization of submaps to one global map

Graph-Based SLAM

Toolboxes

Open source toolboxes for graph optimization can be used to build a SLAM system on.

C++

g2o: https://github.com/RainerKuemmerle/g2o

Ceres Solver: http://ceres-solver.org/

Python

g2opy: https://github.com/uoip/g2opy (Python binding of g2o)

Matlab

OptimizePoseGraph:
https://de.mathworks.com/help/nav/ref/optimizeposegraph.html

Graph-Based SLAM

Pros and Cons

+ Scale to much higher dimensions than EKF and PF

+ Update time stays constant

+ Possibilities to change optimization algorithm

- Optimization can be computationally expensive

- High development effort

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. SLAM Paradigms
    1. EKF SLAM
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. **Front-Ends**
6. Overview of existing solutions
7. Summary

Front-Ends

How to construct constraints from raw sensor data?

SLAM usually applied on LiDAR or camera data

LiDAR: one Layer (only 2D SLAM) or more layers (2D or 3D SLAM)

Camera: Mono-camera, stereo-camera, RGBD → No direct depth information

Front-Ends

What is a frame?

As frame we describe one single recording of a LiDAR (point cloud) or camera (image).

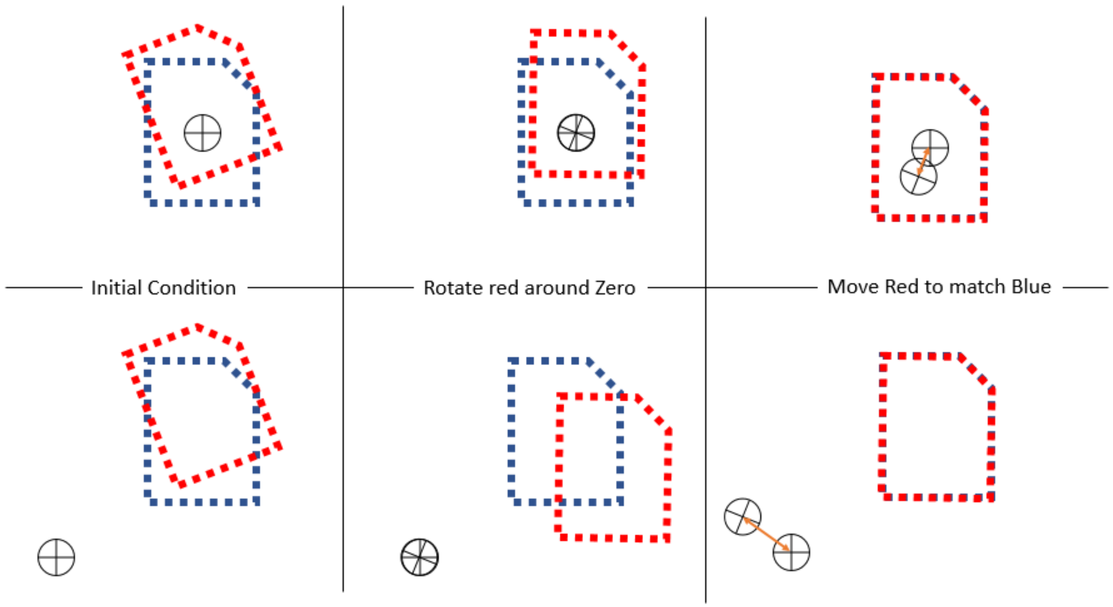
The data stream consists of frames coming at a specific recording frequency.

Front-Ends

LiDAR: Scan Matching

Incrementally align scans to a map or to previous scans

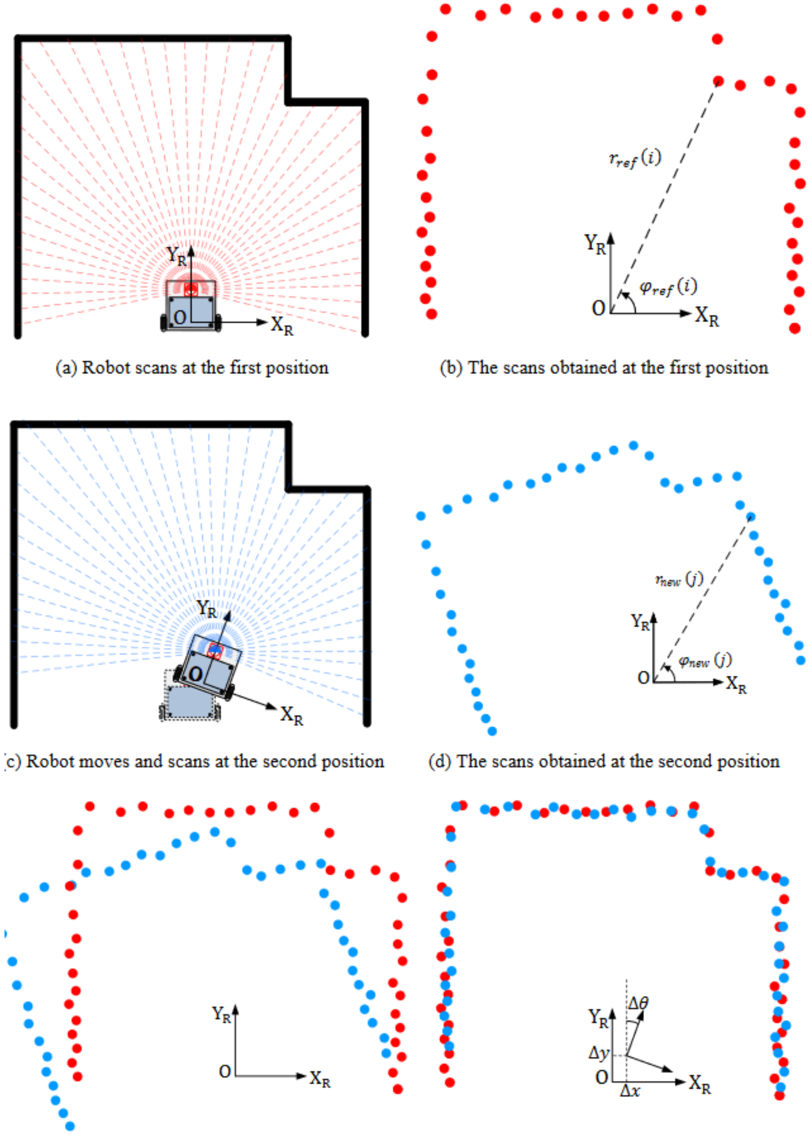
Different ways to realize: ICP, scan-to-scan, scan-to-map, feature-based, …



https://robotics.stackexchange.com/questions/4787/scan-matching-finds-right-rotation-but-false-translation

Front-Ends

LiDAR: Scan Matching



(a) Robot scans at the first position
(b) The scans obtained at the first position
(c) Robot moves and scans at the second position
(d) The scans obtained at the second position
(e)Map scans to the same coordinate
(f) Match scan and estimate tranformation

https://www.researchgate.net/profile/Riadh-Dhaoui/post/How_to_use_Normal_Distribution_Transformation_based_scan_matching_to_calculate_Pose_error_of_robot/attachment/5ed63f3298366a0001aa836b/AS%3A897949219434497%401591099185983/image/scan-matching.PNG

Front-Ends

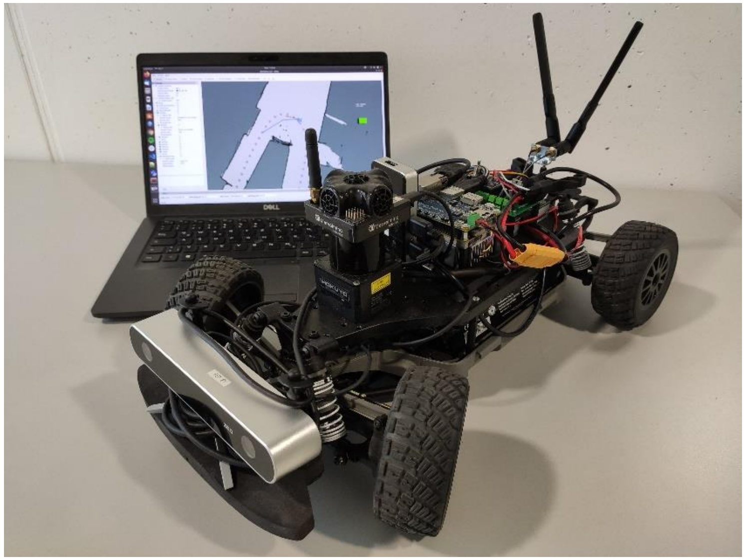
2D LiDAR SLAM

2D LiDAR sensors mainly used for mobile robotics on smaller scale

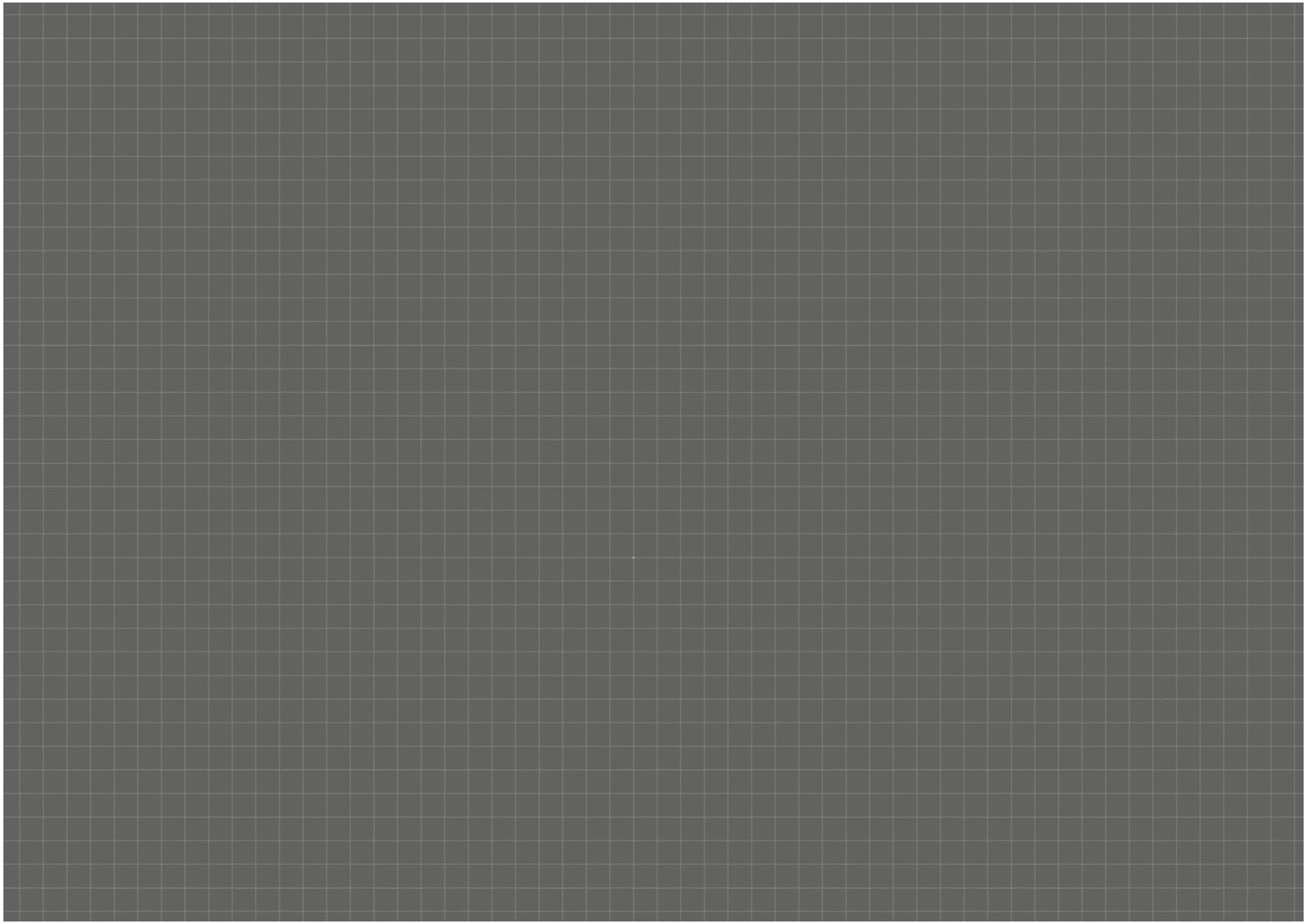
Not sufficient for autonomous vehicles in public areas

SLAM usually based on free / occupied space

Metric occupancy grid maps



Front-Ends: 2D-LiDAR SLAM (Cartographer)



Front-Ends

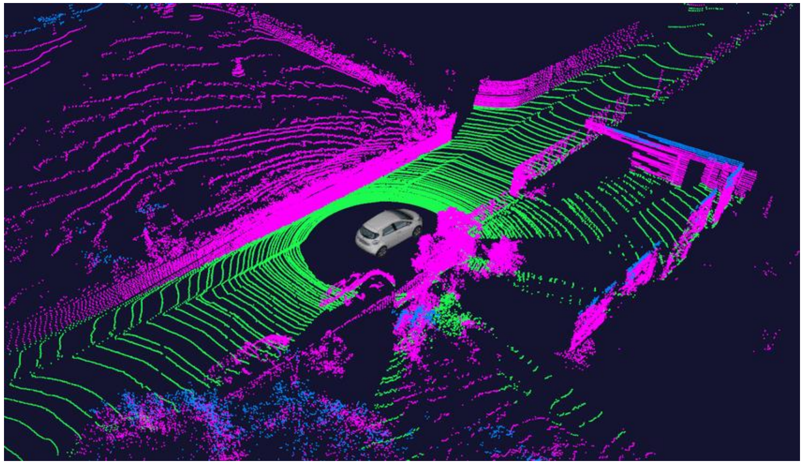
3D LiDAR SLAM

LiDAR sensors output direct depth information

Currently very expensive and mechanically susceptible

Limited frequencies <= 20 Hz

Low resolution compared to cameras

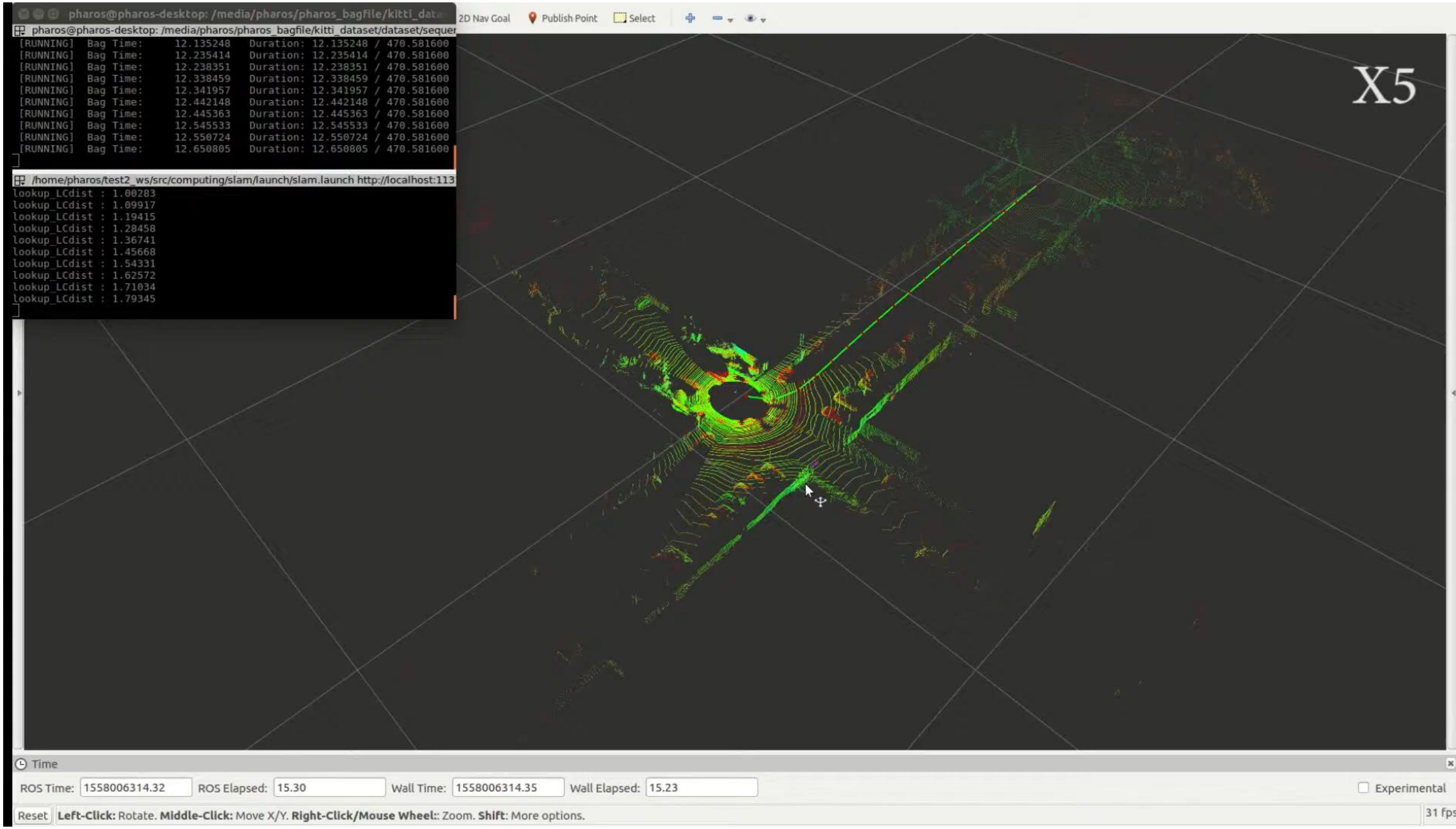


Detection of edges / faces

Struggling in low-feature-environments

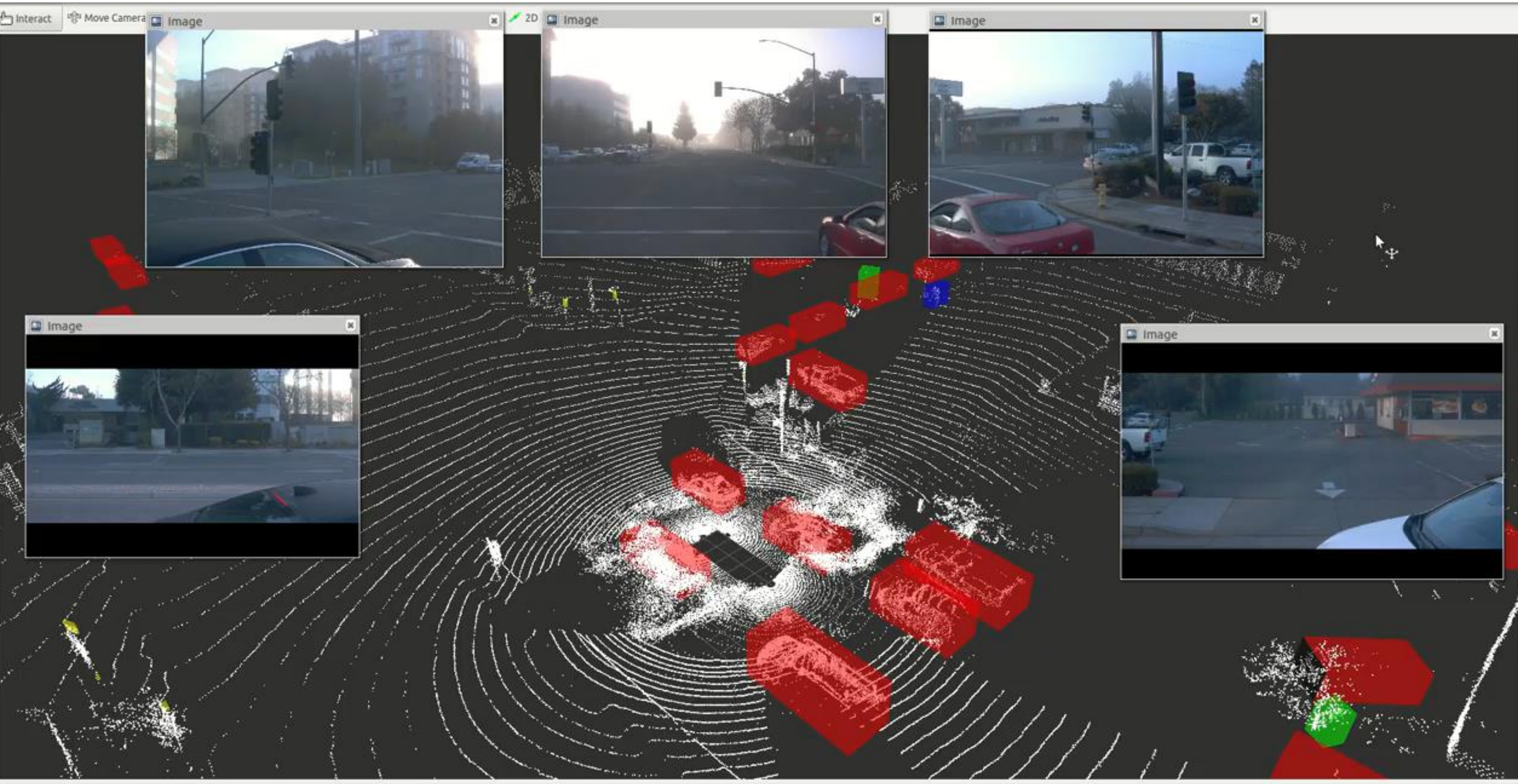
Distortion through scan pattern

3D LiDAR SLAM



https://www.youtube.com/watch?v=fvzluZIrjPA

LiDAR SLAM – Open Challenges



https://www.youtube.com/watch?v=aTv19KRk1-s

Front-Ends

Visual SLAM

Cameras are way cheaper than LiDAR sensors

Depth information has to be calculated / estimated

Direct: Matching of raw camera frames to estimate ego motion
→ **Photometric** error optimized

Indirect: Extraction of features to apply SLAM
→ **Geometric** error optimized

Front-Ends

Stereo cameras

Stereo cameras are cameras with two lenses in fixed positions.

When both record the same scene (big overlap of the images), the depth can be calculated via triangulation.

If you are interested in the working principle, check out these slides to get an overvies:

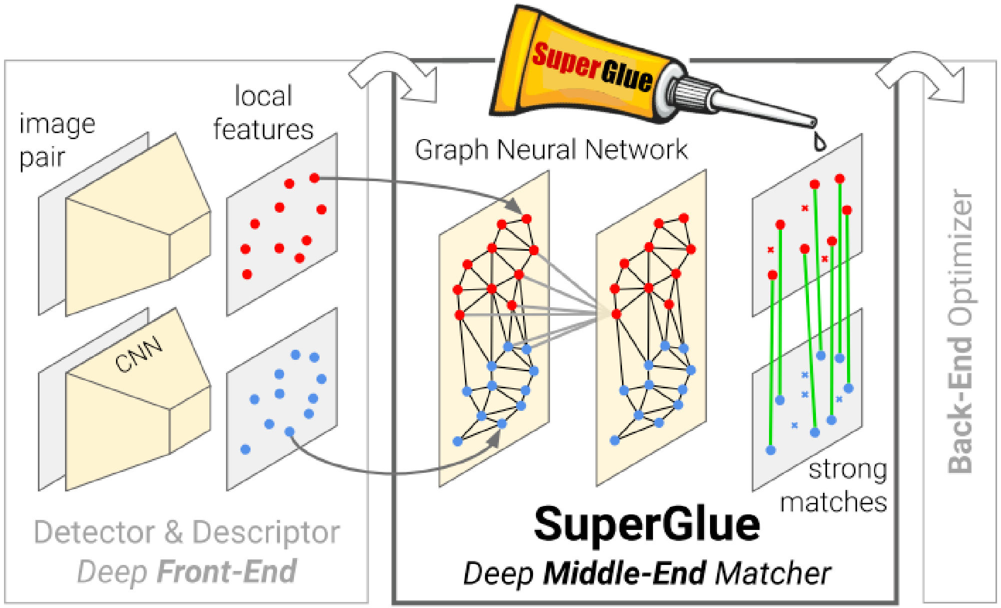
http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12_hres.pdf

Front-Ends

Visual SLAM: Feature extraction

State of the art: corner detection, SIFT, SURF, BRIEF, ORB

New approaches: Machine learning based



https://github.com/magicleap/SuperGluePretrainedNetwork

Front-Ends

Visual SLAM: Feature extraction

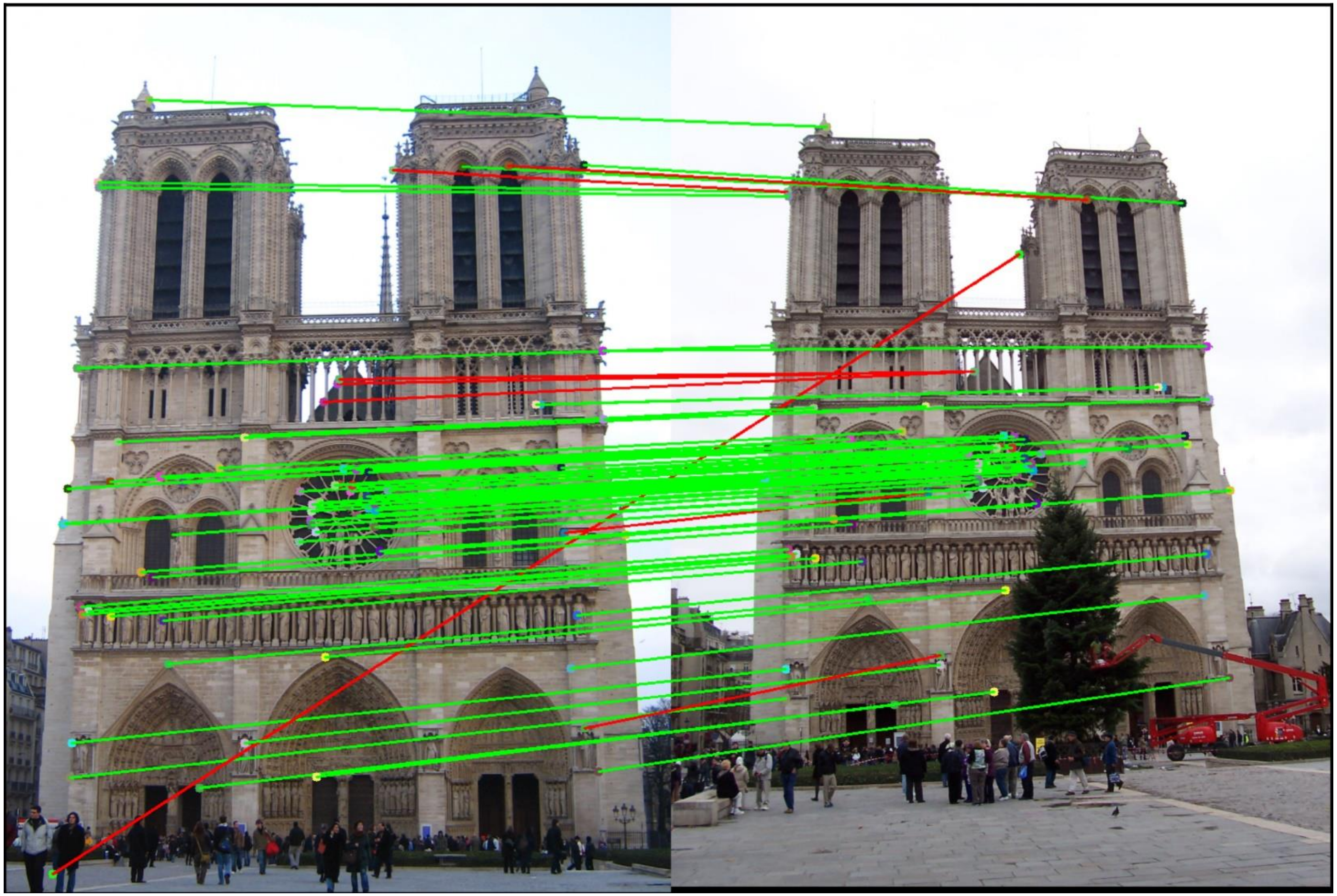
SIFT, SURF, BRIEF, ORB are techniqes for keypoint extraction and matching.

Keypoints are striking features in camera images. Those can be matched between frames. This gives us information about how the camera position and angle changes.

This is the basic idea of the direct visual SLAM.

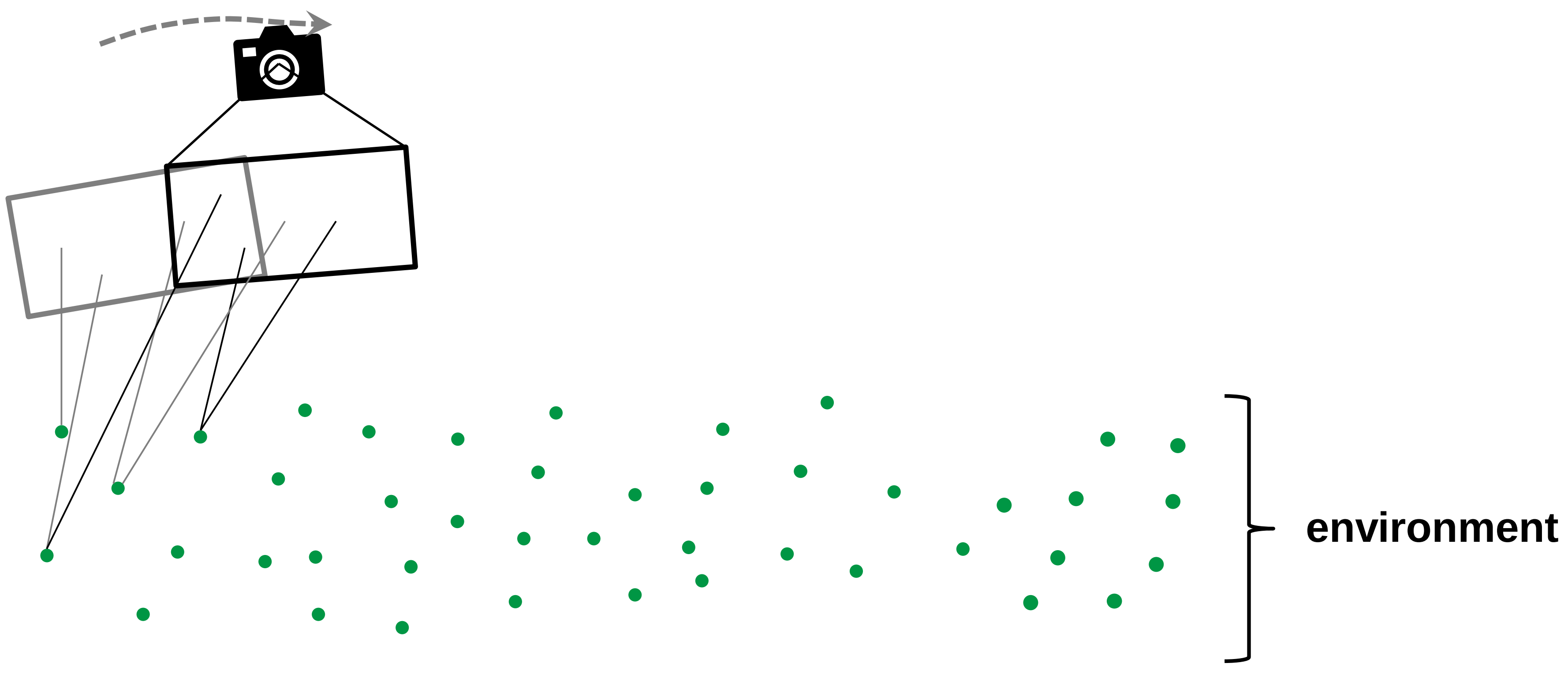
Front-Ends

Indirect Visual SLAM



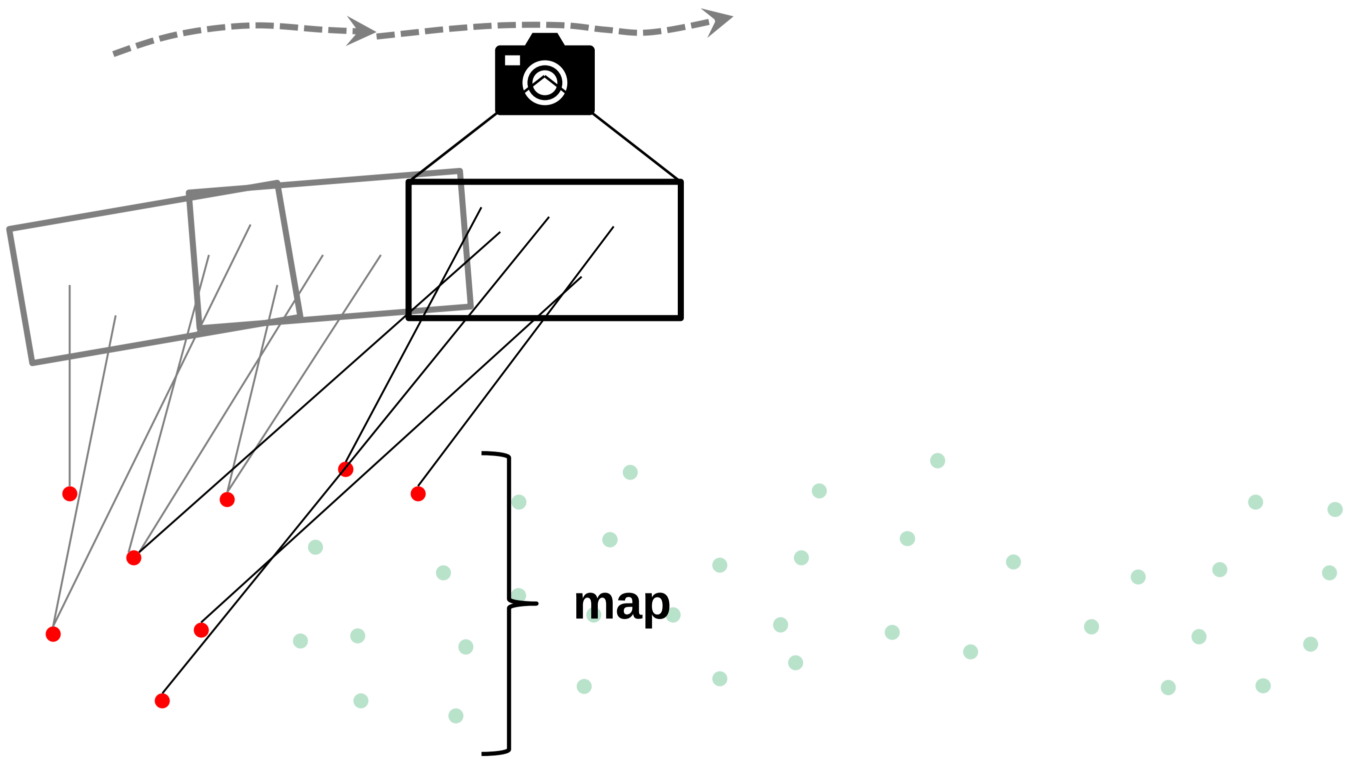
Front-Ends

Indirect Visual SLAM



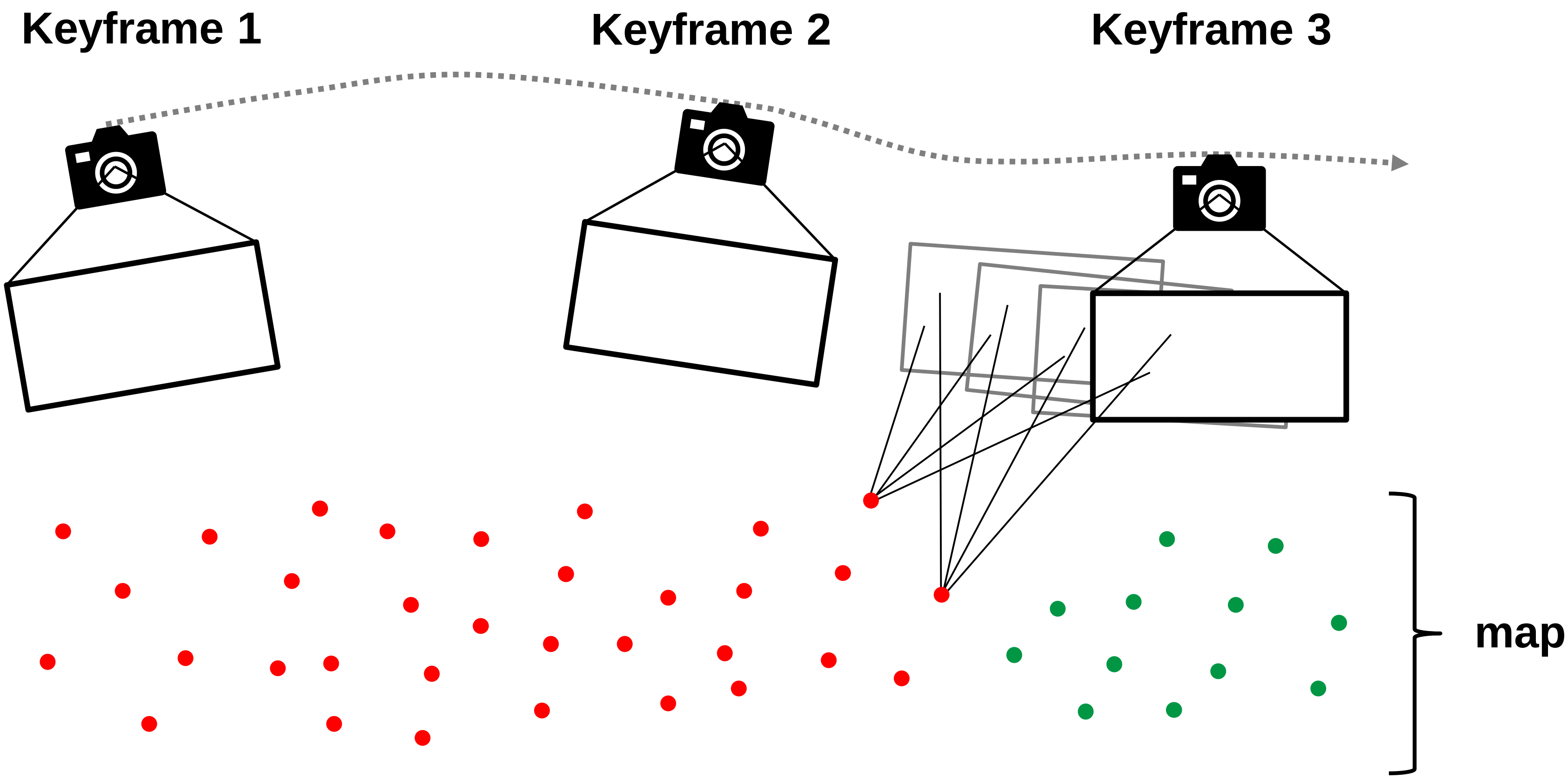
Front-Ends

Indirect Visual SLAM



Front-Ends

Indirect Visual SLAM



Front-Ends

Keyframes

When the features to be matched are too close to each other, uncertainties become quite big.

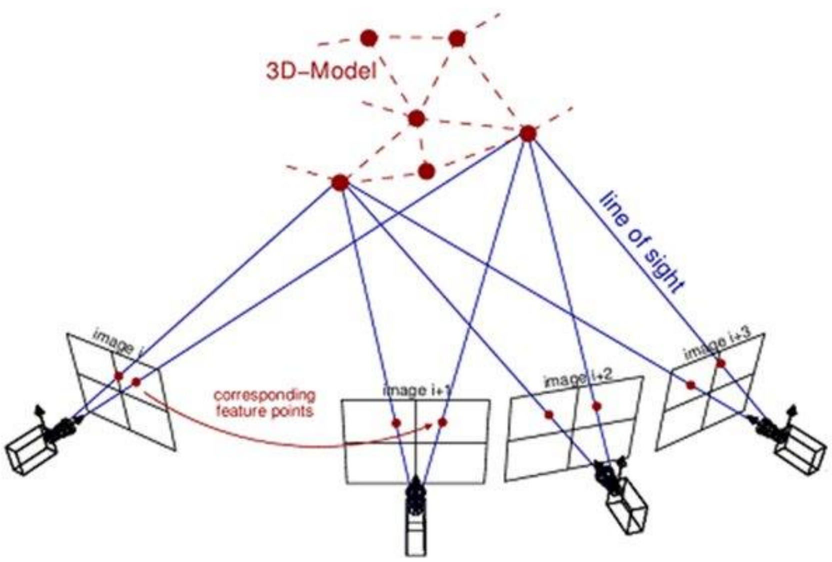
To avoid this, frames are skipped until the uncertainty drops blow a certain threshold. Those selected frames are called **keyframes**.

The rules, when keyframes are added can be used for parametrization of the algorithm.

Front-Ends

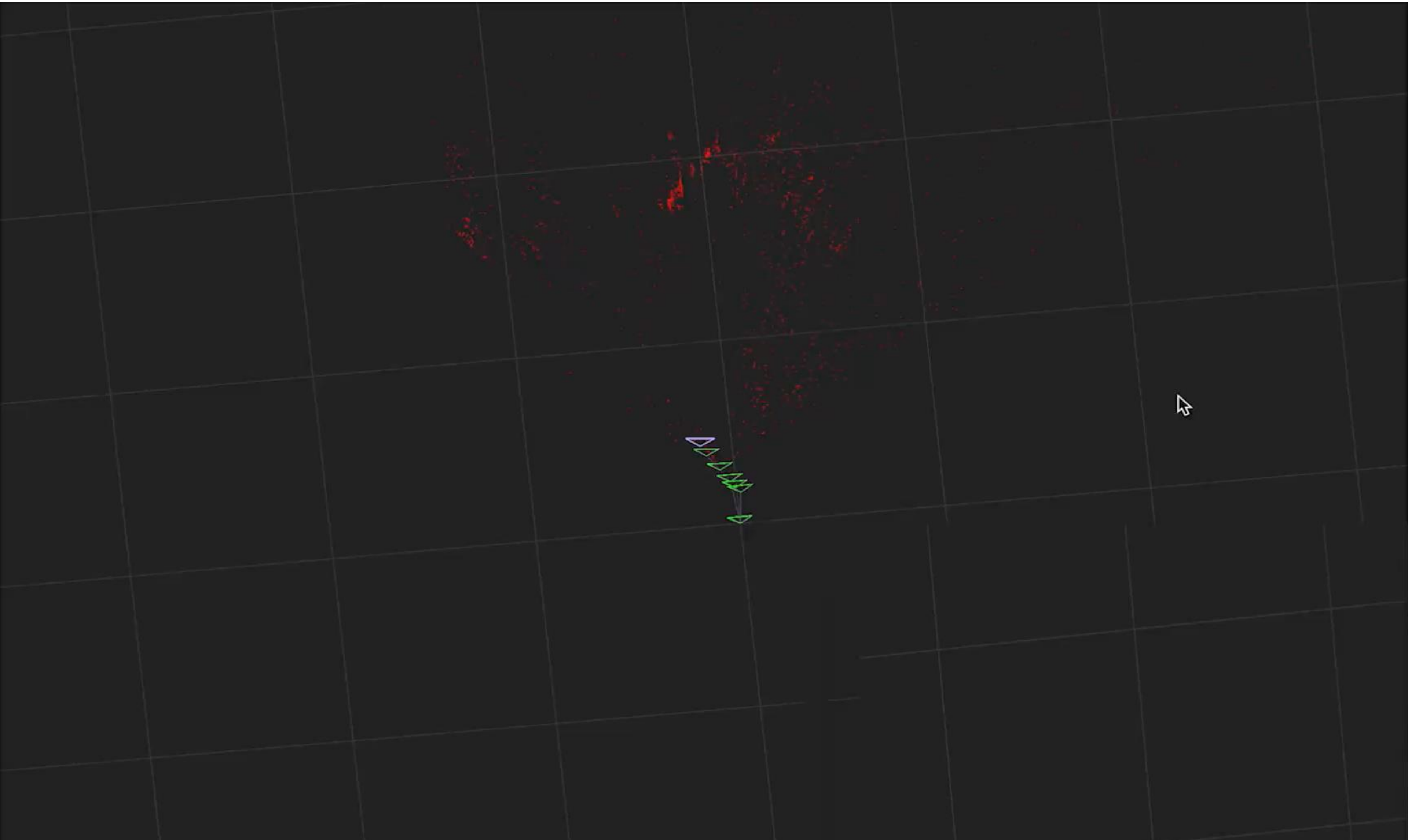
Bundle Adjustmen

Bundle adjustment is the problem of simultaneously estimating the 3D coordinates of detected features, the ego motion and the optical characteristics of the camera.



https://www.researchgate.net/publication/330555127_From_drones_to_geospatial_analysis

Visual SLAM – Indirect SLAM



Front-Ends

Indirect Method: Pros and Cons

+ Wide baseline matching
+ Better illumination invariance
+ Transition from image data to geometry

- Sparse map created
- Depending on edges
- Needs high resolution

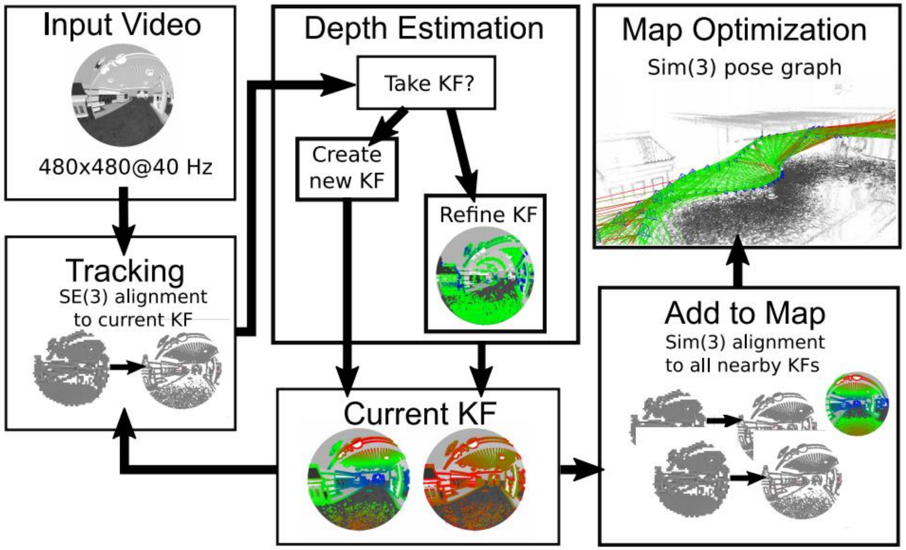
Front-Ends

Direct Visual SLAM

No use of geometric features

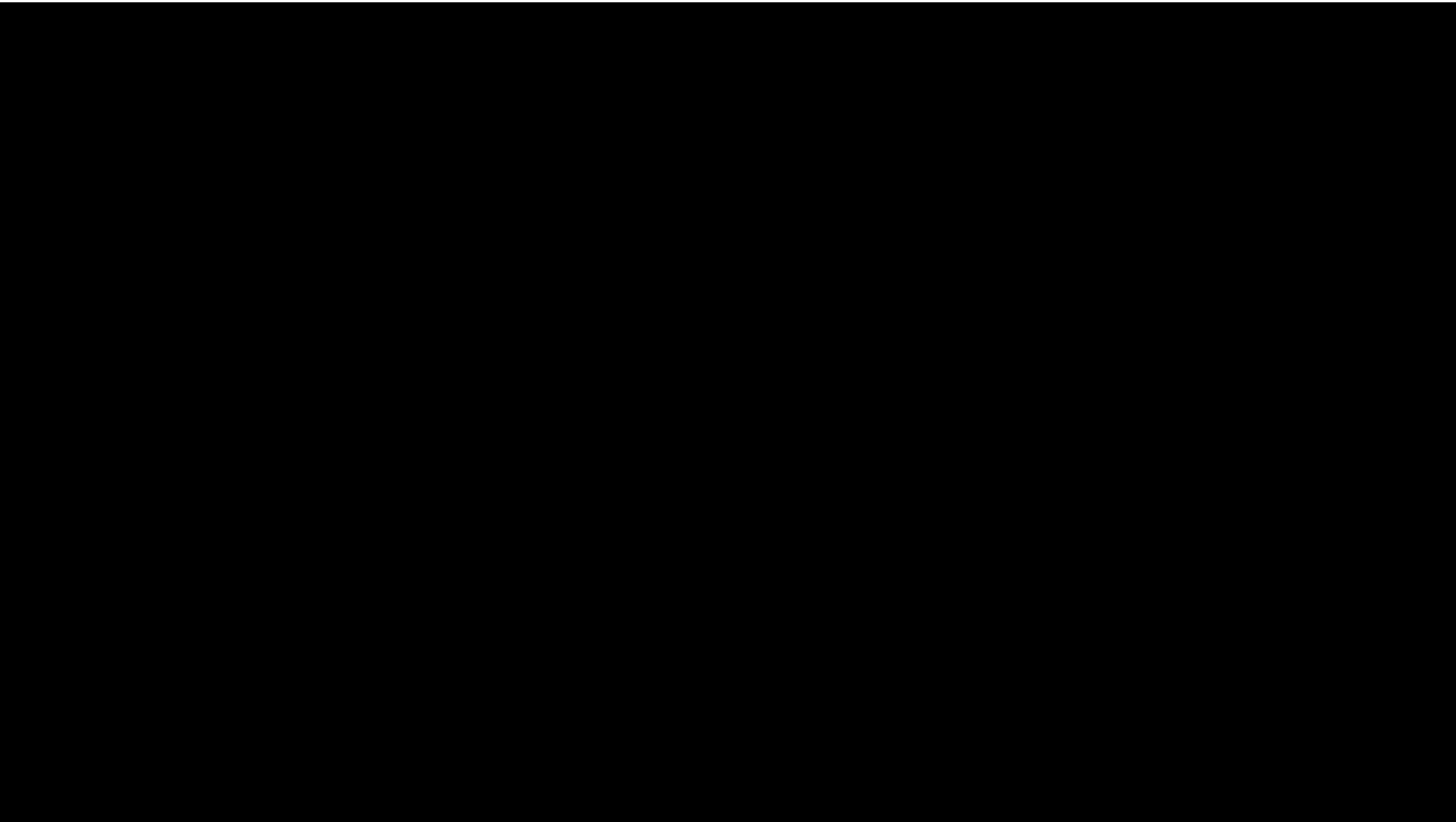
Camera model to determine photometric error

Minimization of photometric error



Large-Scale Direct SLAM with Stereo Cameras (J. Engel, J. Stueckler and D. Cremers), *In International Conference on Intelligent Robots and Systems (IROS)*, 2015

Visual SLAM – Direct SLAM



Front-Ends

Direct Method: Pros and Cons

+ Denser maps
+ Applicable for lower resolution
+ Distortion resistant
+ Includes all available data

- Heavily dependent on illumination
- High frequency camera
- Precise camera model necessary

Visual SLAM – Open challenges

https://www.youtube.com/watch?v=yi5sVTewmXc

Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. SLAM Paradigms
    1. EKF SLAM
    2. Particle Filter SLAM
    3. Graph-Based SLAM
5. Front-Ends
6. **Overview of existing solutions**
7. Summary

Overview of existing solutions

LiDAR SLAM

Name	Year	2D / 3D	ROS interface	Language	Link
Hector	2011	2D	ROS1	C++	http://wiki.ros.org/hector_slam
Gmapping	2017	2D	ROS1	C++	http://wiki.ros.org/gmapping
Cartographer	2016	2D / 3D	ROS1 (ROS2 under development)	C++	https://google-cartographer-ros.readthedocs.io/en/latest/
Hdl_graph_slam	2019	3D	ROS1	C++	https://github.com/koide3/hdl_graph_slam
PyICP-SLAM	2018	3D	/	Python	https://github.com/gisbi-kim/PyICP-SLAM
LeGO-LOAM	2018	3D	ROS1	C++	https://github.com/RobustFieldAutonomyLab/LeGO-LOAM
Lidarslam_ros2	2020	3D	ROS2	C++	https://github.com/rsasaki0109/lidarslam_ros2

Overview of existing solutions

Visual SLAM

Name	Year	Direct/indirect	Camera	Loop Closure	IMU	Link
LSD	2014	Direct	Mono	✓		https://vision.in.tum.de/research/vslam/lsdslam
SVO	2014	Semi-direct	Mono			https://github.com/uzh-rpg/rpg_svo
ORB SLAM 3	2020	Indirect	Mono, stereo, RGBD	✓	✓	https://github.com/UZ-SLAMLab/ORB_SLAM3
DSO	2016	Direct	Mono			https://vision.in.tum.de/research/vslam/dso
LDSO	2018	Direct	Mono	✓		https://vision.in.tum.de/research/vslam/ldso
OpenVSLAM	2019	Indirect	Mono, stereo, RGBD	✓		https://github.com/xdspacelab/openvslam
Basalt	2020	Indirect	Stereo	✓	✓	https://vision.in.tum.de/research/vslam/basalt

Overview of existing solutions

Visual SLAM

Name	Jahr	Open Source)	direkt/indirekt	Kameratyp	Multi-Kamera	IMU	Loop Closure	Relokalisierung
PTAM [9]	2007	✓	indirekt	Mono			✓	✓
LSD-SLAM [5]	2014	✓	direkt	Mono			✓	
SVO [6]	2014	✓	semi-direkt	Mono				
ORB-SLAM [12]	2015	✓	indirekt	Mono			✓	✓
ORB-SLAM2 [13]	2016	✓	indirekt	Mono, stereo, RGBD			✓	✓
DSO [4]	2016	✓	direkt	Mono				
SVO 2.0 [7]	2017		semi-direkt	Mono, stereo, RGBD	✓	✓	✓	✓
LDSO [8]	2018	✓	direkt	Mono			✓	
VINS-Mono [15]	2018	✓	direkt	Mono		✓	✓	✓
OpenVSLAM [18]	2019	✓	indirekt	Mono, stereo, RGBD			✓	✓
ORB-SLAM3 [2]	2020	✓	indirekt	Mono, stereo, RGBD		✓	✓	✓
Basalt [23]	2020	✓	indirekt	stereo		✓	✓	

Overview of existing solutions

Current Research

Dynamic SLAM [Deeb2019], [Fan2018], [Henein2020], [Saputra2018], [Jian2019]

Semantic SLAM [Yang2019], [Bowman2017]

Multi-Robot-SLAM [Nam2017], [Dubé2017]

Combined Camera-LiDAR-SLAM [Shin2018], [Chen2017], [Jian2019]

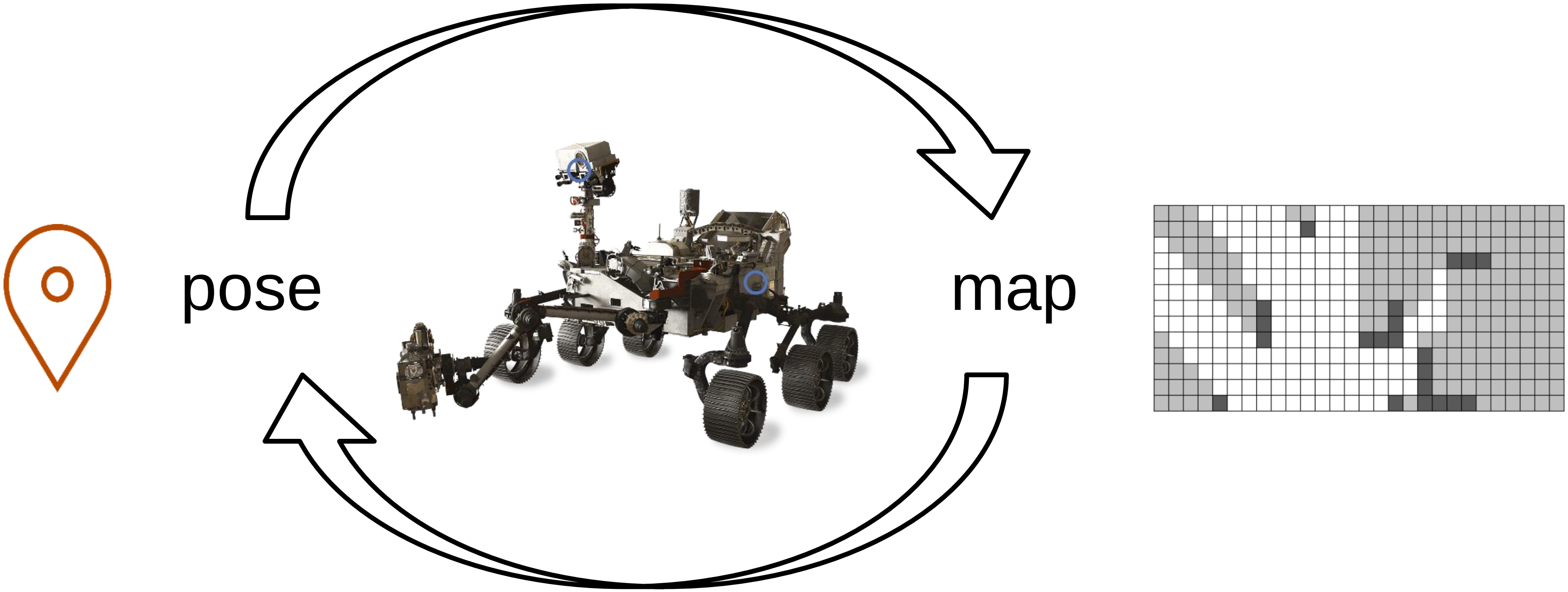
Mapping & Localization II
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Previously …
2. The SLAM Problem
3. The SLAM Algorithm
4. SLAM Paradigms
   1. EKF SLAM
   2. Particle Filter SLAM
   3. Graph-Based SLAM
5. Front-Ends
6. Overview of existing solutions
7. **Summary**

Summary – What did we learn today

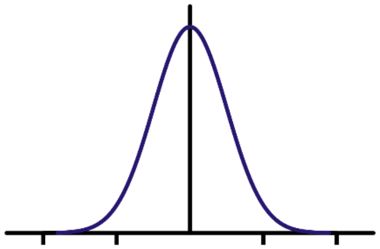


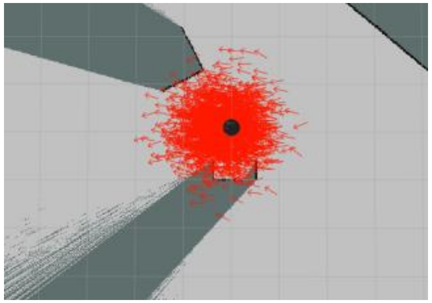
Summary – What did we learn today

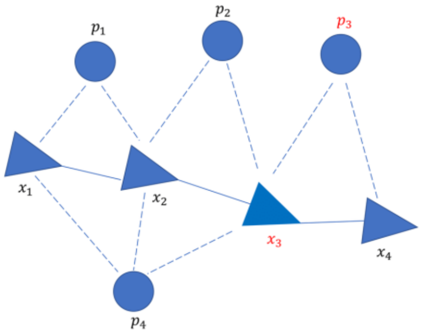
EKF

Particle

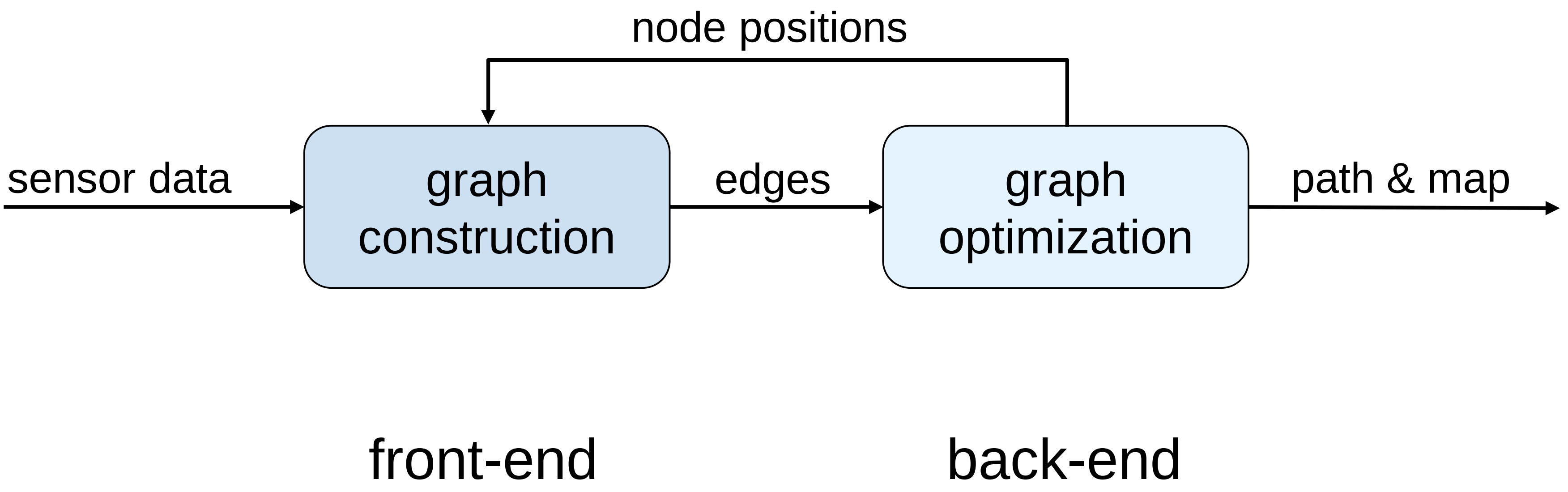
Graph-based



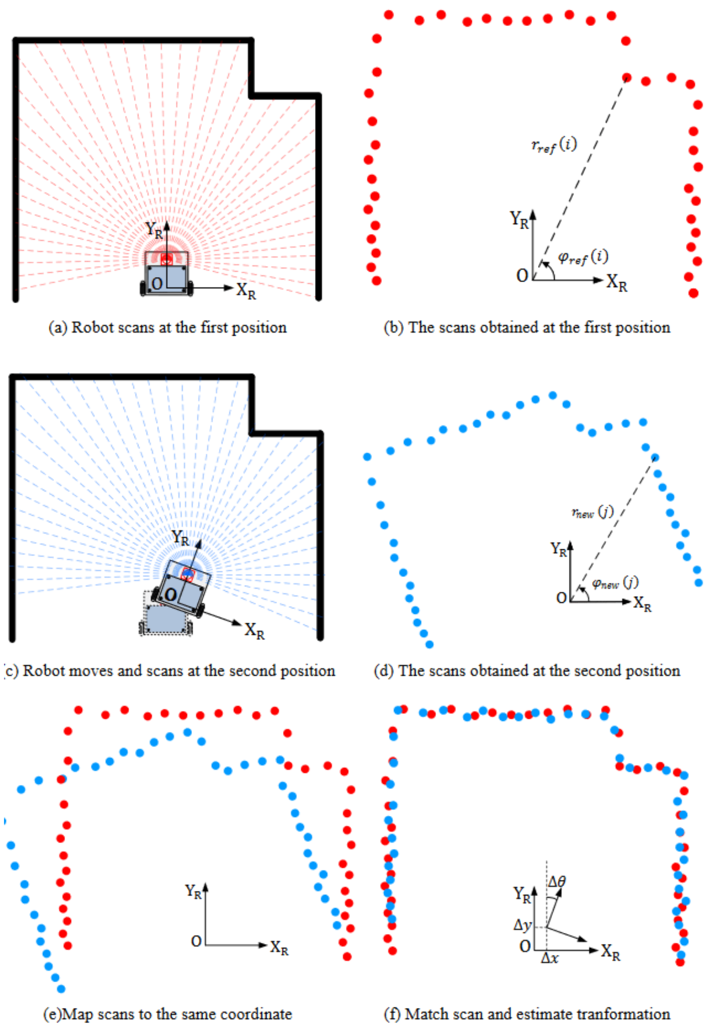


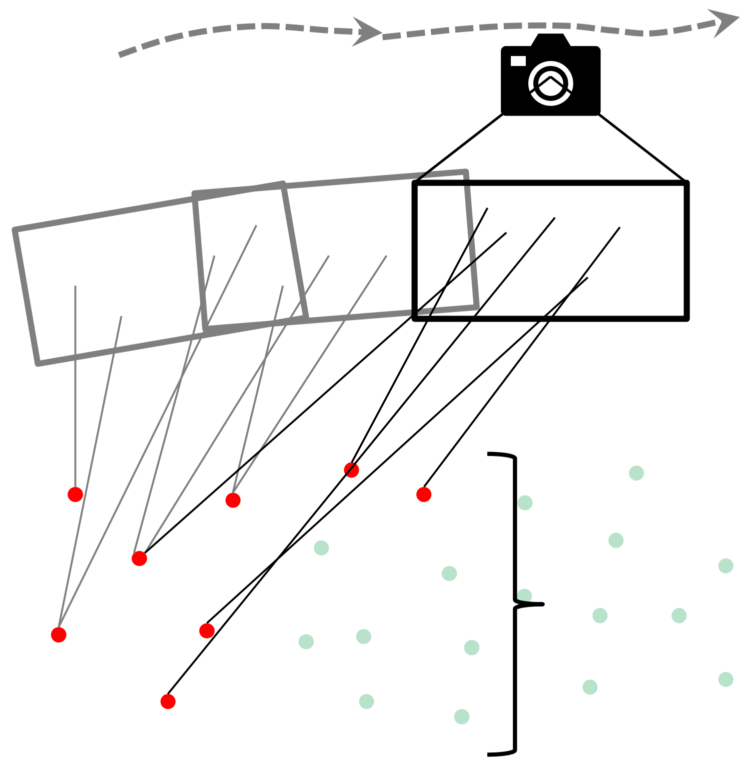


Summary – What did we learn today



Summary – What did we learn today





Summary – What did we learn today

SLAM: A chicken-and-egg-problem

Combination of information about ego vehicle and environmental perceptions

Three main paradigms: **EKF-SLAM**, **Particle-Filter SLAM** and **Graph-based SLAM**

EKF-SLAM: based on extended Kalman-Filter → Gaussian distribution

Particle-Filter SLAM:

- Estimation of path of each particle
- Probability based resampling of particles

Summary – What did we learn today

Graph-based SLAM

- Nodes: Poses of ego vehicle and landmarks
- Edges: Constraints through odometry and perceptions

“The application defines the filter”

Different front-ends

LiDAR SLAM: 2D and 3D

Visual SLAM: direct and indirect

Research still going on in many fields