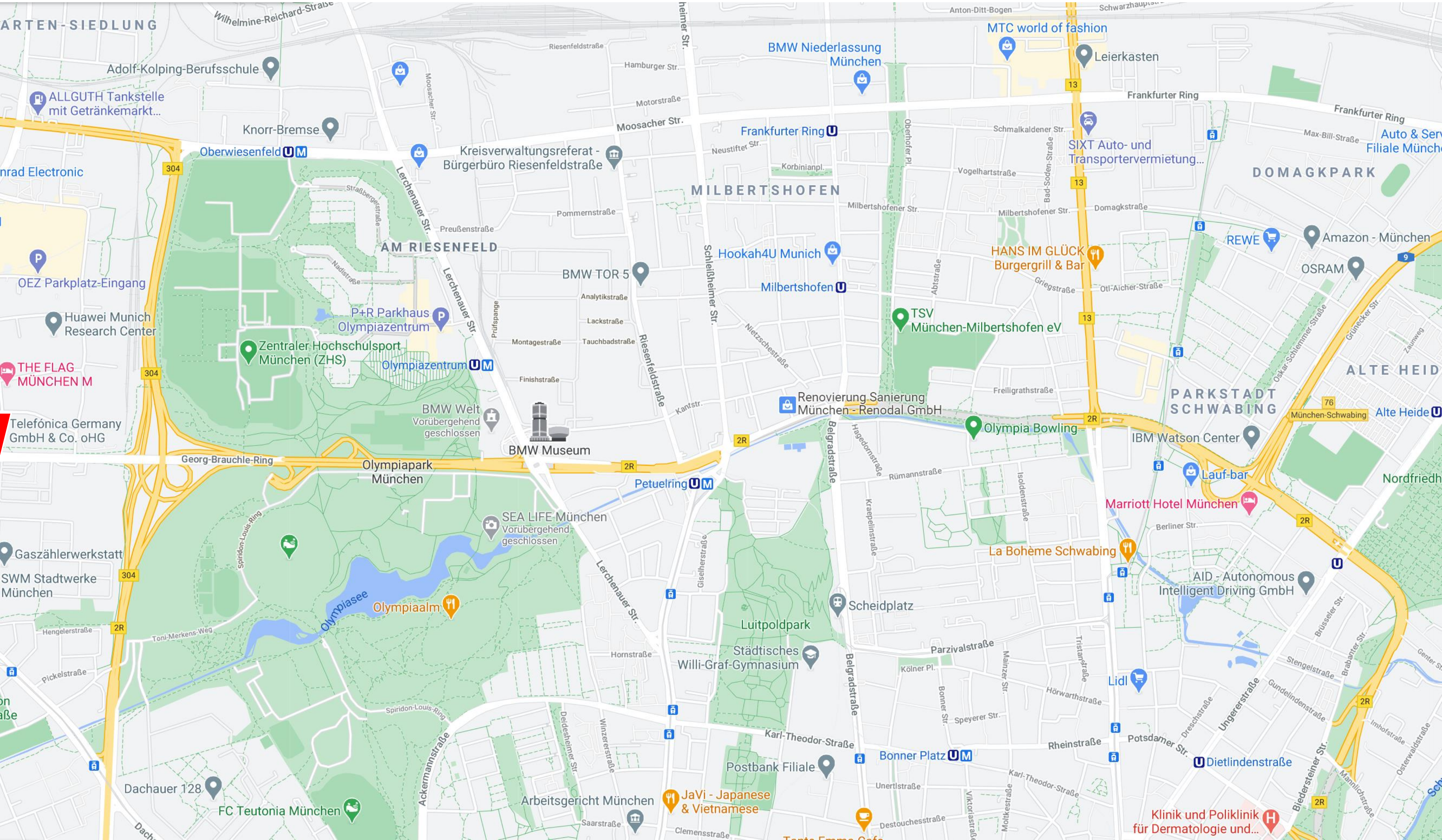
Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

What is happening here?

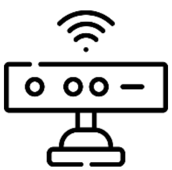


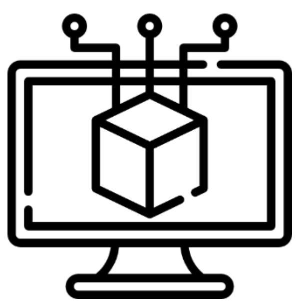
Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

Autonomous Driving Software Engineering

Prof. Dr.-Ing. Markus Lienkamp

Phillip Karle, M. Sc.





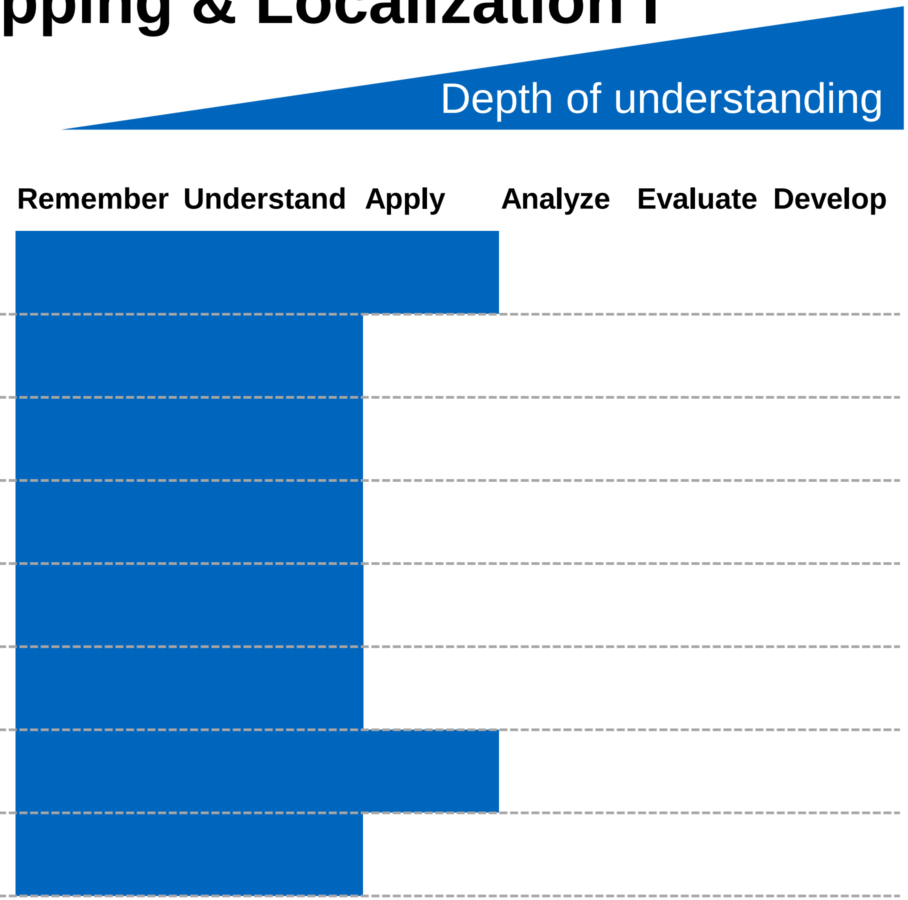
Lecture Overview

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 2: Mapping & Localization I

After the lecture you are able to…

- … are able to apply basic Bayesian methods
- … understand the different concepts and methods of probabilistic robotics: Kalman, Bayes, Monte-Carlo, …
- … understand how sensors and motion are modelled to estimate the robot state
- … understand how vehicle sensors are used for state estimation
- … to implement and analyze a basic Kalman Filter
- … understand how different sensors combine for mapping purposes
- … know the differences of map representation, how they are created and what they are used for
- … understand the need of HD maps and how there can be approaches without (see Tesla)



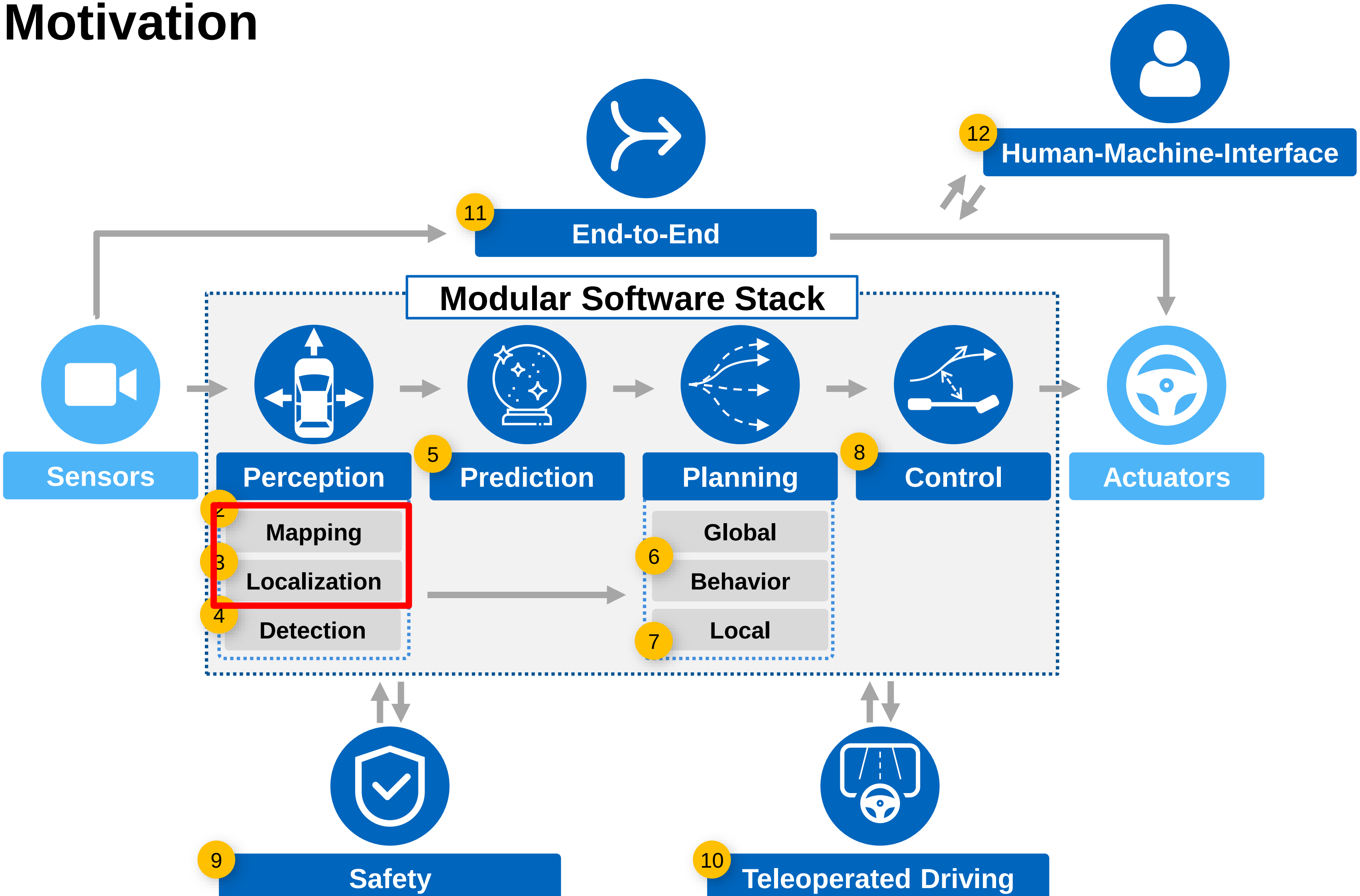
Localization & Mapping I
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. **Motivation**
2. Introduction to Probabilistics
3. Bayesian Filtering
4. Probabilistic Localization
5. Map representations
6. Summary

Motivation



X = Lectures

Motivation

How to measure position?

Position can not be measured directly

→ Has to be estimated

→ Fusion of data from different sources

IMU

Wheel Encoders

Steering Angle Sensors

Camera

LiDAR

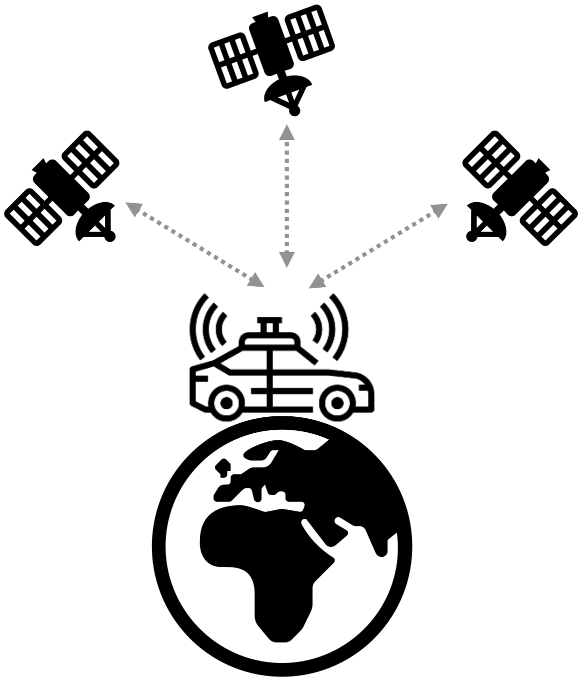
Radar

GNSS (distance to satellites)

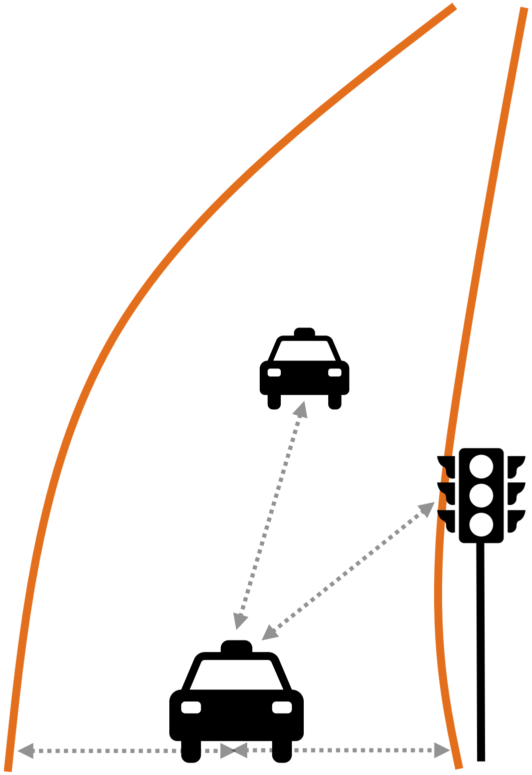
→ Vehicle Ego State

→ Perceive Environment

Motivation
Position & Localization



Global Position



Relative Position

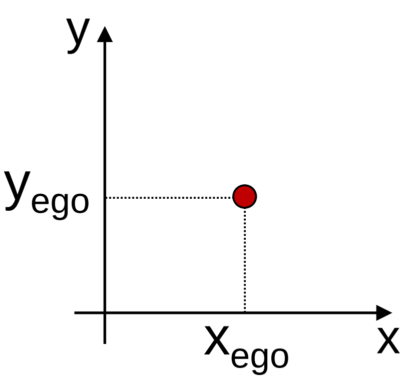
Motivation

Definition

Position

Coordinate vector relative to the global coordinate system

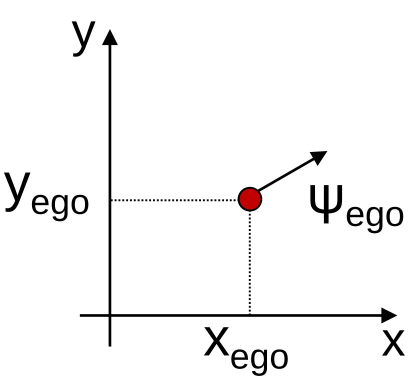
→ translation



Pose

Position + Heading ψ

→ translation + rotation



Localization & Mapping I
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

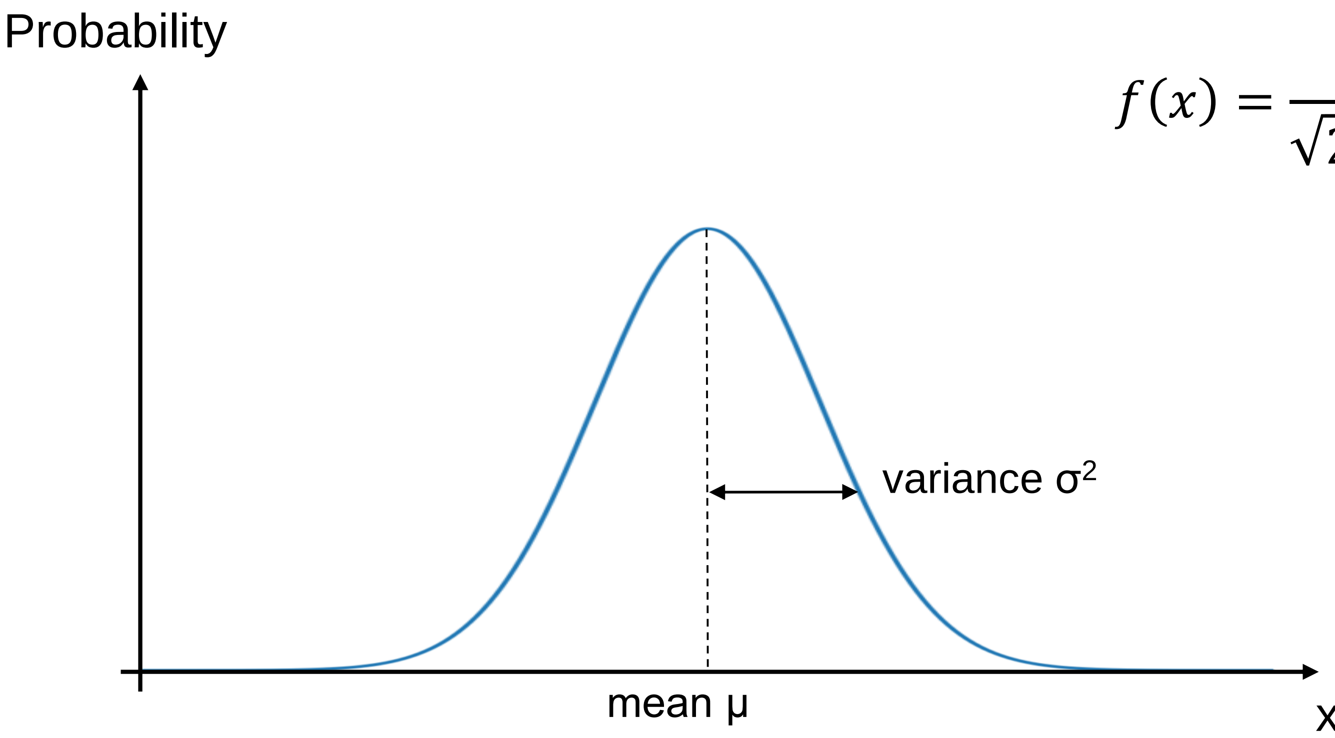
Agenda

1. Motivation
2. **Introduction to Probabilistics**
3. Bayesian Filtering
4. Probabilistic Localization
5. Map representations
6. Summary

Introduction to Probabilistics – Gaussian Distribution

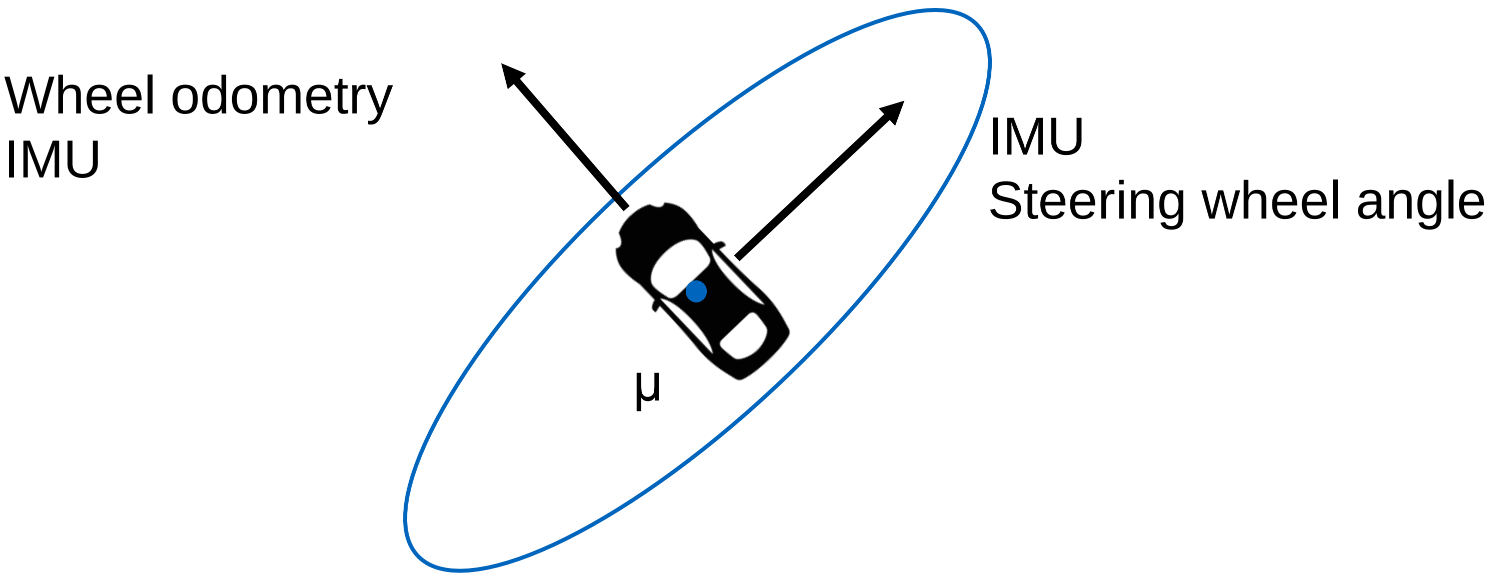
- Measurements can be represented as Gaussians
- Defined by mean μ and variance σ^2

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$



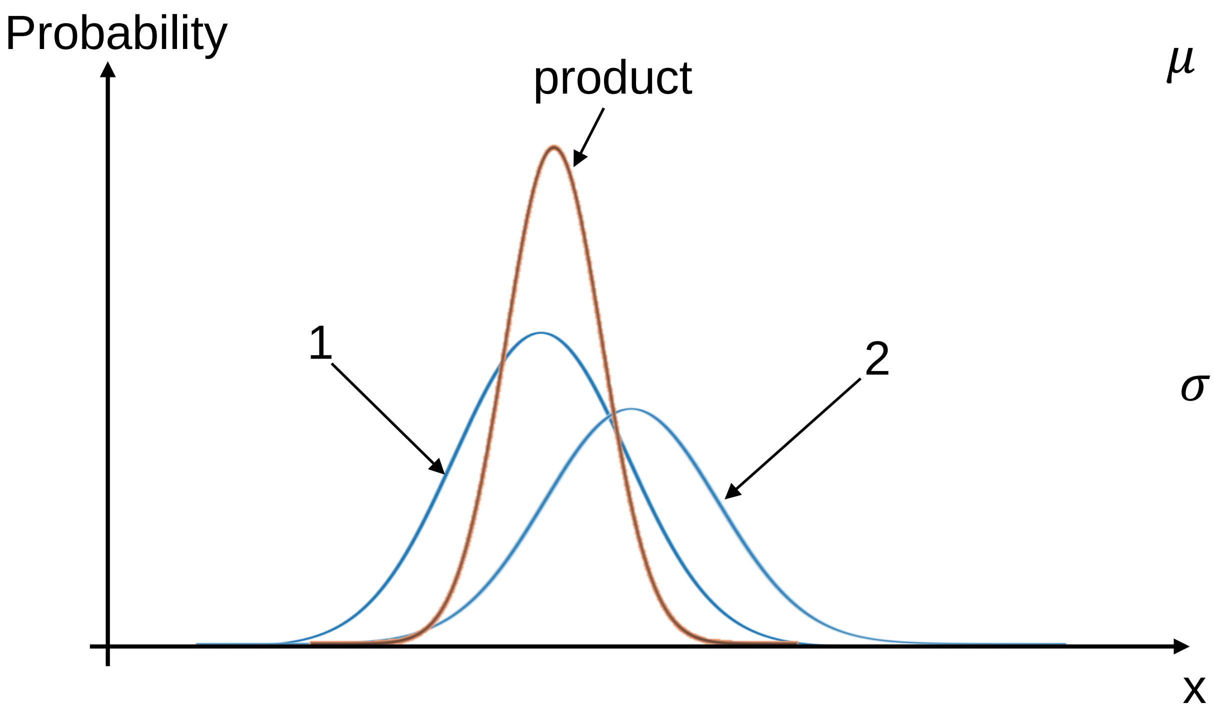
Introduction to Probabilistics – Gaussian Distribution

- Multivariate Gaussian
- Defined by mean μ and covariance Σ
- Σ can have different values in different dimensions



Multiplication of Gaussians

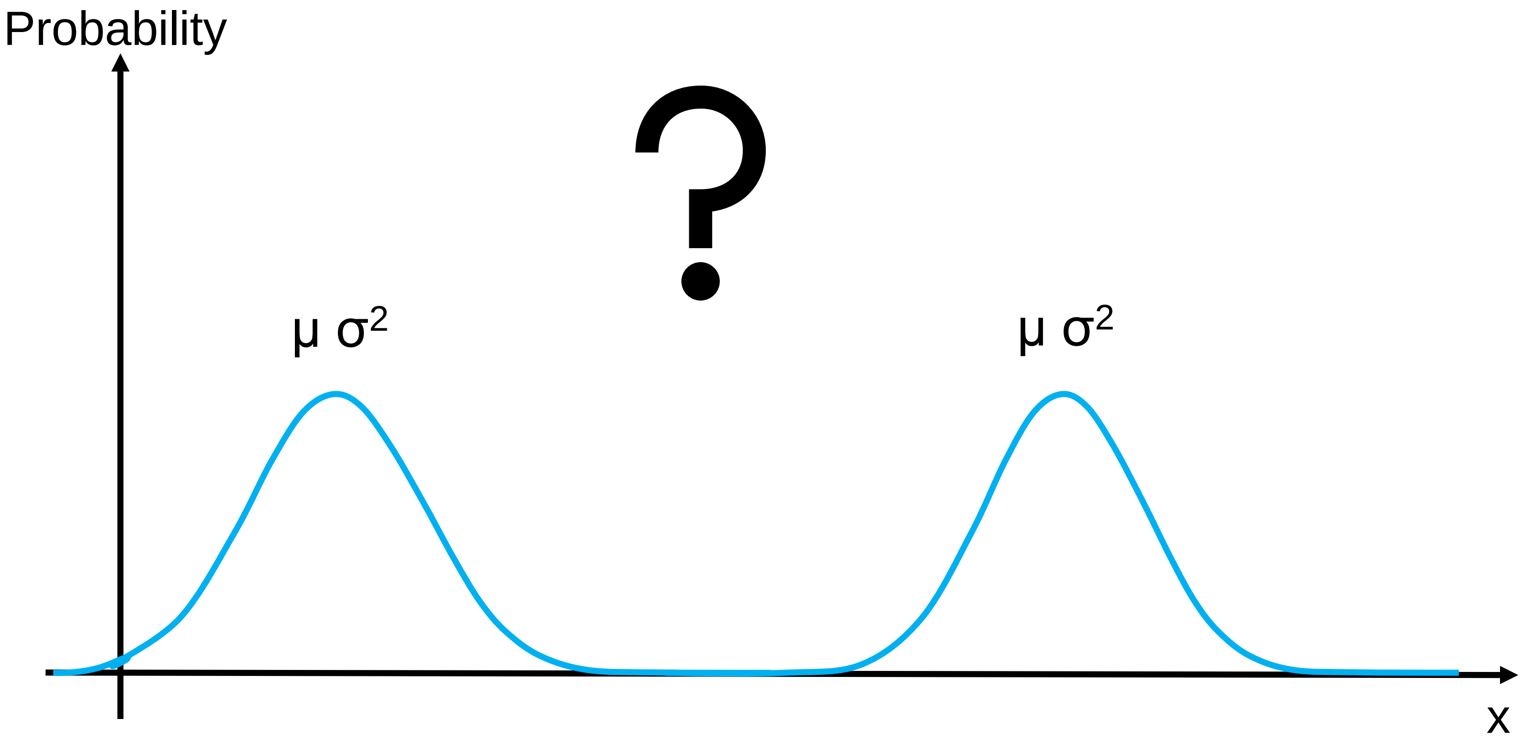
- Multiplication of Gaussians
- Multiple sensor inputs for same variable



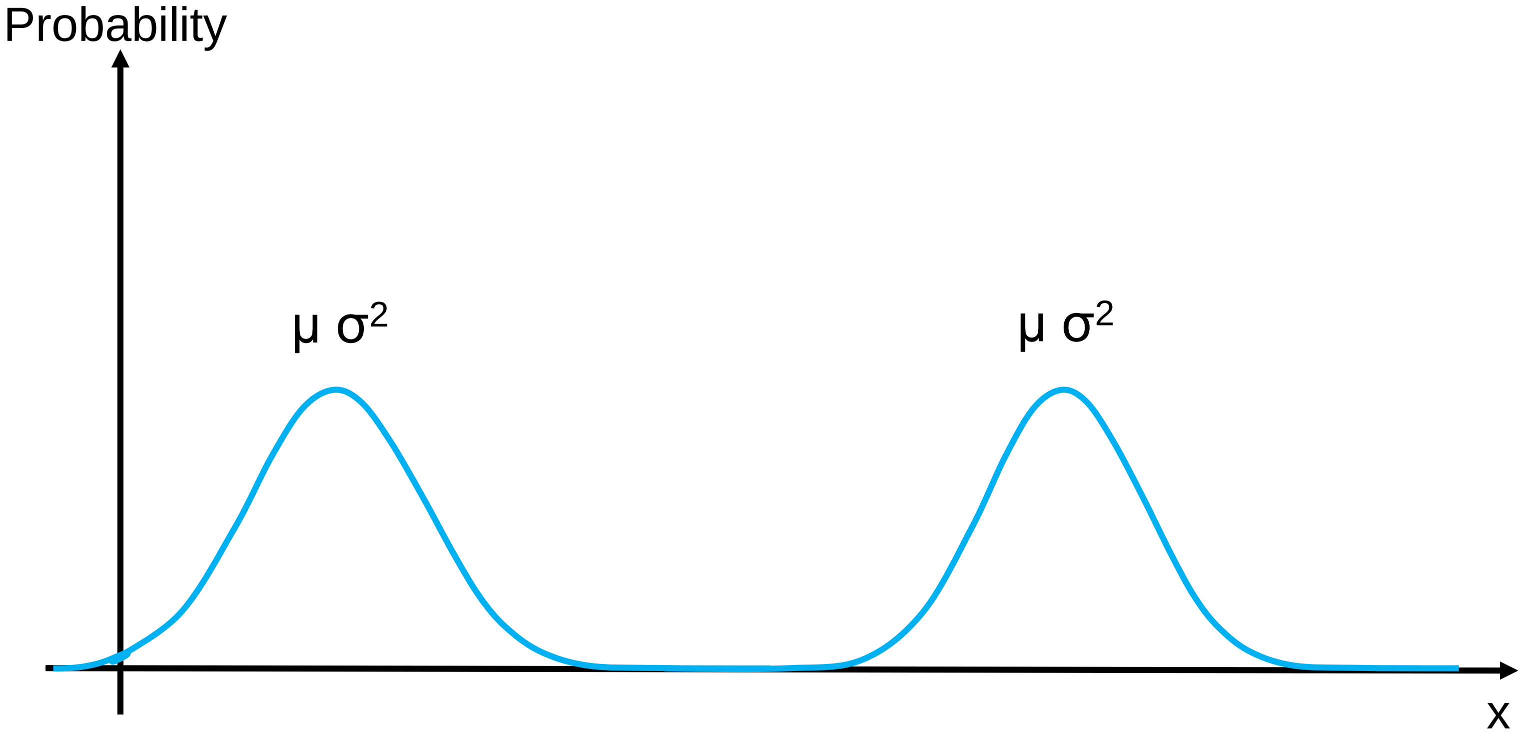
$$\mu = \frac{\frac{\mu_1}{2\sigma_1^2} + \frac{\mu_2}{2\sigma_2^1}}{\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}} = \frac{\mu_1\sigma_2^2 + \mu_2\sigma^2}{\sigma_2^2 + \sigma_1^2}$$

$$\sigma^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

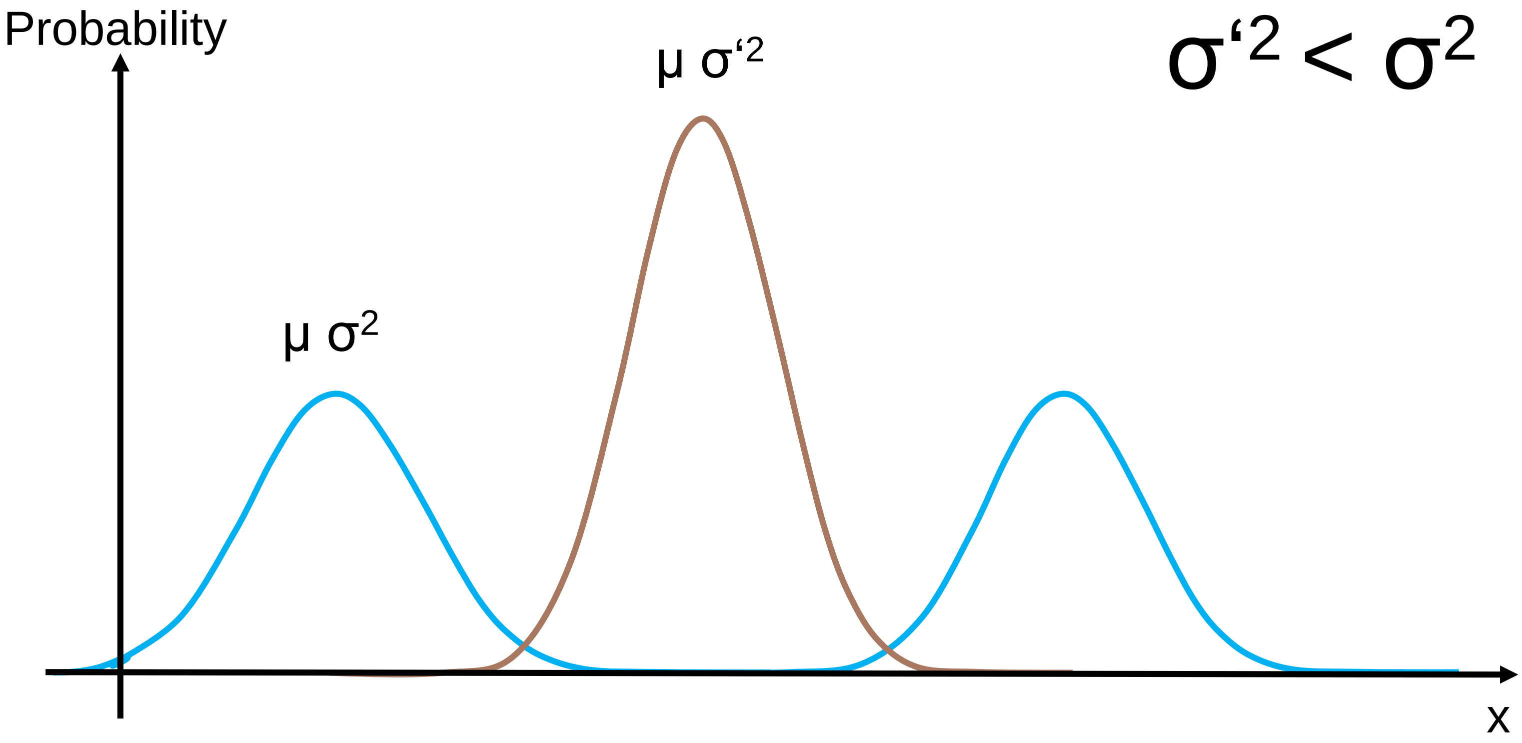
Multiplication of Gaussians



Multiplication of Gaussians



Multiplication of Gaussians



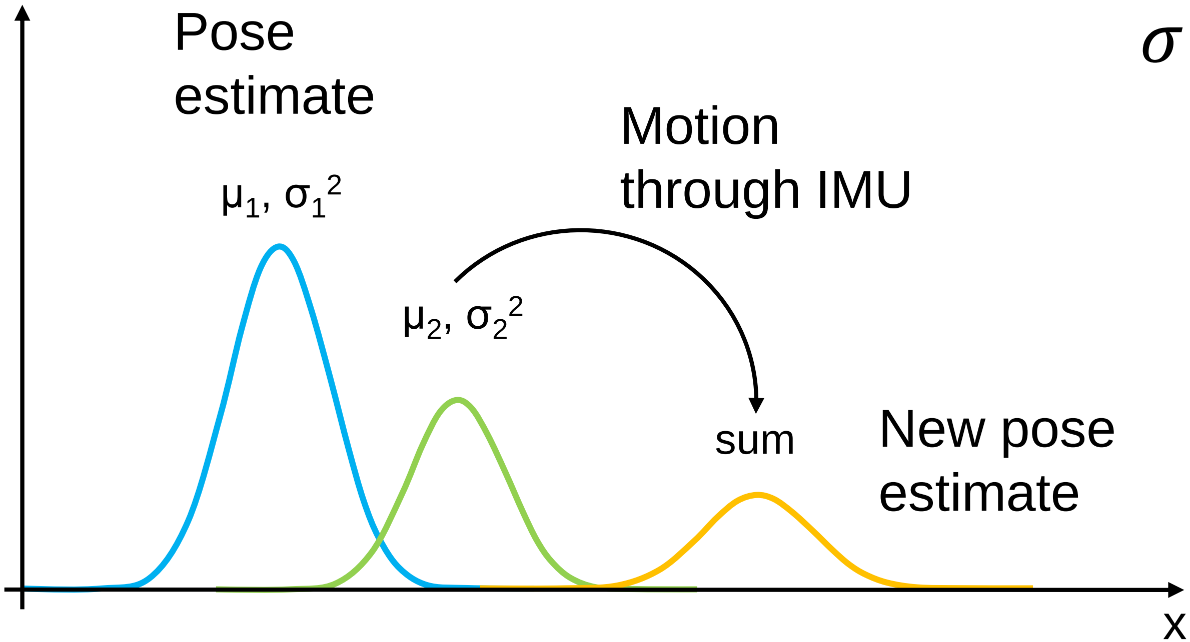
→ **Even low quality information improves state estimation**

Addition of Gaussians

- Addition of Gaussians

$$\mu = \mu_1 + \mu_2$$

$$\sigma = \sigma_1 + \sigma_2$$



→ **Probabilistic motion prediction increases uncertainty**

Localization & Mapping I
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

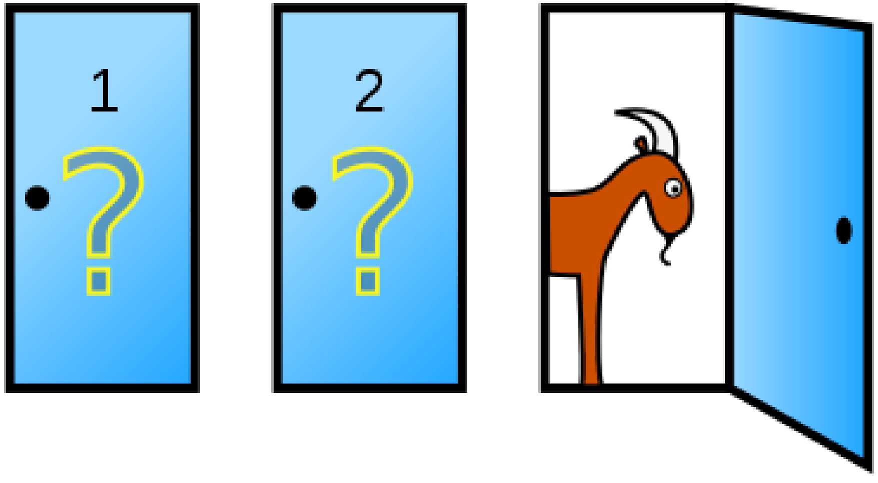
1. Motivation
2. Introduction to Probabilistics
3. **Bayesian Filtering**
4. Probabilistic Localization
5. Map representations
6. Summary



Bayesian Filter – Introduction

Conditional Probability

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$



Bayesian Filter – Introduction

Conditional Probability – Monty Hall Problem

“Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?”
(Parade magazine 1990)

Many people think, the chance would be 1/2 as there are two doors and one car. But the game master had more information and always opens up a door with a goat.
So the second decision which door to open is not independent of what happened before.
Let’s break it down. With the first decision, we have a 1/3 chance to directly choose the car and 2/3 to choose a goat. If we keep this differentiation through our second decision. We change and take the 2/3 from before and don’t let the game master unsettle us. 2/3 to win a car.
If you think this is easy, look it up in the internet. When this problem first occurred, many well renowned Mathematics professors were wrong.

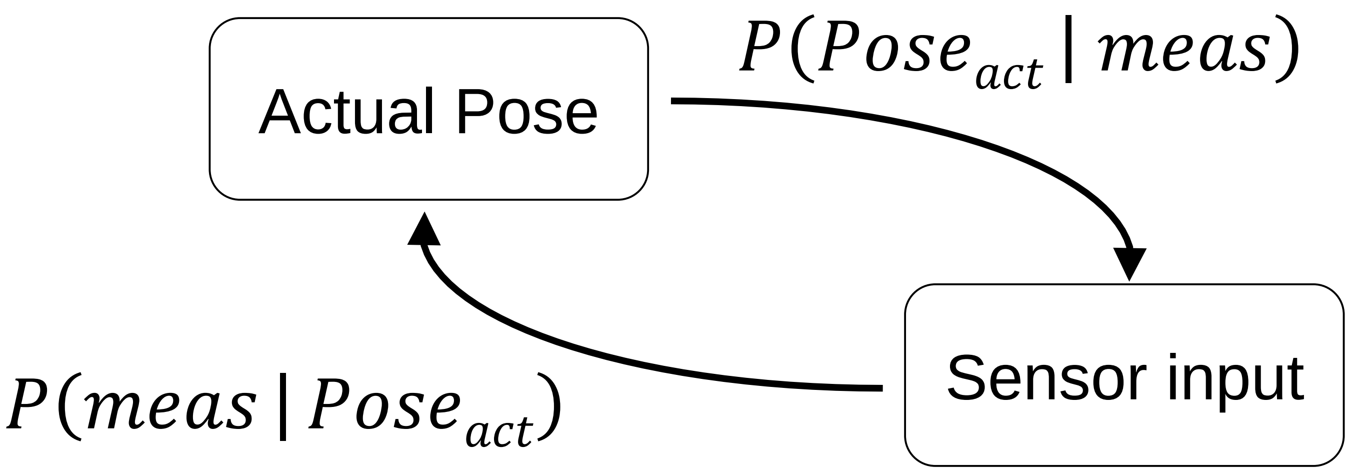
Bayesian Filter – Introduction

Bayes Rule

$$P(Pose_{act} \mid meas) = \frac{P(meas \mid Pose_{act}) \cdot P(Pose_{act})}{P(meas)}$$

Sensor information (e.g. datasheet, tests, etc.)
"How good is the sensor"

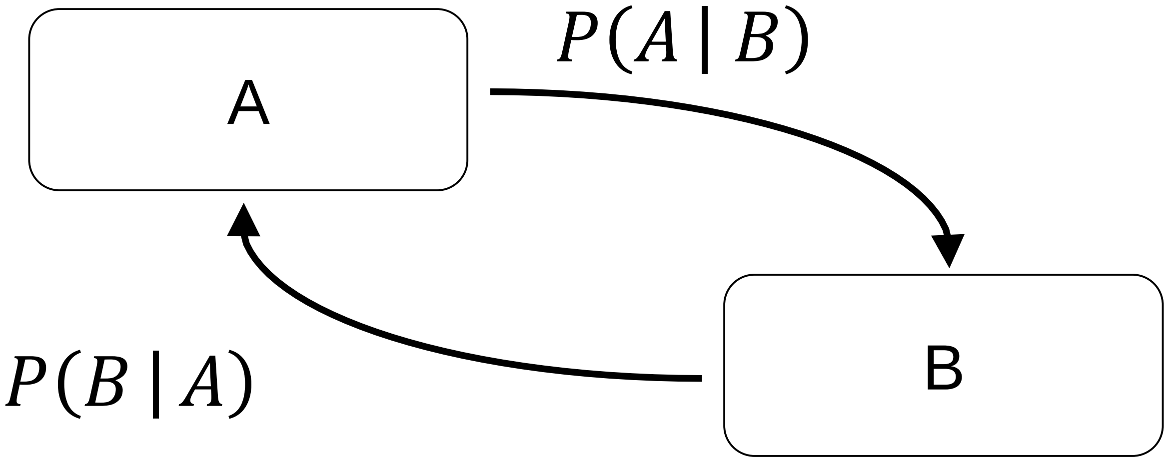
- **Noise**
- **Offset**
- **…**



Bayesian Filter – Introduction

Bayes Rule

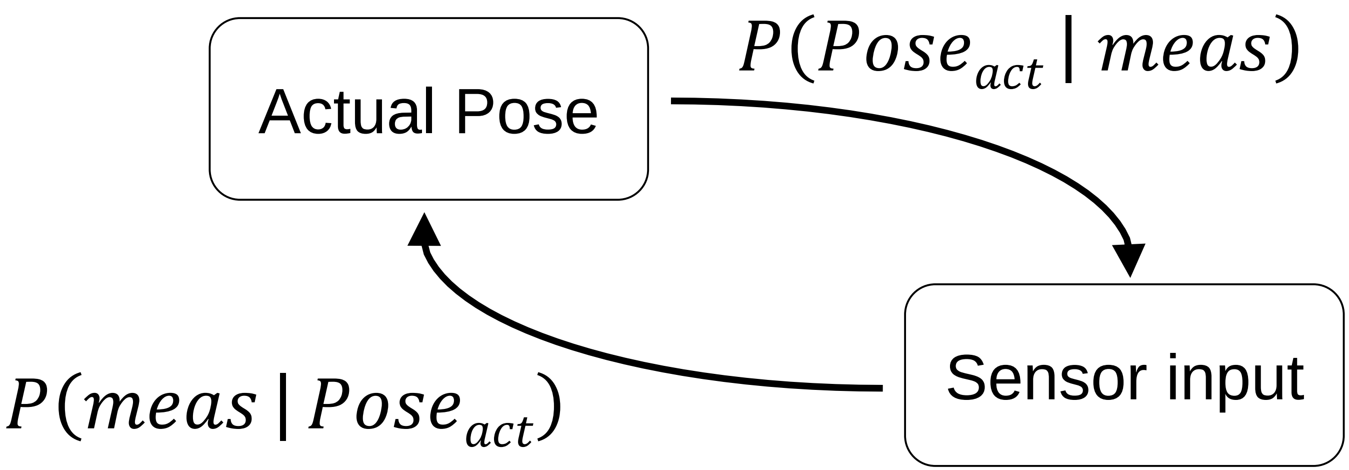
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$



Bayesian Filter – Introduction

Bayes Rule

$$P(Pose_{act} \mid meas) = \frac{P(meas \mid Pose_{act}) \cdot P(Pose_{act})}{P(meas)}$$



Bayesian Filter – Introduction

Given

- Observations z and motion data u for each timestep
- Sensor model $P(z \mid x)$
- Motion model $P(x \mid u, x')$
- Prior knowledge of state $P(x)$

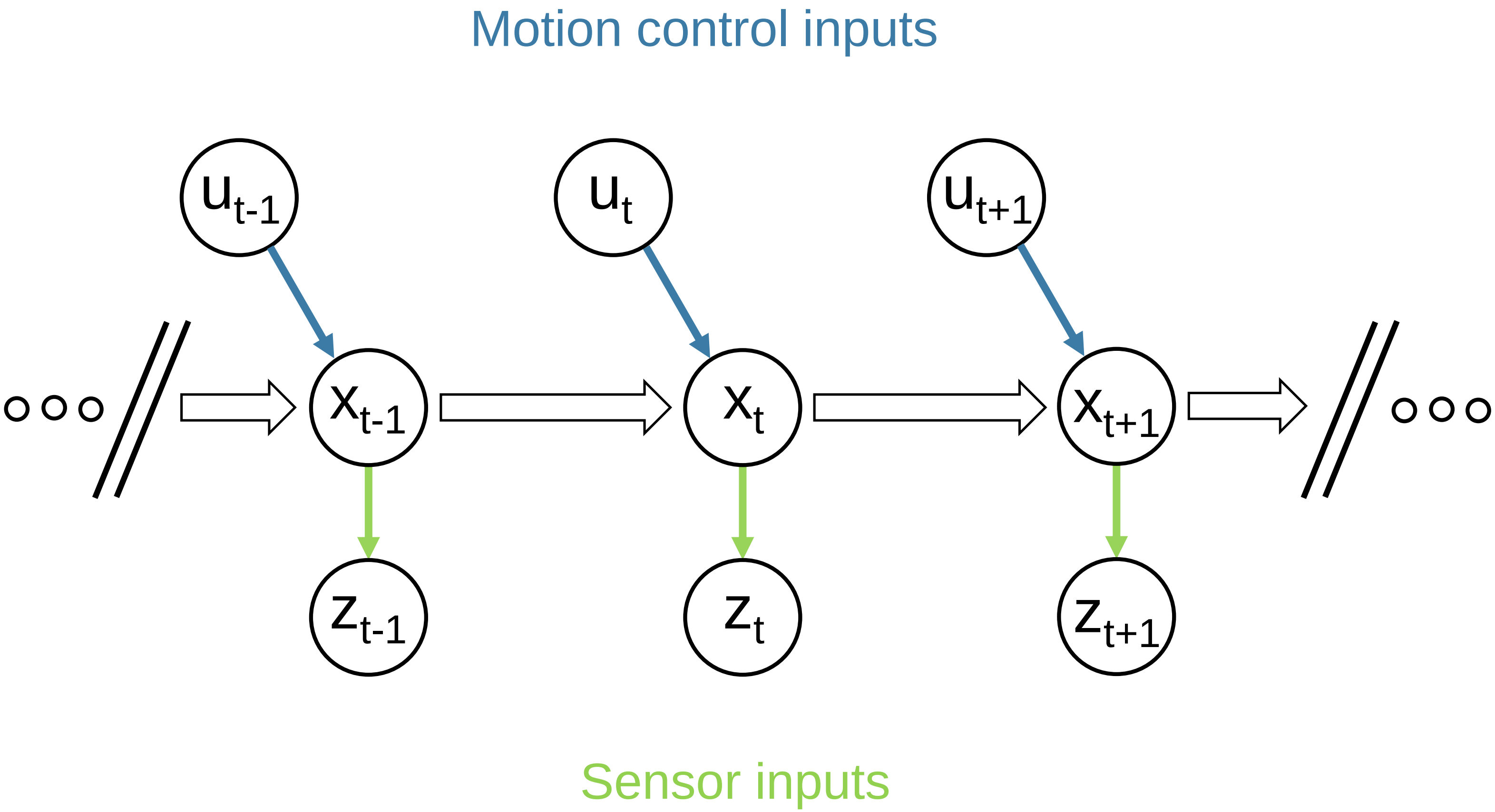
Wanted

- Estimate of the state X of a dynamic system → Not observable
- Posterior of state is called Belief $Bel(x_t) = P(x_t \mid u_1, z_1, \ldots, u_t, z_t)$

Markov Assumption

- Static environment
- Noise is independent
- No model errors

Bayesian Filter – Introduction

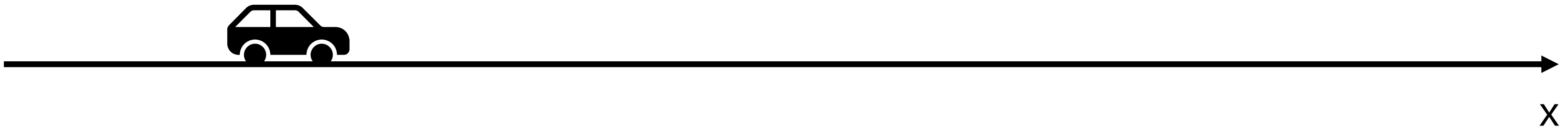


Kalman Filter – Cycle

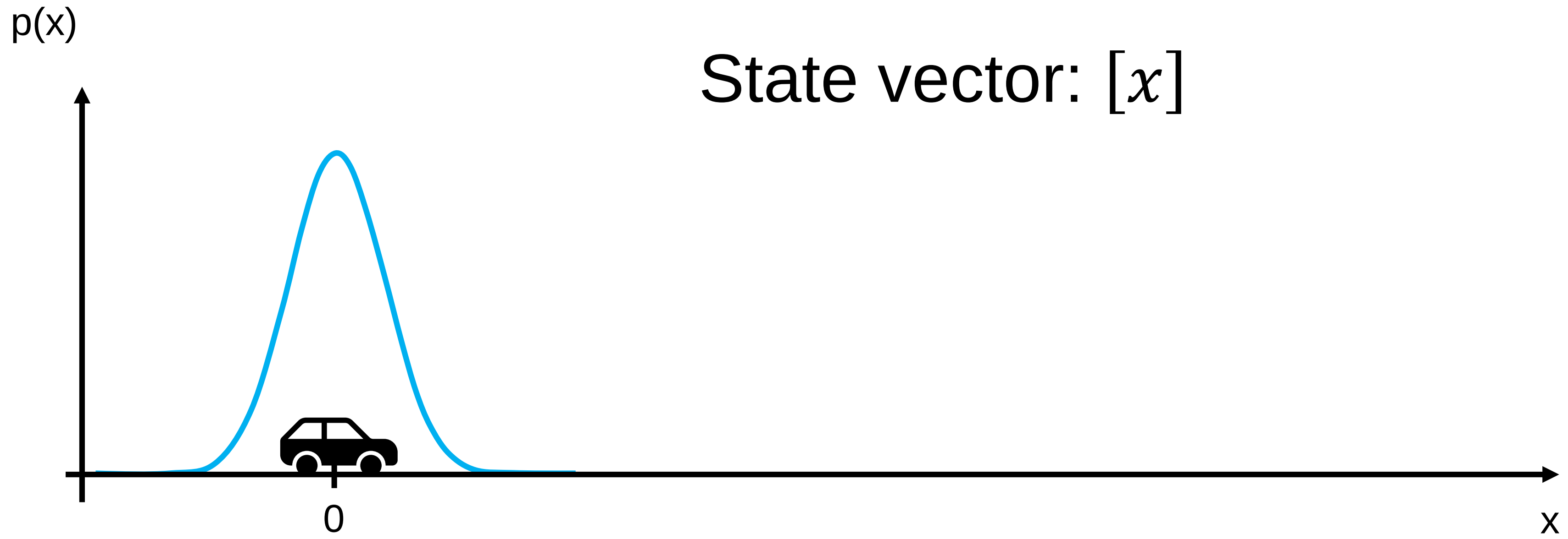
Vehicle at initial state

No information about pose

Initial guess needed → We can assume initial position x and velocity v as 0



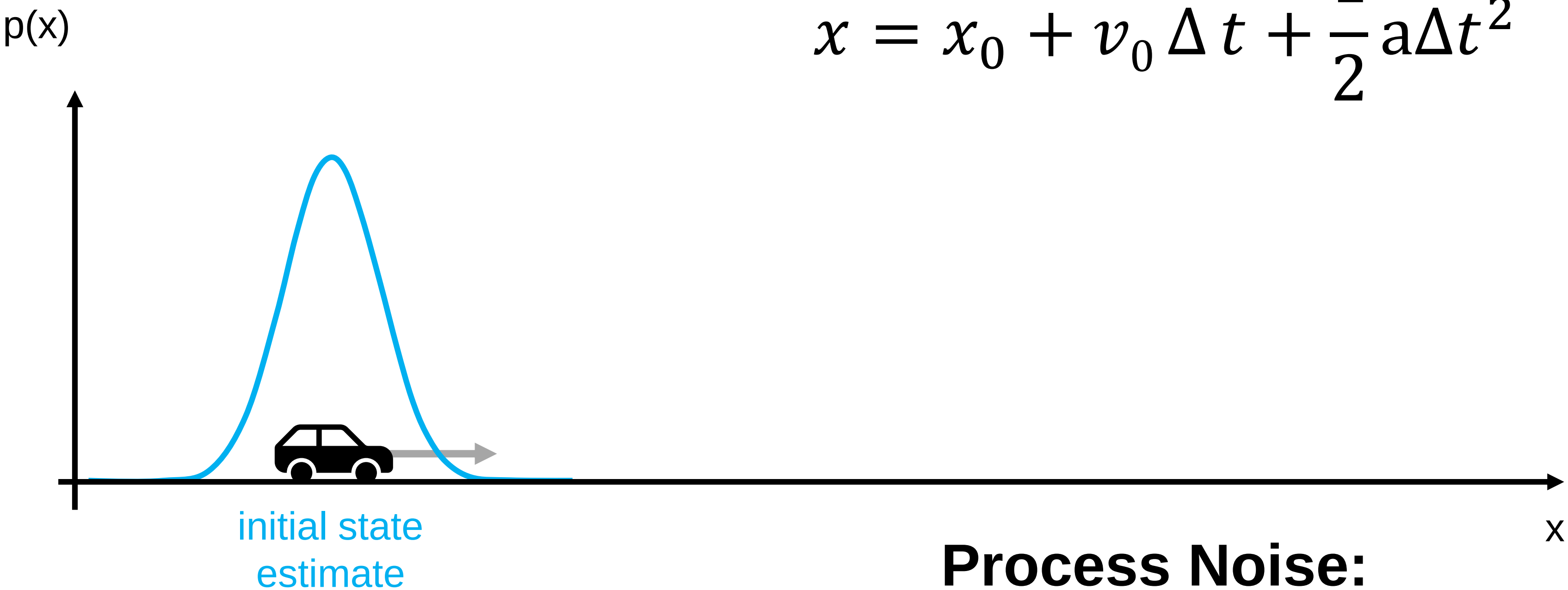
Kalman Filter – Initial State Estimate

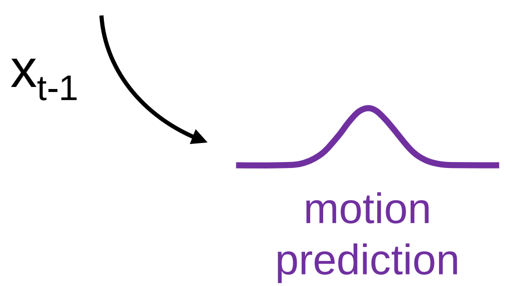


State vector: $[x]$

Kalman Filter – Motion Prediction

$$x = x_0 + v_0 \Delta t + \frac{1}{2} a \Delta t^2$$

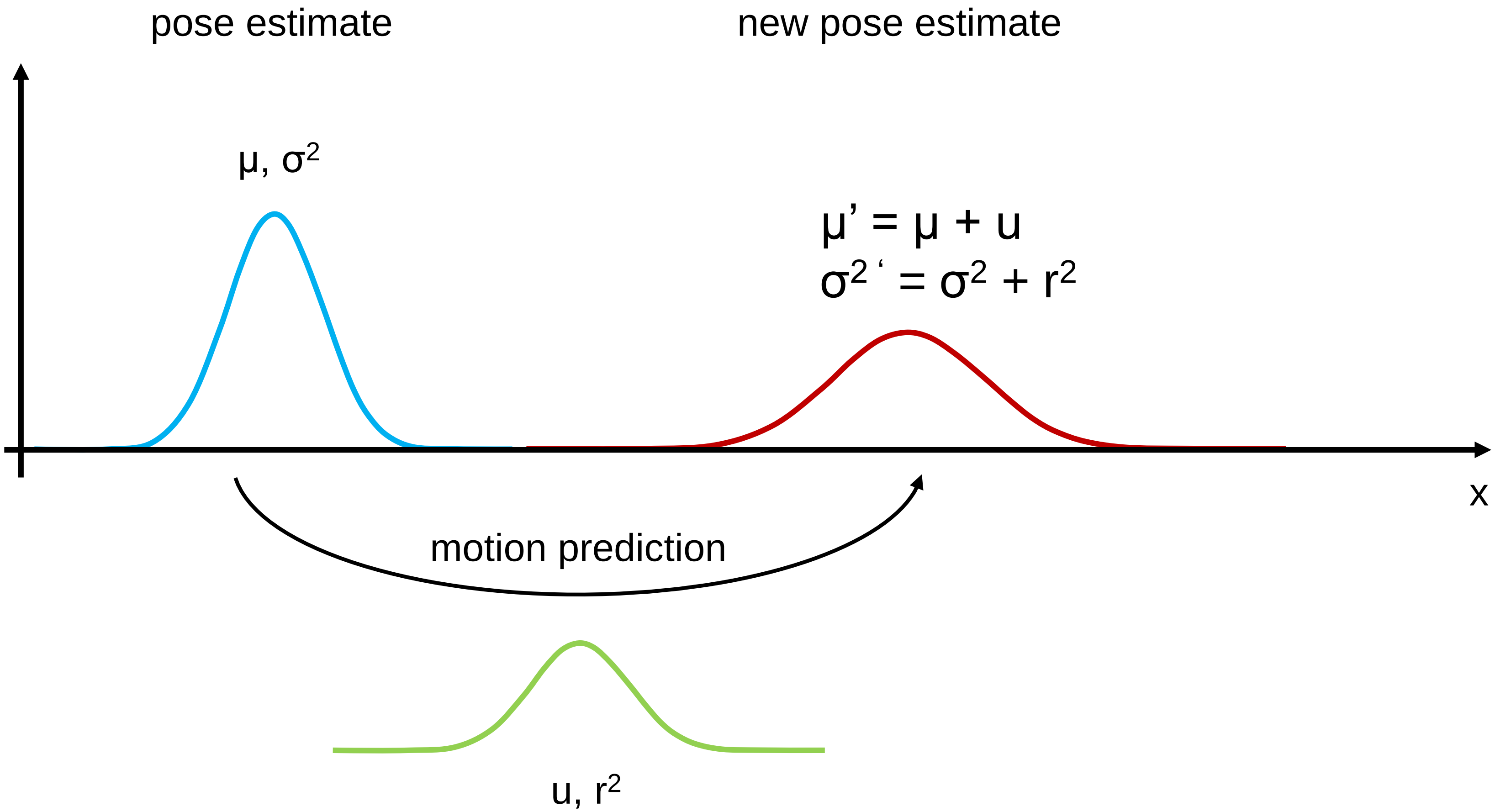




Process Noise:

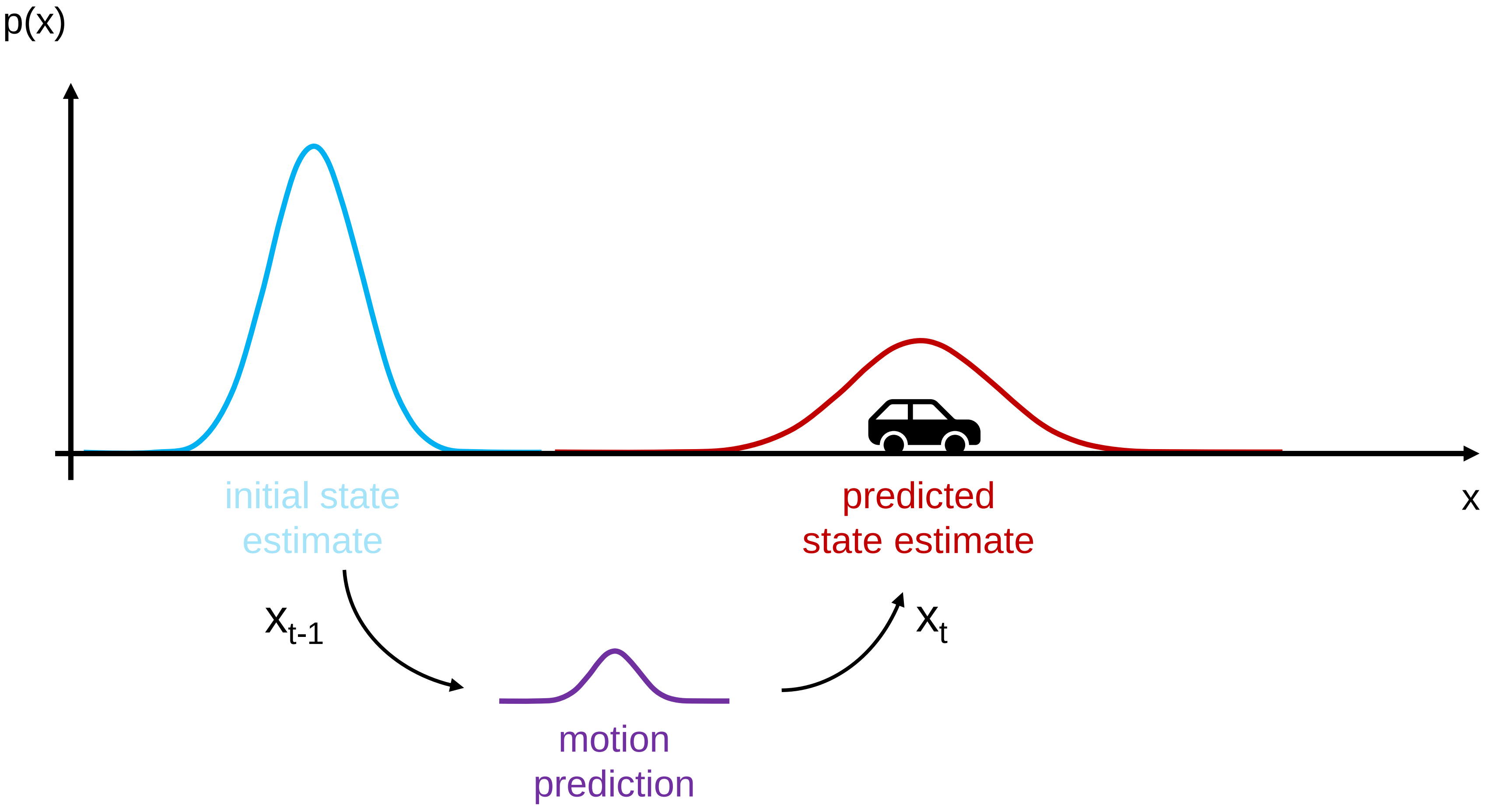
- Wind
- Street profile
- Etc.

Kalman Filter – Motion Prediction



Motion update increases uncertainty (σ^2)

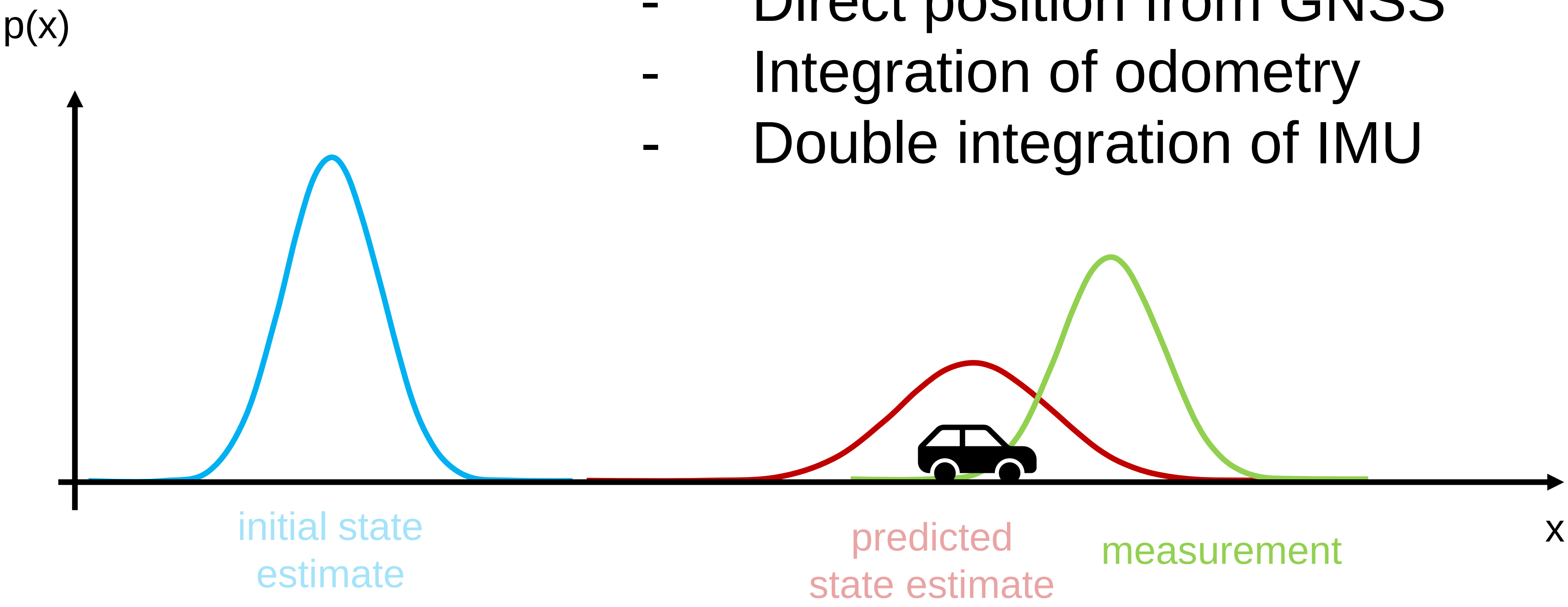
Kalman Filter – Motion Prediction



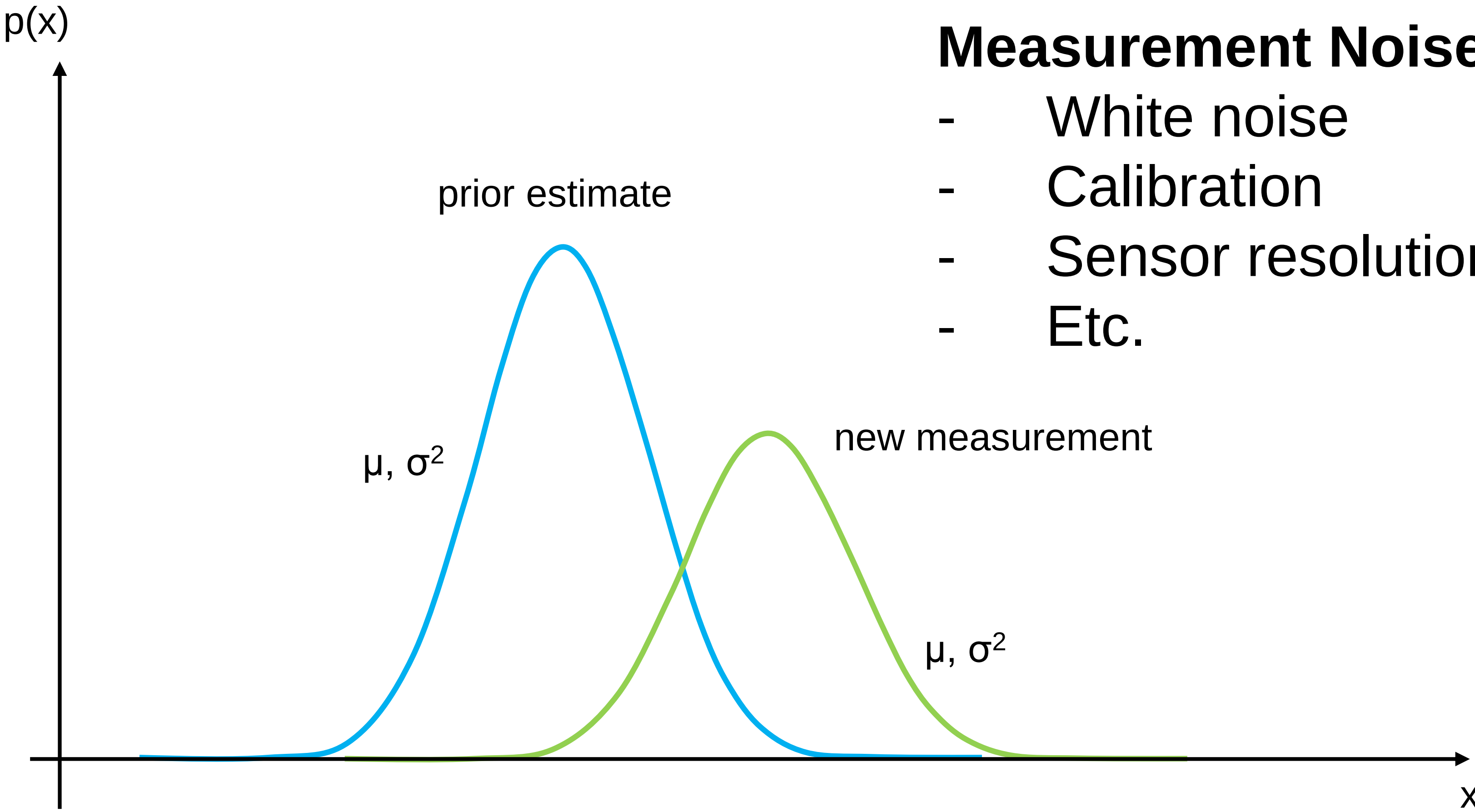
Kalman Filter – Measurement Update

Measurement model:

- Direct position from GNSS
- Integration of odometry
- Double integration of IMU



Kalman Filter – Measurement Update



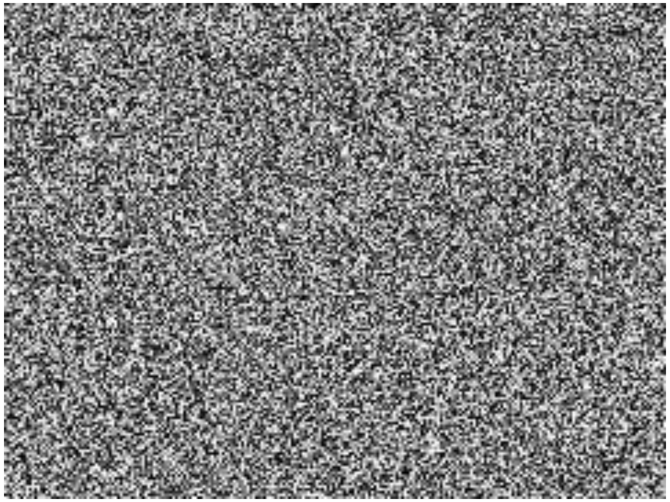
Measurement Noise:

- White noise
- Calibration
- Sensor resolution
- Etc.

Kalman Filter – Measurement Update

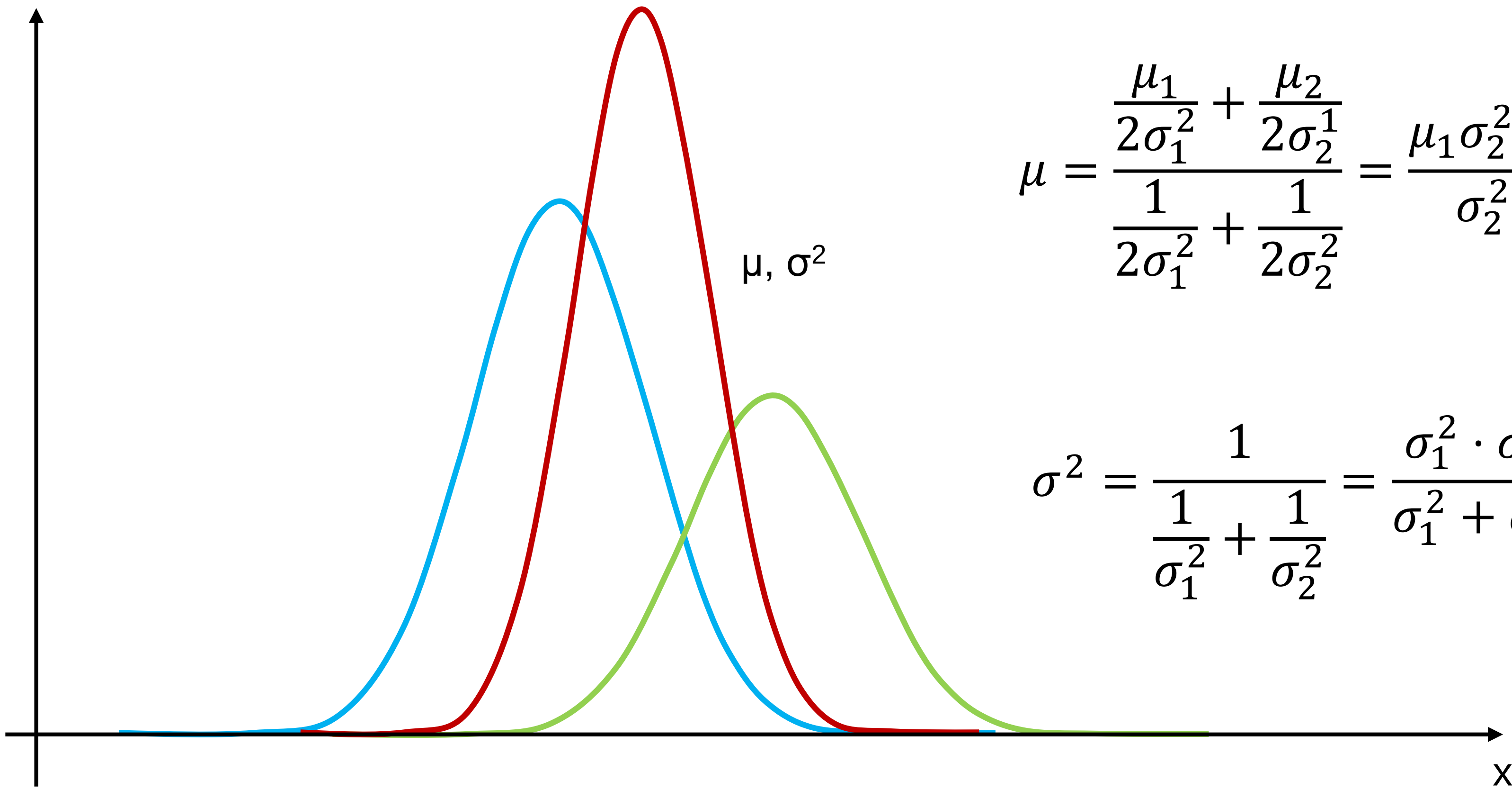
White Noise

White noise is a random signal with equal intensity at different frequencies. It is used as a statistical model for many kinds of signals.
Almost every signal you could think of has some kind of white noise.
Think of an old TV that did not have proper signal. What you saw then was just the white noise.



Kalman Filter – Measurement Update

Updated estimate: multiplication of Gaussians

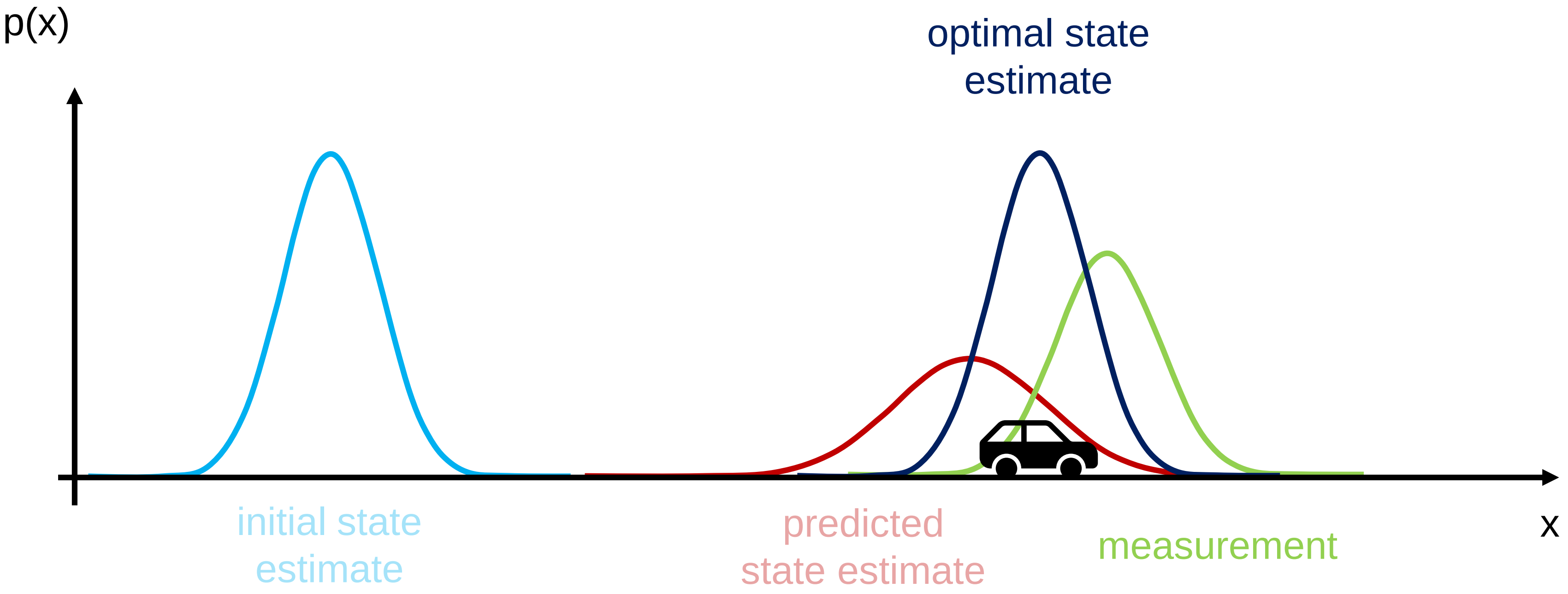


$$\mu = \frac{\dfrac{\mu_1}{2\sigma_1^2} + \dfrac{\mu_2}{2\sigma_2^1}}{\dfrac{1}{2\sigma_1^2} + \dfrac{1}{2\sigma_2^2}} = \frac{\mu_1\sigma_2^2 + \mu_2\sigma^2}{\sigma_2^2 + \sigma_1^2}$$

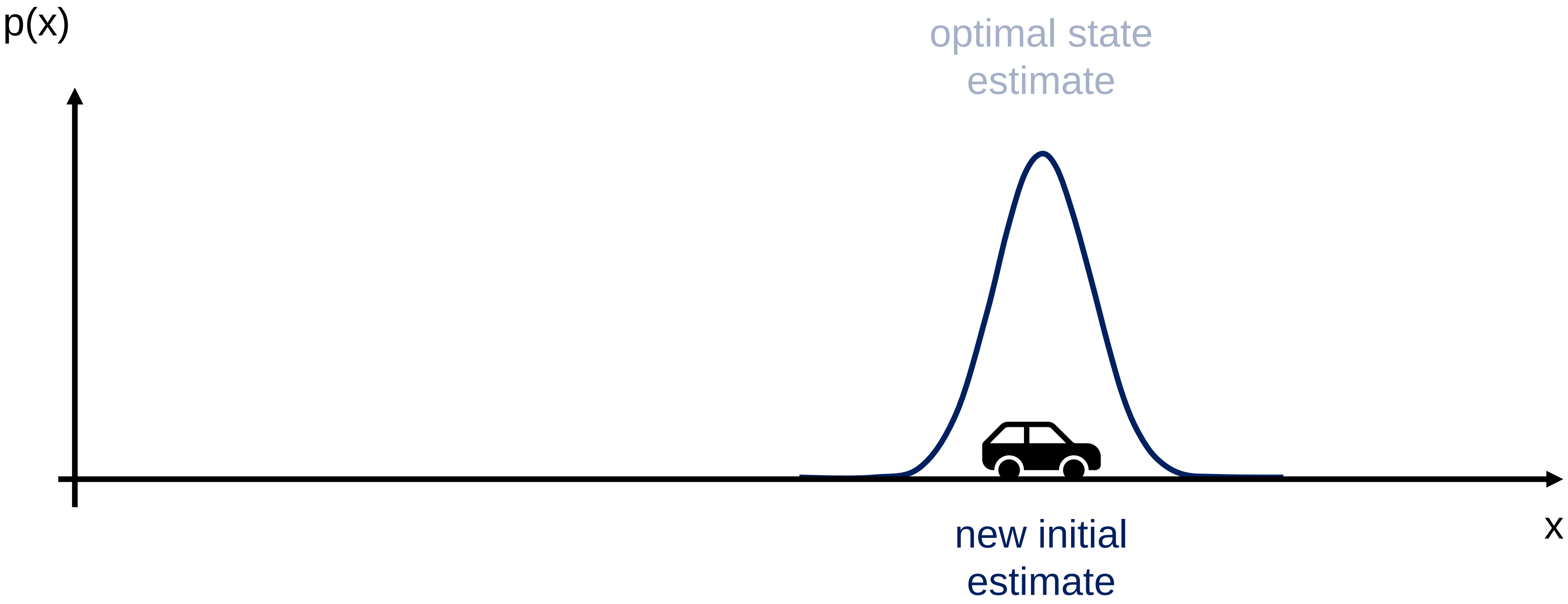
$$\sigma^2 = \frac{1}{\dfrac{1}{\sigma_1^2} + \dfrac{1}{\sigma_2^2}} = \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Measurement update decreases uncertainty (σ^2)

Kalman Filter – Optimal State Estimate



Kalman Filter – Optimal State Estimate

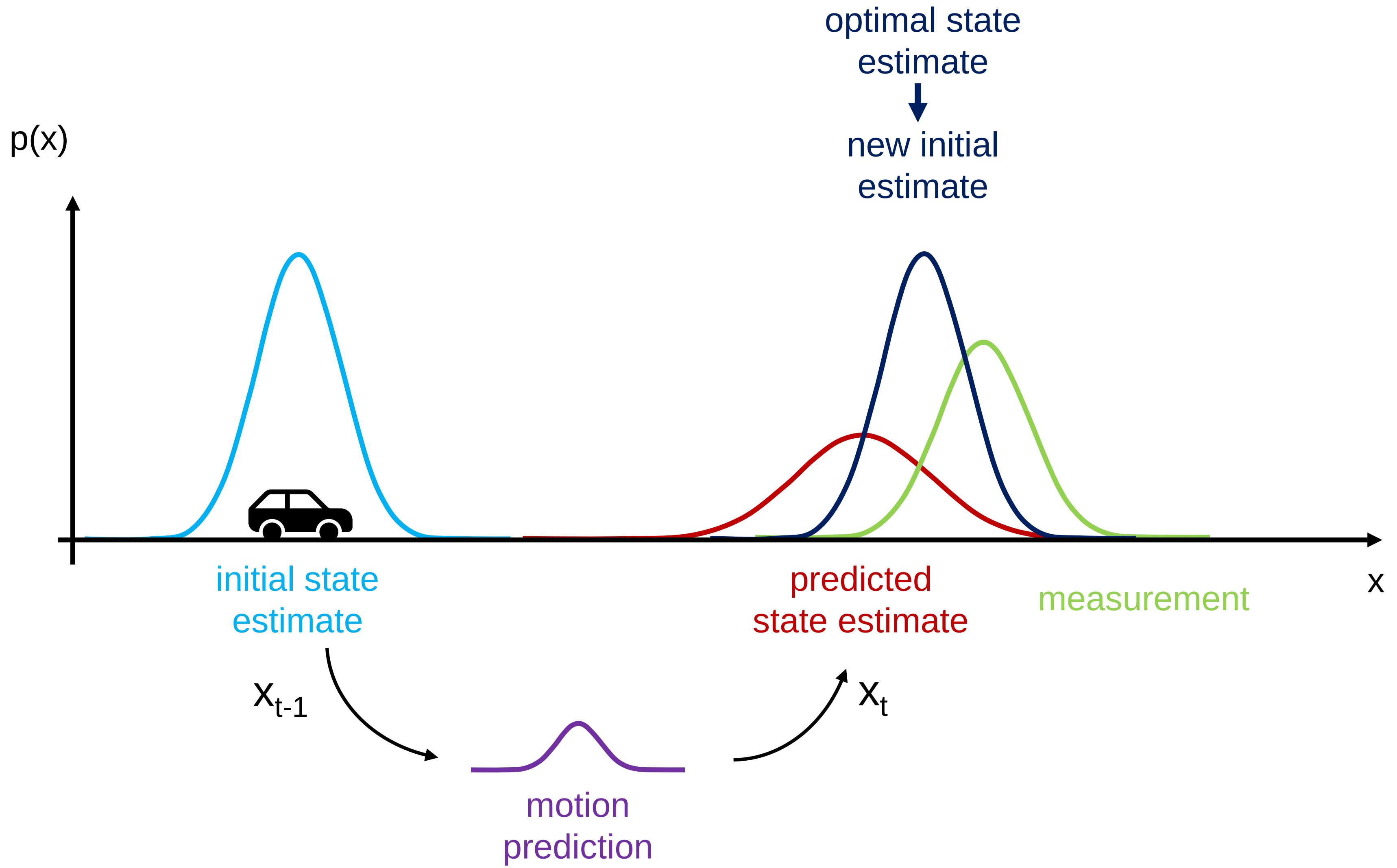


Kalman Filter – Optimal State Estimate

State Estimation

State estimation has many applications in the field of robotics and autonomous driving. Since many states of a vehicle cannot be measured directly but might be important for the control of the vehicle (e.g. jerk, yaw angle, etc.) state estimators are used.
State estimators basically enable the estimation of non-observable states.
The Kalman filter is the optimal state estimator.

Kalman Filter – Cycle



Kalman Filter – Motion Models

Point mass model

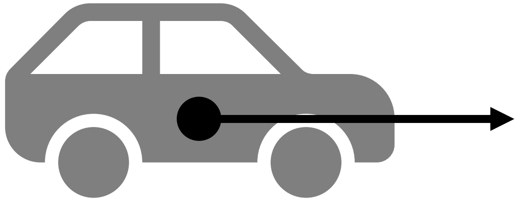
Constant velocity

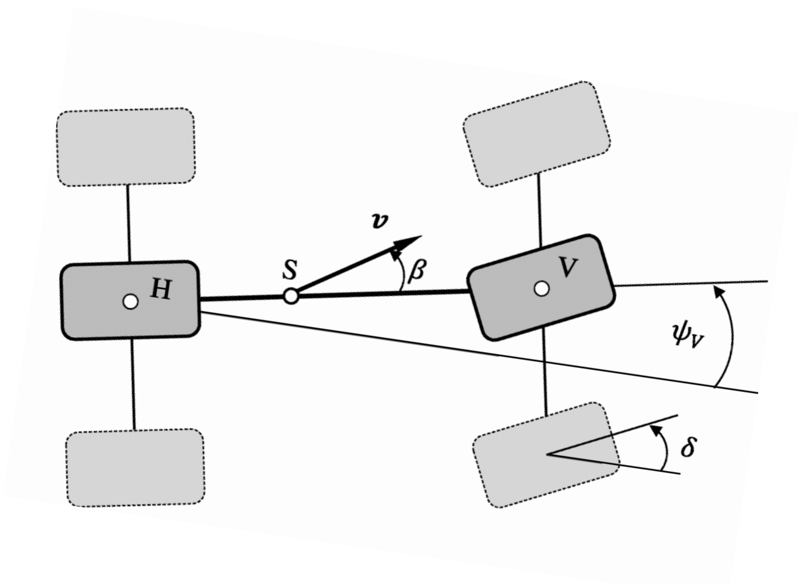
Constant acceleration

Constant turn rate

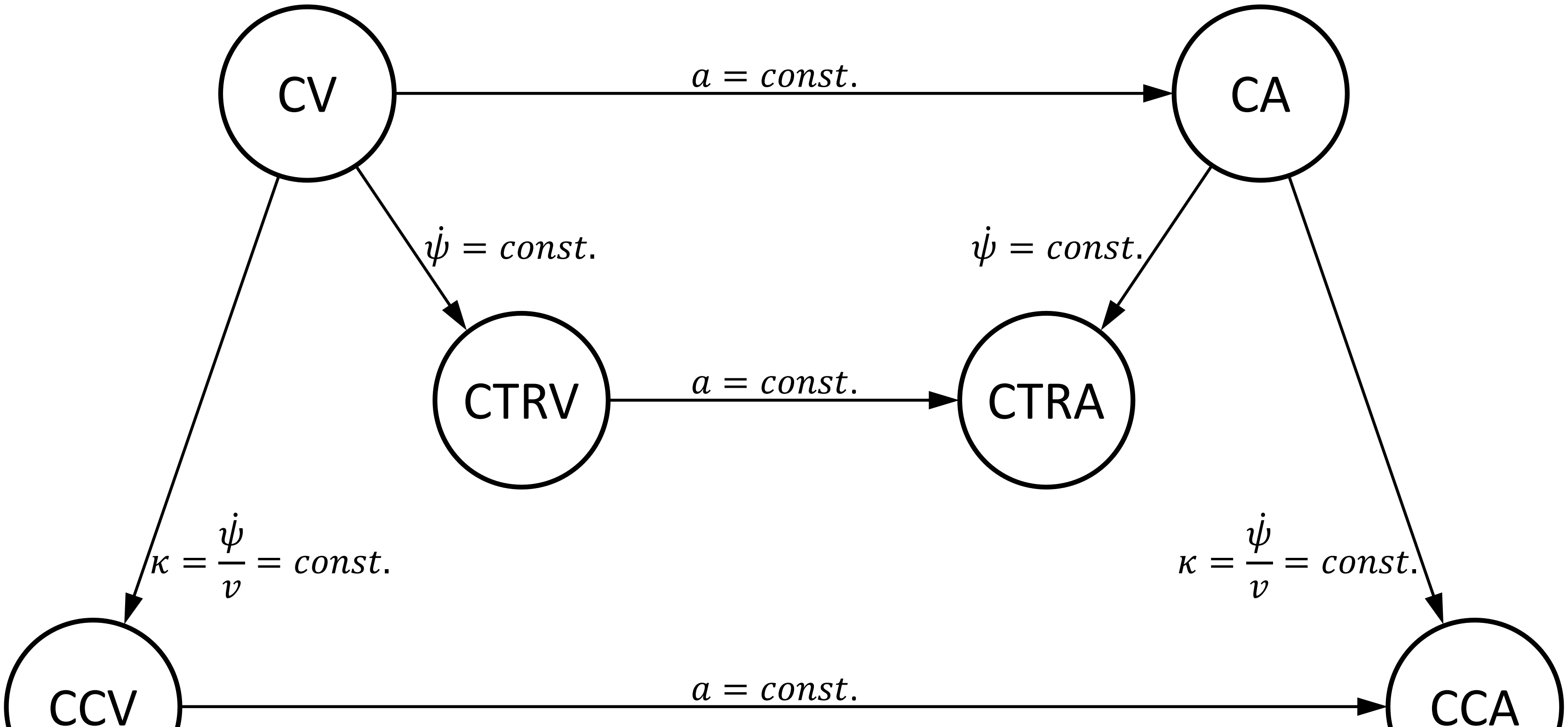
One-track model

Two-track model





State Estimation: Kinematic Models – Overview



CV	Constant Velocity	CTR	Constant Turn Rate
CC	Constant Curvature	CA	Constant Acceleration
$\dot{\psi}$	Turn Rate	κ	Curvature
a	Acceleration	v	Velocity

R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," *2008 11th International Conference on Information Fusion*, pp. 1–6, 2008.

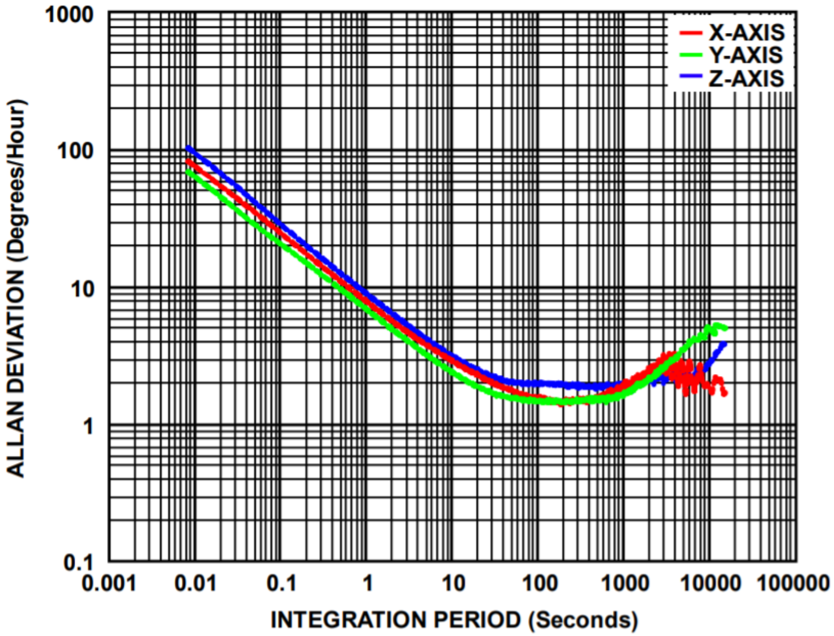
Kalman Filter – Measurement Models

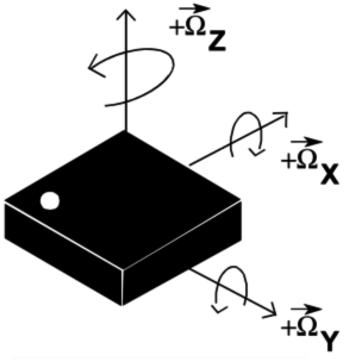
Direct position from GNSS

Integration of odometry

Double integration of IMU

Variances depending on sensor models





Kalman Filter – Implementation

- **Measurement Update**
  - Bayes Rule → Multiplication

- **Motion Update / Prediction**
  - Total Probability → Addition

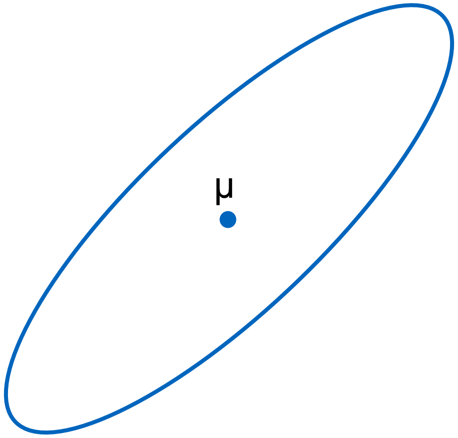
```
def update(mean1, var1, mean2, var2):
    new_mean = (var2 * mean1 + var1 * mean2) / (var1 + var2)
    new_var = 1/(1/var1 + 1/var2)
    return [new_mean, new_var]

def predict(mean1, var1, mean2, var2):
    new_mean = mean1 + mean2
    new_var = var1 + var2
    return [new_mean, new_var]
```

Kalman Filter – Implementation

What if we are not considering a one dimensional space?

- Matrix Representation
- Variance $\sigma^2 \rightarrow$ Covariance Σ
- Covariance Matrix Q



Kalman Filter

Prediction

Project the state ahead

$$x_{k+1} = Ax_k + Bu_k$$

Project the error covariance ahead

$$P_{k+1} = AP_kA^T + Q$$

Correction

Compute the Kalman Gain

$$K_k = P_kH^T(HP_kH^T + R)^{-1}$$

Update the estimate via measurement

$$x_k = x_k + K_k(z_k - Hx_k)$$

Update the error covariance

$$\mathrm{P}_k = (I - K_kH)P_k$$

Initialize R, P, Q once

Kalman Filter

Kalman Gain

The Kalman gain is the weighting of the different sources of information (dynamic model, motion model). It automatically adjusts according to the residuals (the deviations of the single sources of information).
This means that the Kalman filter “gets to know the system”. And has more or less “trust” in different sensors or dynamic models.

When a sensor fails, the residuals will become very big and thus the filter will use this sensor less over time.

Kalman Filter

$$x_t = A_t x_{t-1} + B u_t + \varepsilon_t$$ prediction

$$z_t = C_t x_t + \delta_t$$ correction

linear

noises

x_t … Non observable state

z_t … Observable measured state

A_t … (n x n) how state evolves without control or noise

B_t … (n x l) how control u_t changes the state → **Motion model**

C_t … (k x n) how to map state x to observation z → **Sensor model**

ε_t, δ_t… random variables → process and measurement noise
(covariance Q_t and R_t)

Kalman Filter

A … State transition

B … Control input model

H … Observation

Q … Process Covariance

R … Observation Covariance

Extended Kalman Filter

Prediction

Project the state ahead

$$x_{k+1} = g(x_k, u)$$

Project the error covariance ahead

$$P_{k+1} = J_A P_k J_A^T + Q$$

Correction

Compute the Kalman Gain

$$K_k = P_k J_H^T (J_H P_k J_H^T + R)^{-1}$$

Update the estimate via measurement

$$x_k = x_k + K_k (z_k - h(x_k))$$

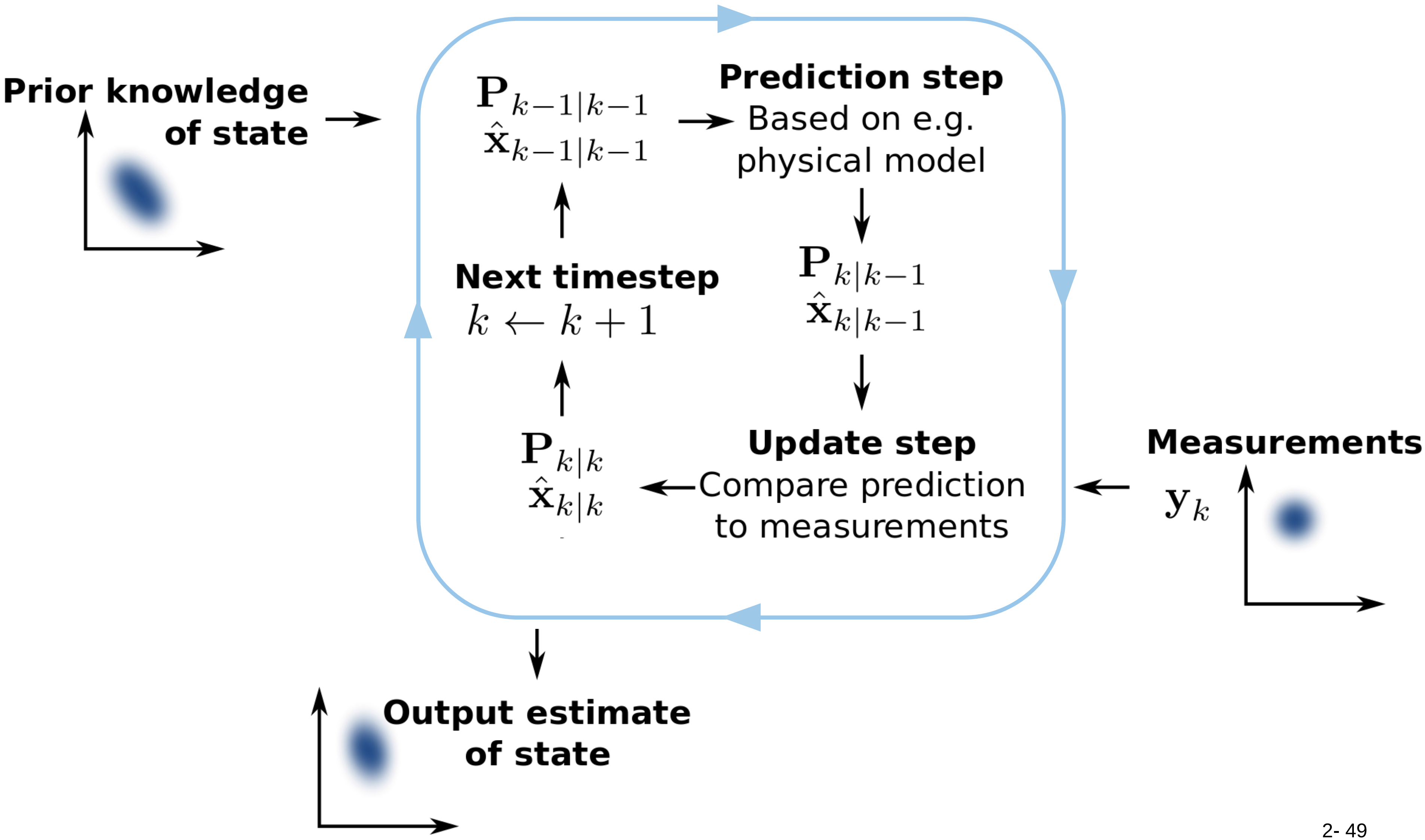
Update the error covariance

$$\mathrm{P}_k = (I - K_k J_H) P_k$$

Initialize P, R, Q, I once

J are the Jacobians

Kalman Filter



Kalman Filter – Outlook

Nonlinear models

- Extended Kalman Filter (EKF) → Linearization
- Unscented Kalman Filter (UKF) → Deterministic sampling of points

Non-Gaussian Distributions

- Particle Filter (PF)

More complex models

- Ackerman odometry estimation
- One-track-model / two-track-model

Kalman Filter – Outlook

State Estimator	Model	Assumed Distribution	Computational Cost
Kalman Filter	Linear	Gaussian	Low
Extended Kalman Filter	Locally linear	Gaussian	Low – Medium (depending on Jacobians)
Unscented Kalman Filter	Nonlinear	Gaussian	Medium
Particle Filter	Nonlilnear	Non-Gaussian	High

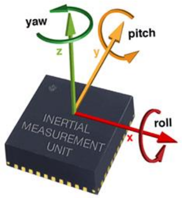
Kalman Filter - Additional Resources

http://ais.informatik.uni-freiburg.de/teaching/ws17/mapping/pdf/slam04-ekf.pdf

http://ais.informatik.uni-freiburg.de/teaching/ws17/mapping/pdf/slam06-ukf.pdf

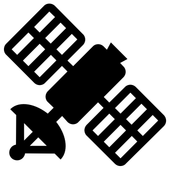
http://ais.informatik.uni-freiburg.de/teaching/ws17/mapping/pdf/slam07-eif.pdf

Kalman Filter - Example



IMU

Wheel
odometry



GNSS

Relative position

High frequency

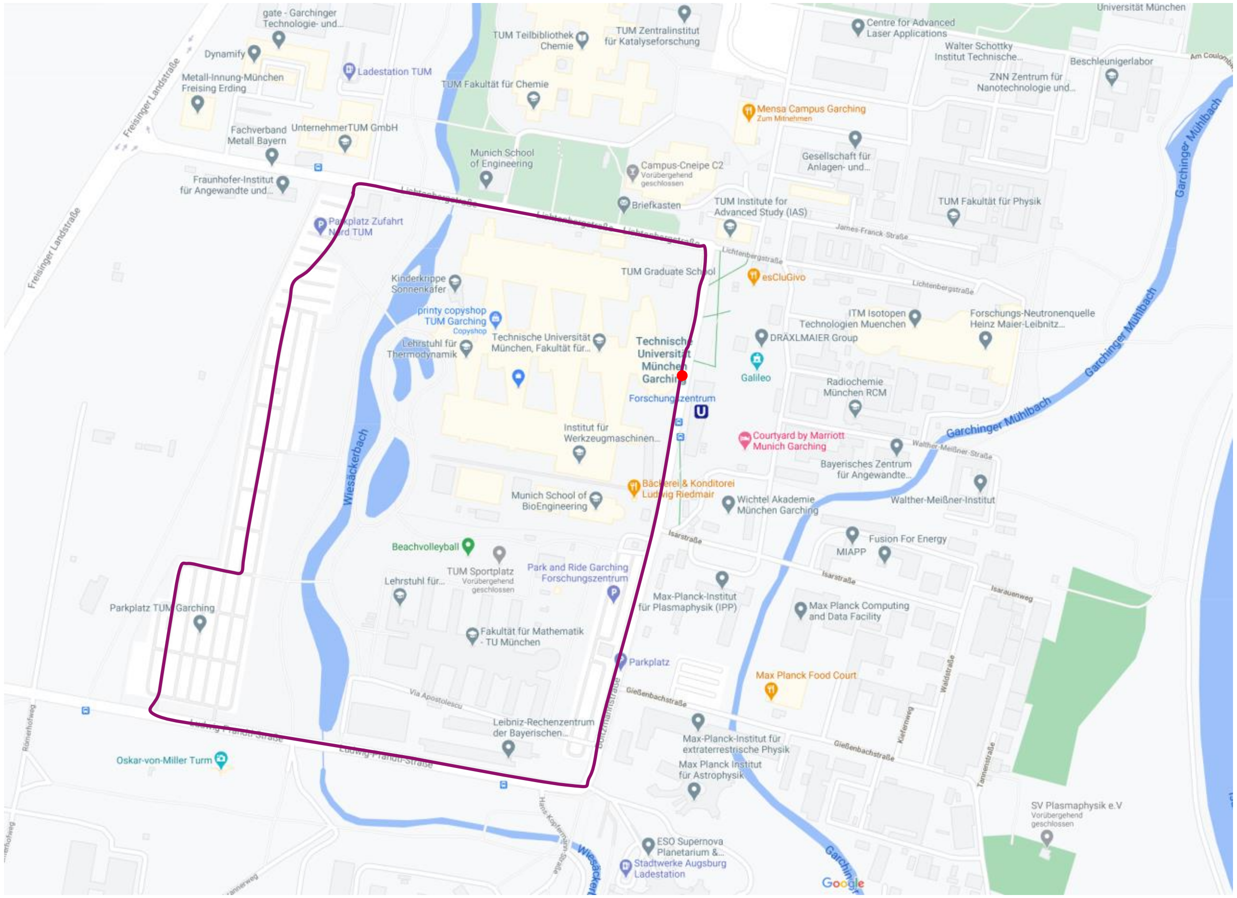
Error wind-up

Global position

Low frequency

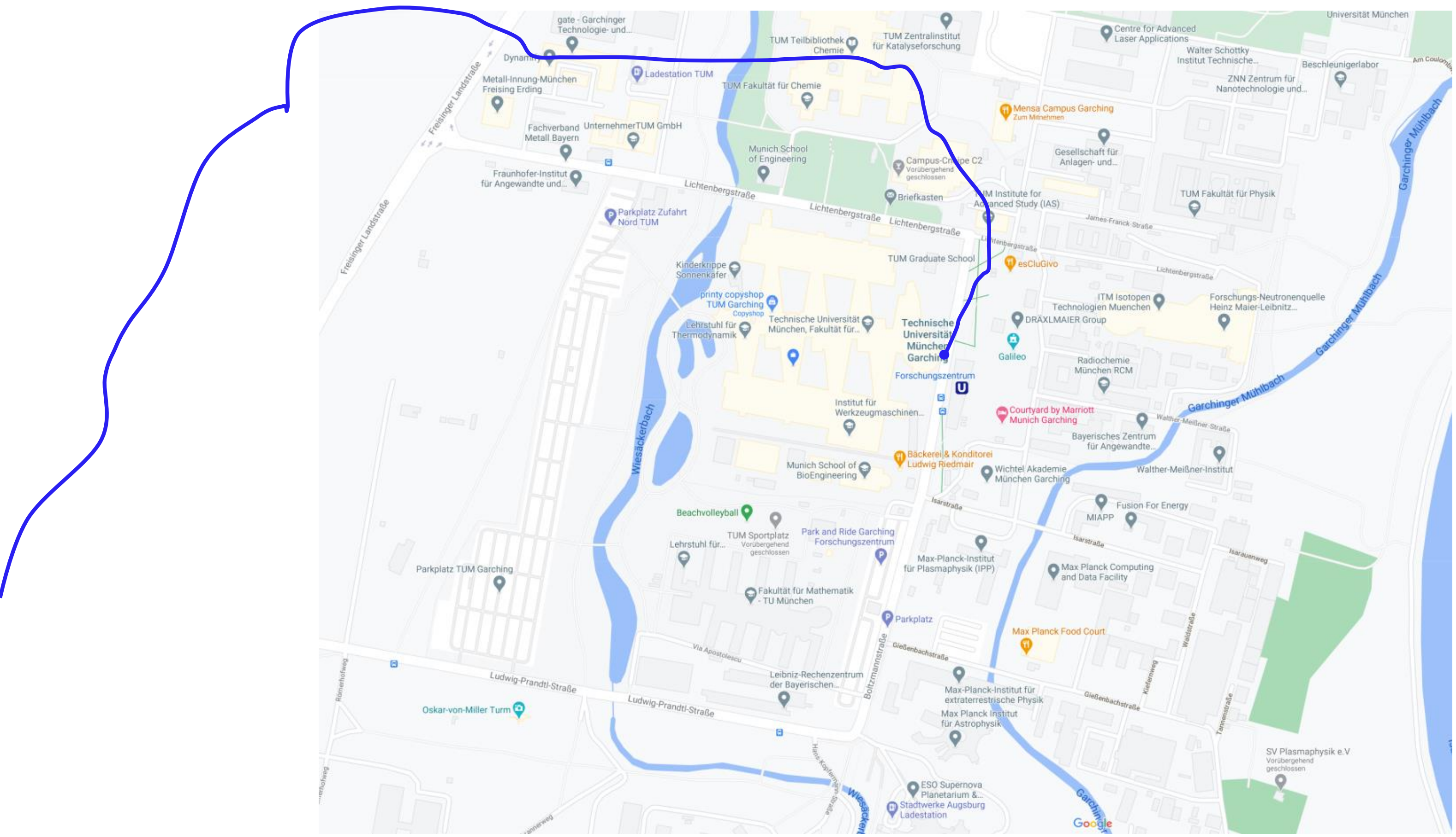
noisy

Kalman Filter Results – Ground Truth



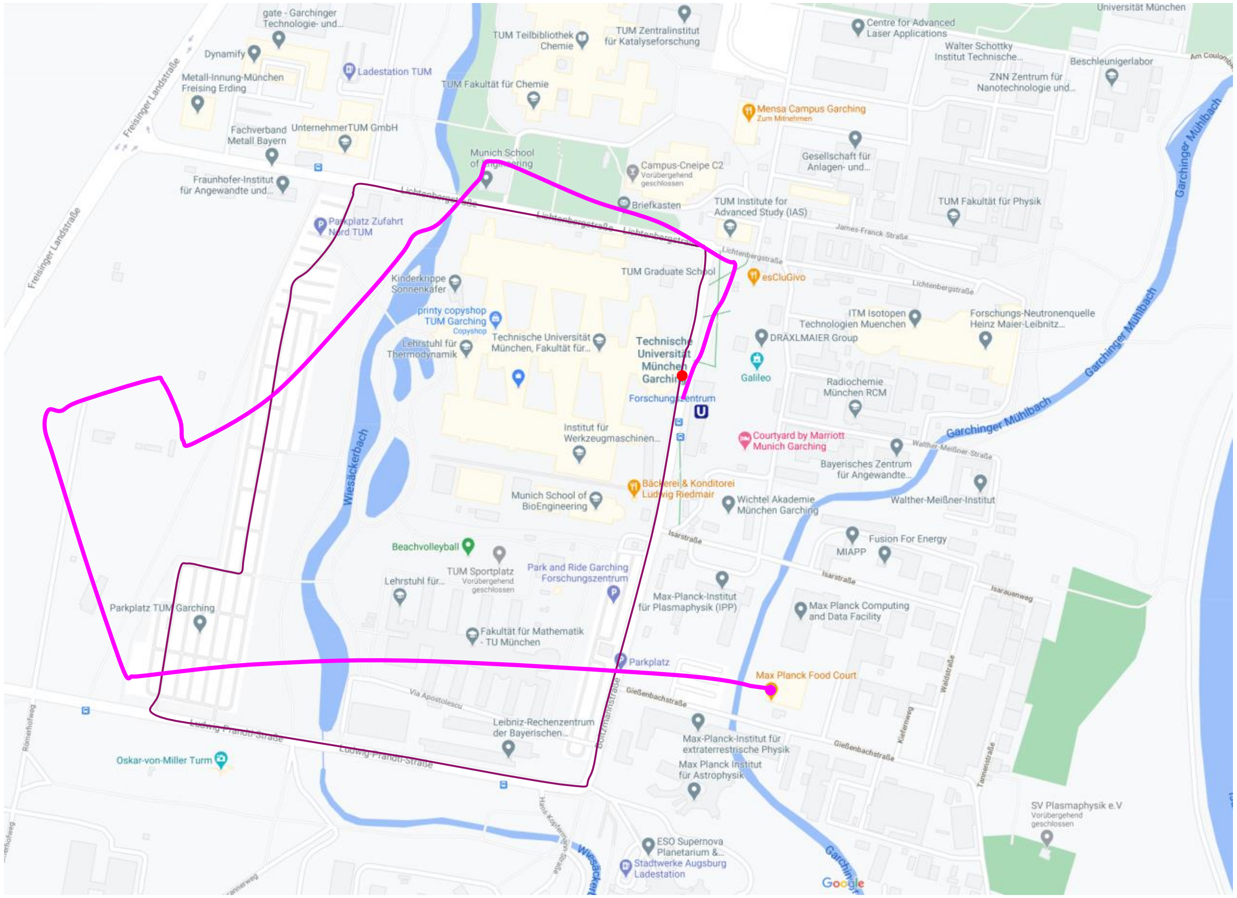
—— Ground truth

Kalman Filter Results



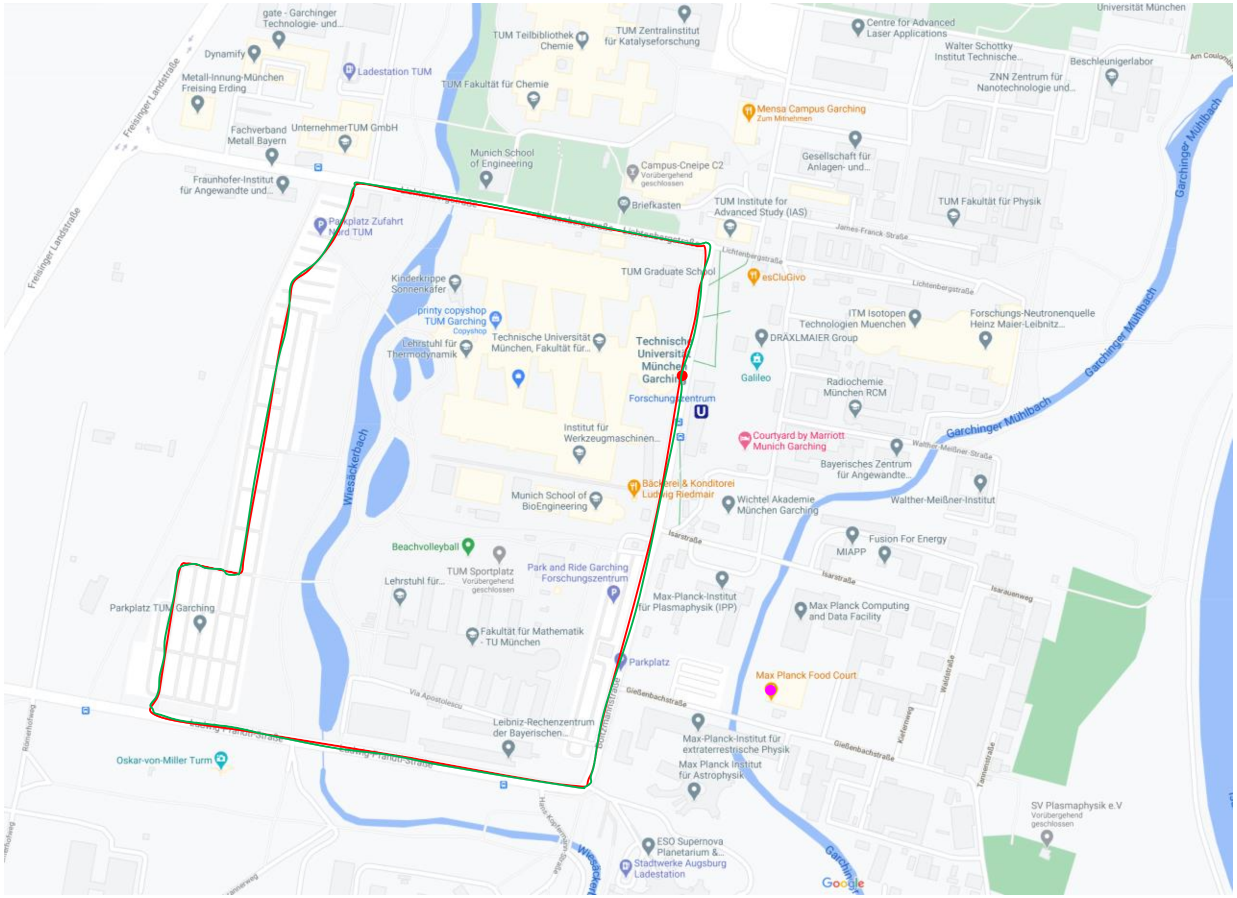
IMU integration

Kalman Filter Results



Kalman estimation: IMU + Wheel Odometry, no GNSS

Kalman Filter Results



Kalman estimation: IMU + Wheel Odometry + GNSS

Kalman Filter Results

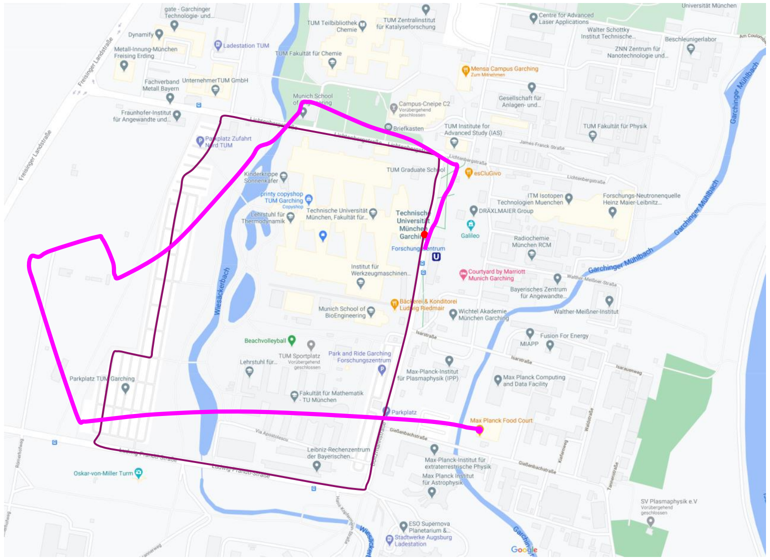
Huge improvement compared to single IMU

Global reference necessary

Errors wind up over time

Position estimate drifts away from ground truth

Quality heavily depends on tuning of KF



⇨ **Relation to environment needed for sufficient localization for AD Level 5: no relying on GNSS**

Why GPS Fails?

GPS Navigation can fail due to multiple reasons like environmental factors, such as clouds, dense forests and tall buildings. The quality of the receiver in your smartphone can also block the signal transmissions between GPS satellites and receivers.

⇨ **Relation to environment needed for sufficient localization for AD Level 5: no relying on GNSS**

https://www.gossipdiaries.com/why-and-how-gps-fails-many-times/

Localization & Mapping I
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

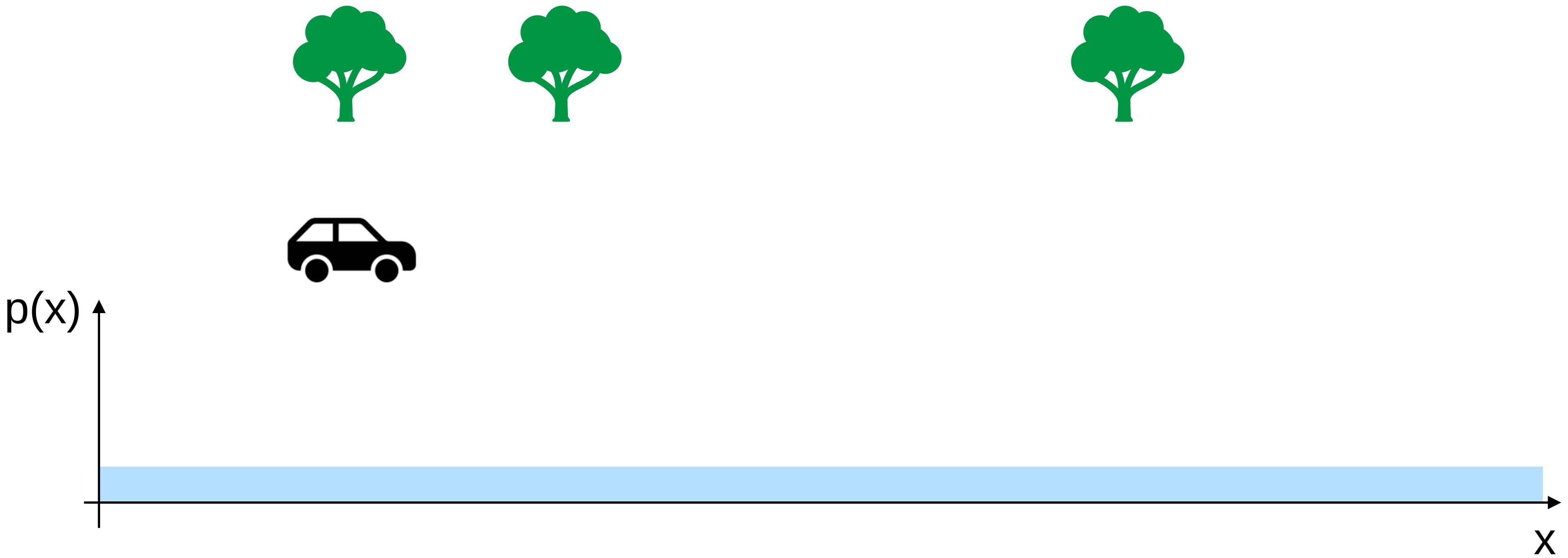
1. Motivation
2. Introduction to Probabilistics
3. Bayesian Filtering
4. **Probabilistic Localization**
5. Map representations
6. Summary

Probabilistic Localization

Known:

- Environment
- Ego-Motion (with uncertainty)
- Environment perception (with uncertainty)

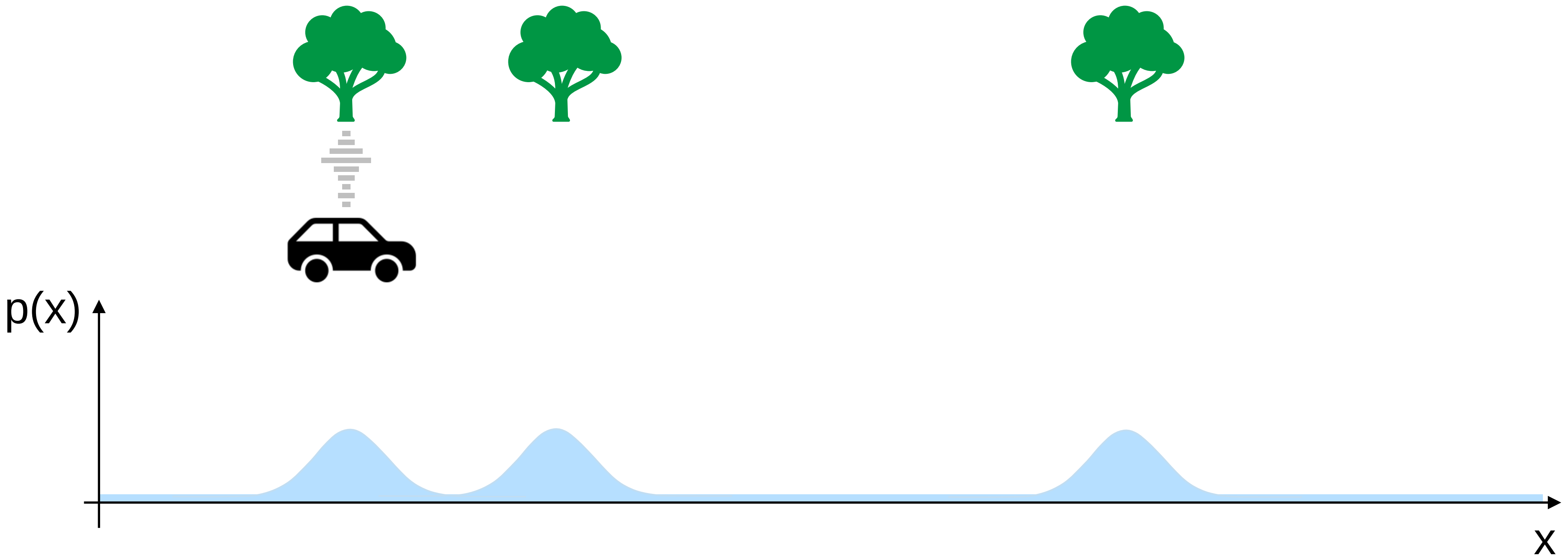
Probabilistic Localization



No motion update

No perception measurement

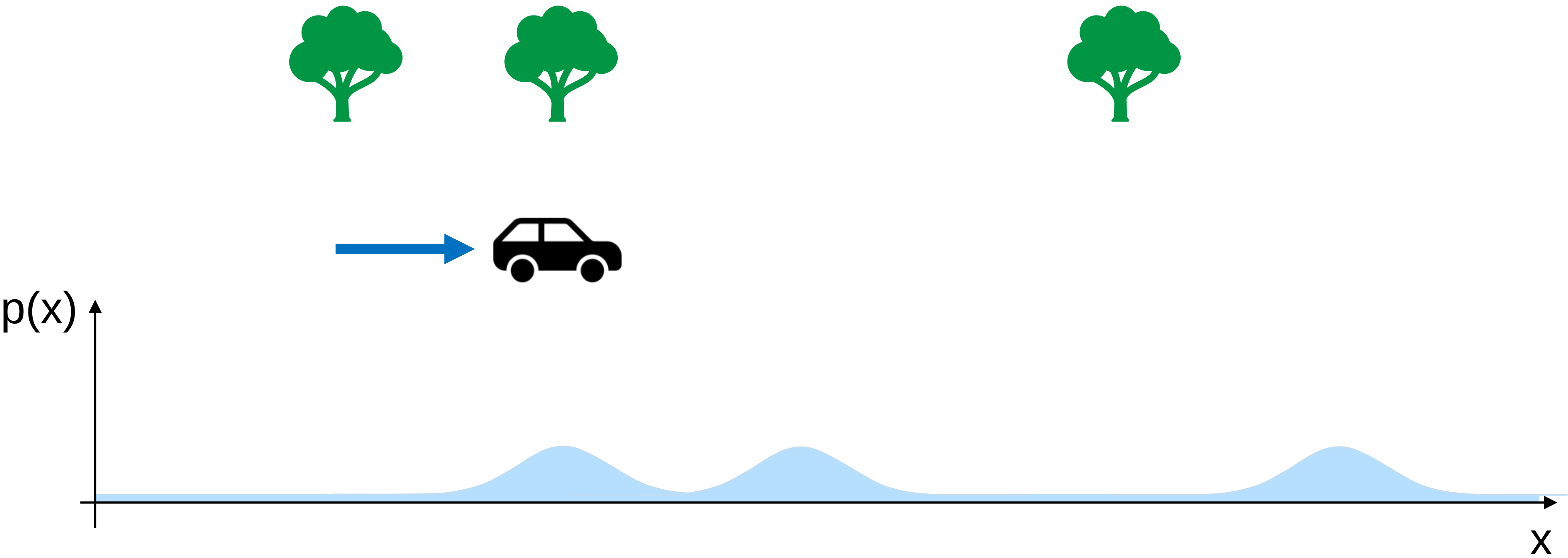
Probabilistic Localization



No motion update

Tree perceived

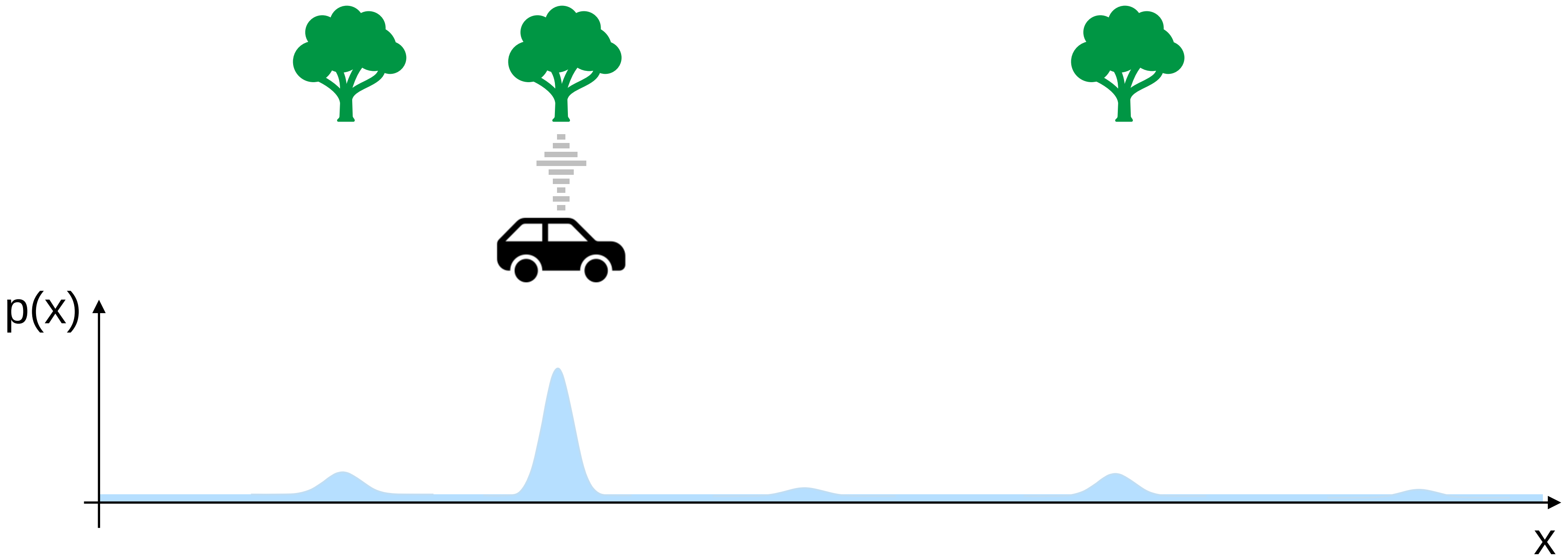
Probabilistic Localization



Motion update

No perception

Probabilistic Localization



No motion update

Second tree perceived

Probabilistic Localization

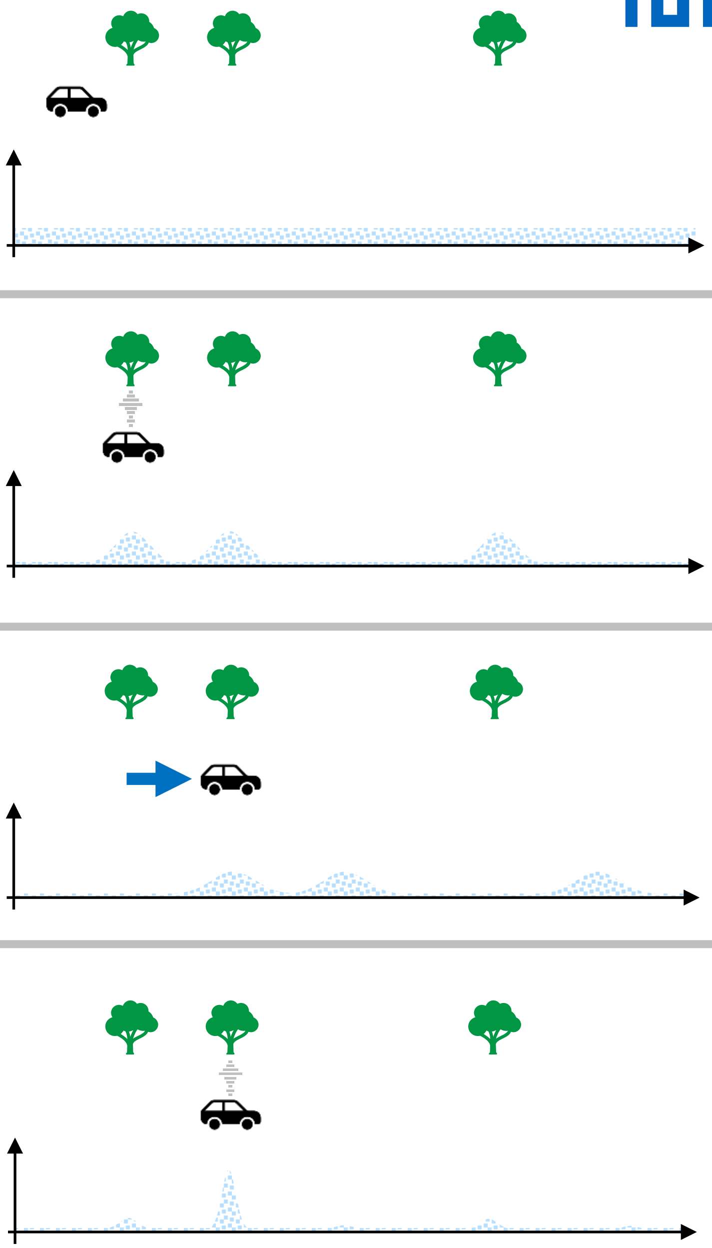
Particle Filter

Real probabilistic distributions are non-gaussian

Particles represent single ‘guesses’ of the state

Distribution of particles can represent different probabilistic distributions

Monte Carlo Localization



Probabilistic Localization

Monte Carlo Localization

When we suppose a known map in which we want to localize, particle filter localization is also known as Monte Carlo localization.

Probabilistic Localization

Definition

Odometry: Estimating the movement of the ego vehicle

- Can also be without position
- Wheel speeds
- IMU
- Visual Odometry

SLAM: Estimating the ego path and a map of the surrounding

- LiDAR
- Camera
- Integration of additional sensors like GPS/IMU

Probabilistic Localization

Localization without a map

For some applications no map is needed

Extraction of relevant information for the ego-vehicle

Only relative position to single ‘objects’ considered

Example:

Lane Keeping Assistant

https://github.com/StevieG47/Lane-Detection

Localization & Mapping I
Prof. Dr. Markus Lienkamp

Florian Sauerbeck, M. Sc.

Agenda

1. Motivation
2. Introduction to Probabilistics
3. Bayesian Filtering
4. Probabilistic Localization
5. **Map representations**
6. Summary

Map representations – Introduction

Kalman Filter – How to estimate the vehicle state

Localization – Use of relation to environment

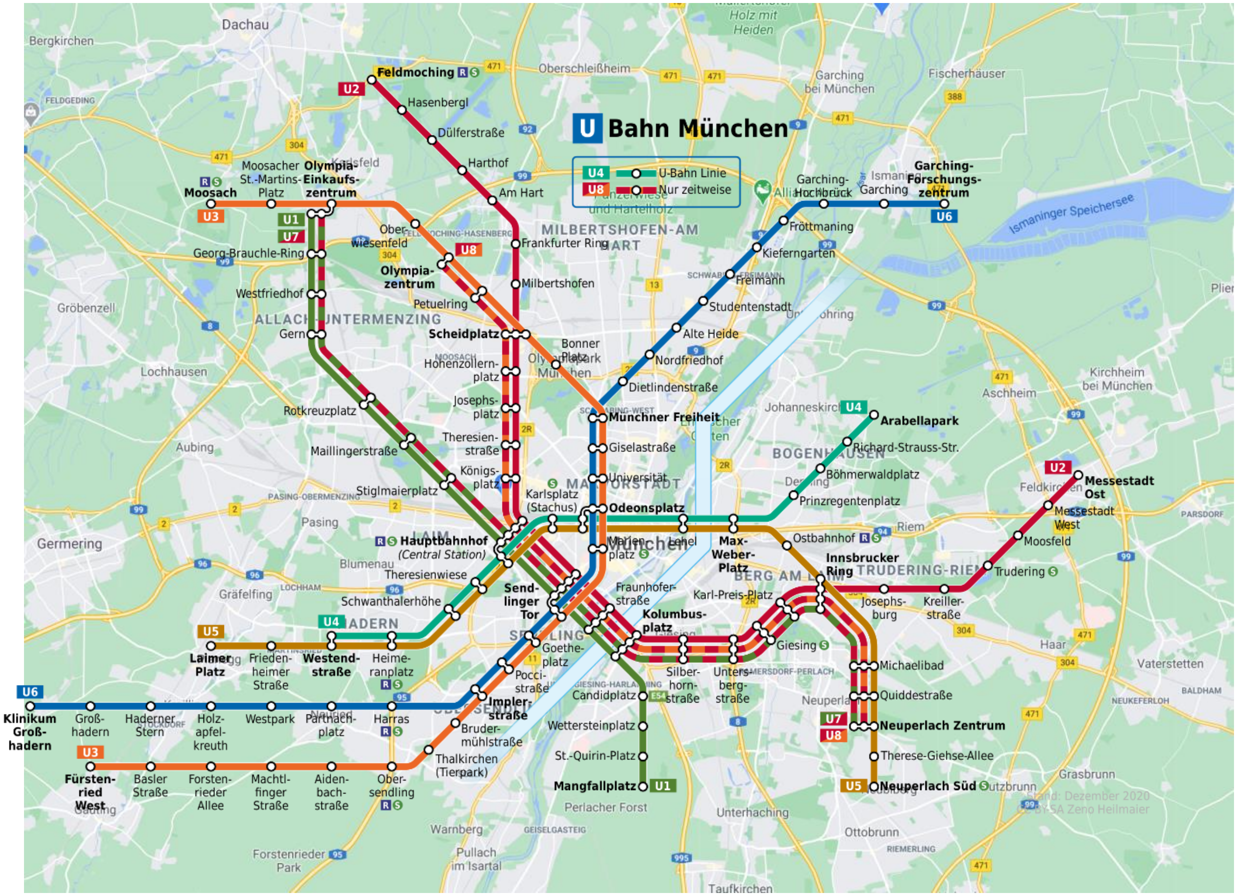
Problem

Environment can not be used directly and has to be depicted

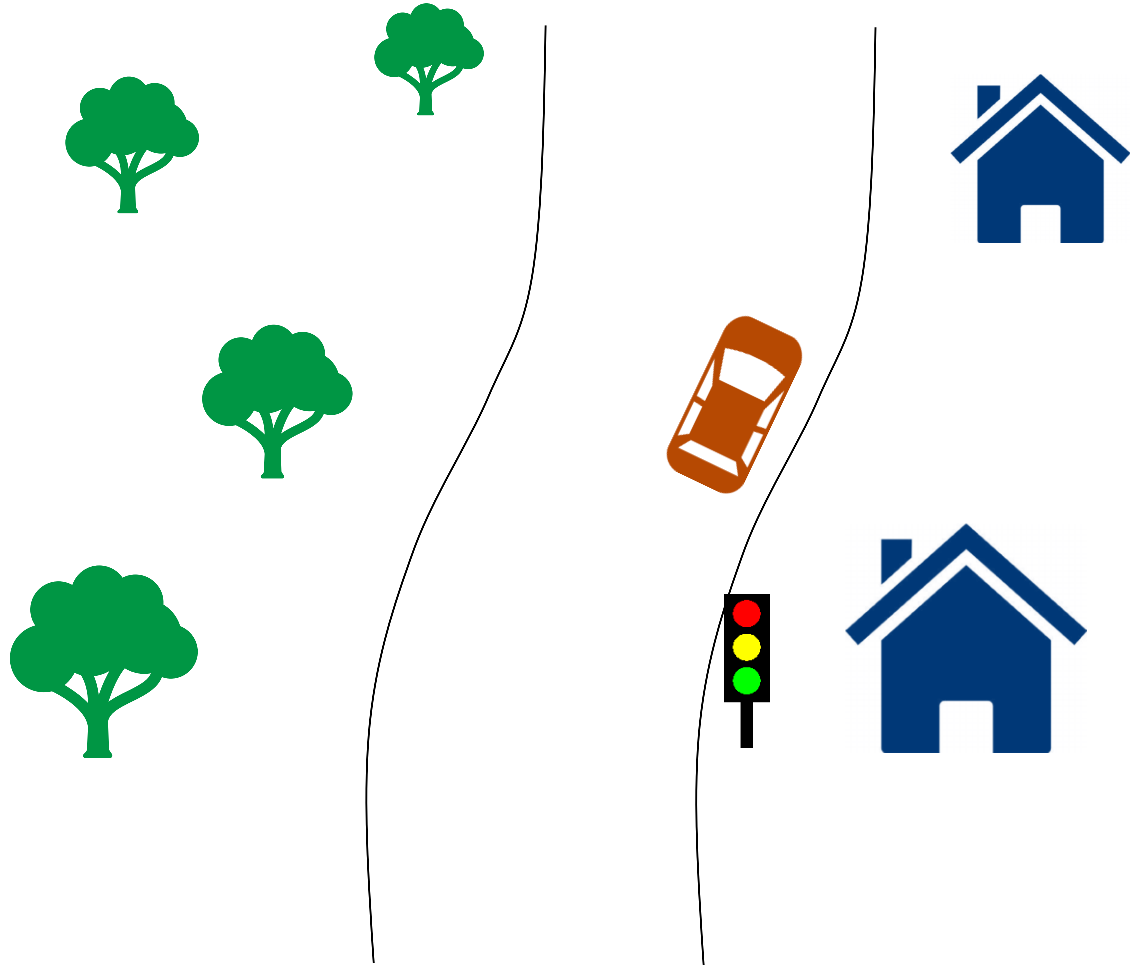
→ Need of different **map representations** for different applications

Map representations

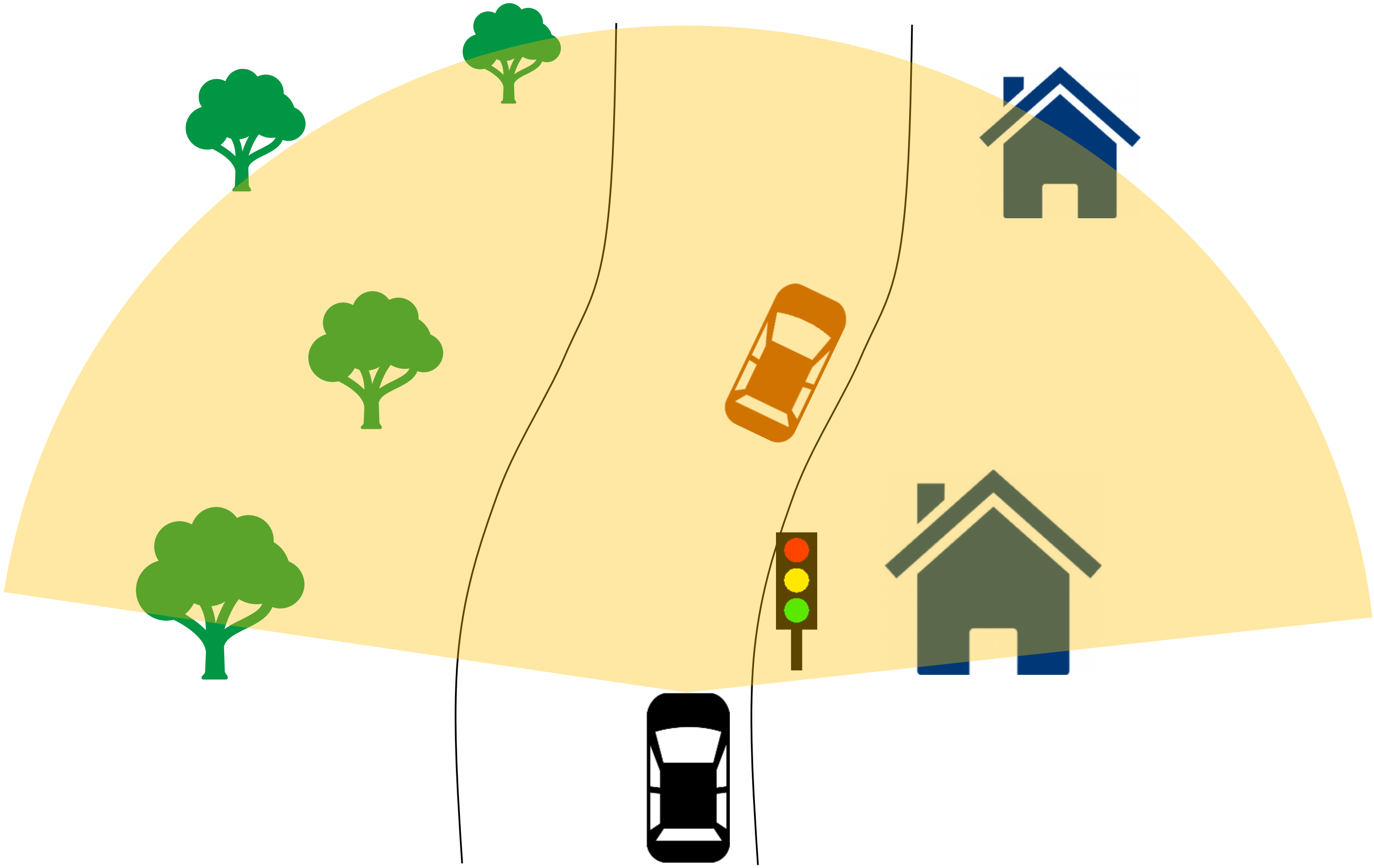
Either **topological** or **metric** based on application



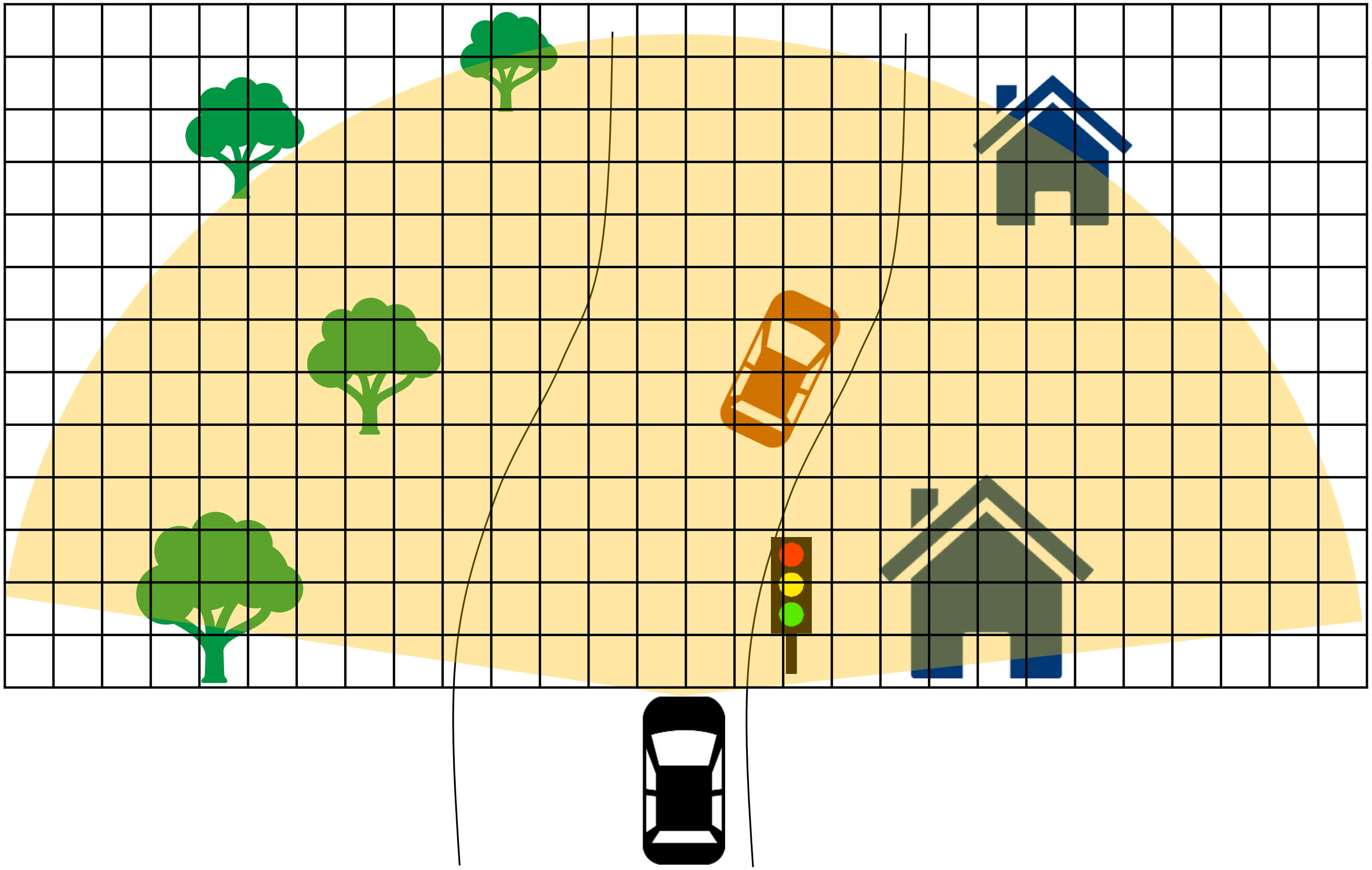
Map representations – Example Setup



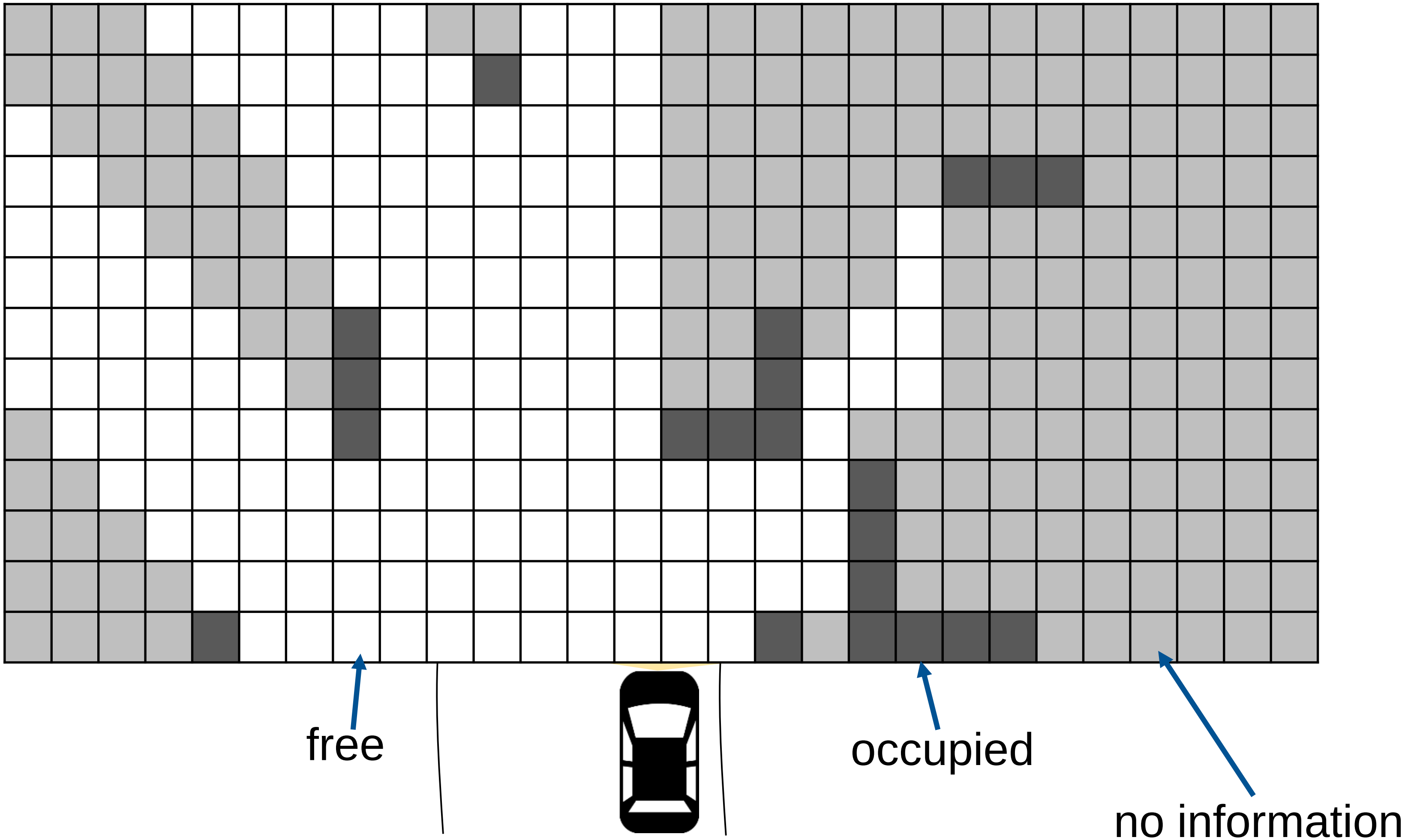
Map representations – Example Setup



Map representations – Occupancy Grid Maps



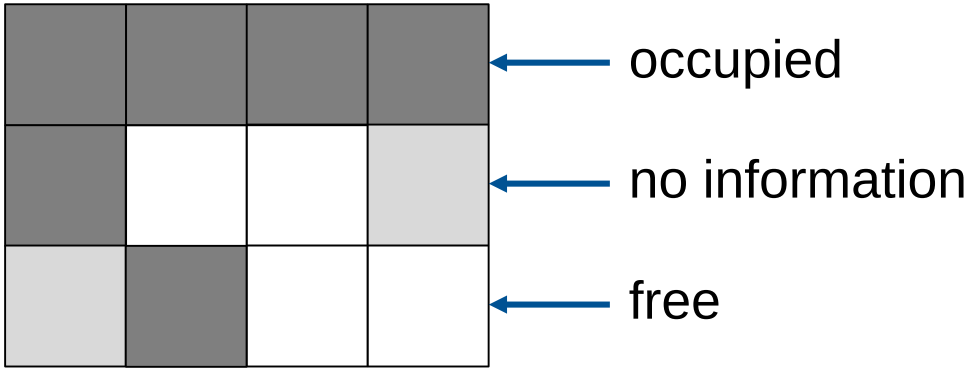
Map representations – Occupancy Grid Maps



Map representations

Occupancy Grid Maps

- Discretization of the map
- Each cell can be occupied, free or no information
- Mainly used for 2D-mapping
- Mainly used with LiDAR-sensors



Map representations

Occupancy Grid Maps

- Fixed discretization as map resolution
- Overlay of laser measurements depending on ego-motion

Map representations

Occupancy Grid Maps

The map in the picture on the slide before was generated with a 1D-LiDAR. This measures the distance to surrounding objects at a specific height.

The single scans are accumulated using the estimated movement of the vehicle.

To get more insight into this, be excited for the next lecture.

Map representations

Occupancy Grid Maps: Pros and Cons

+ Easily understandable
+ Directly usable for path planning
+ Relatively easy to create

- Difficult to use in 3D
- Limited precision of localization
- Fixed discretization

Map representations

Point Cloud Map

Occupancy Grid Map: Depicts world as a 2D space

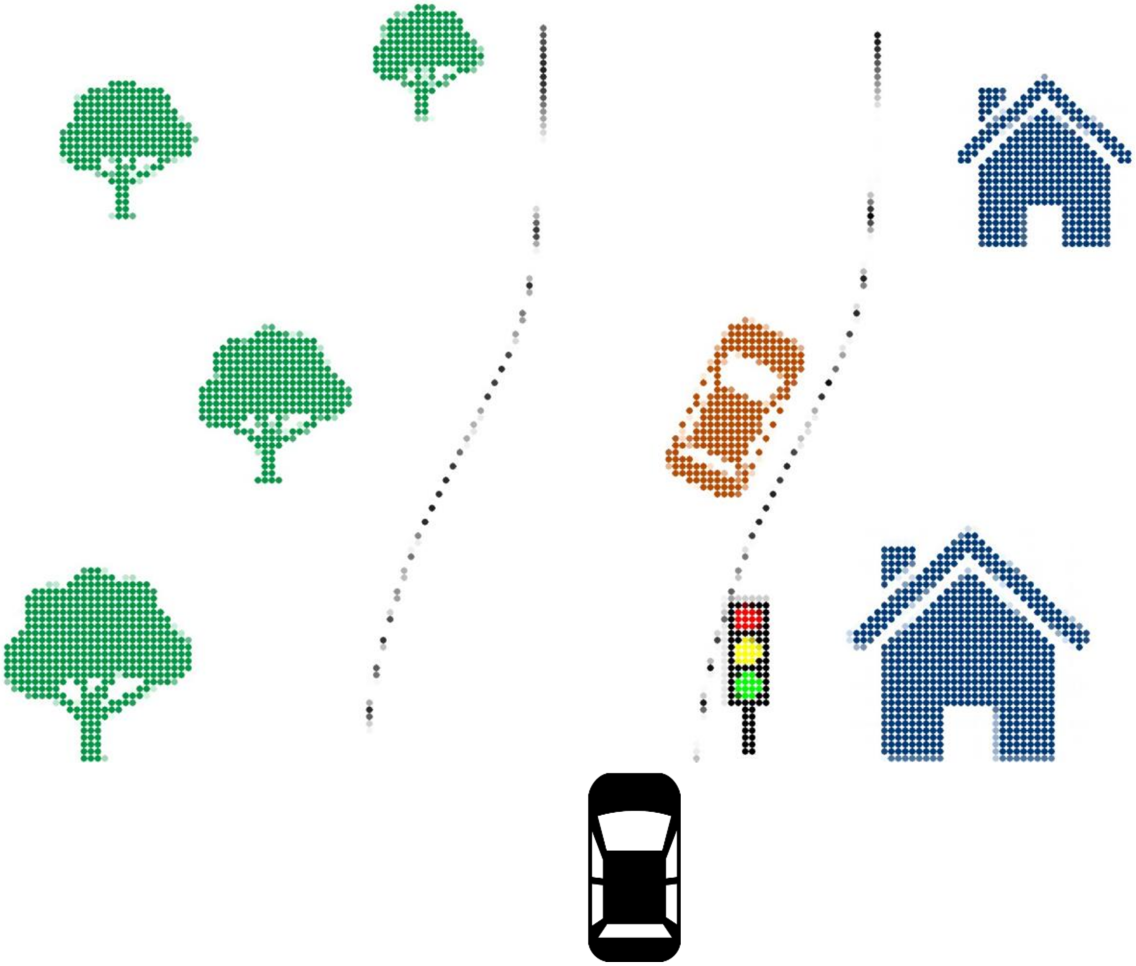
Problem

3D information is needed for level 5 autonomous driving

Solution

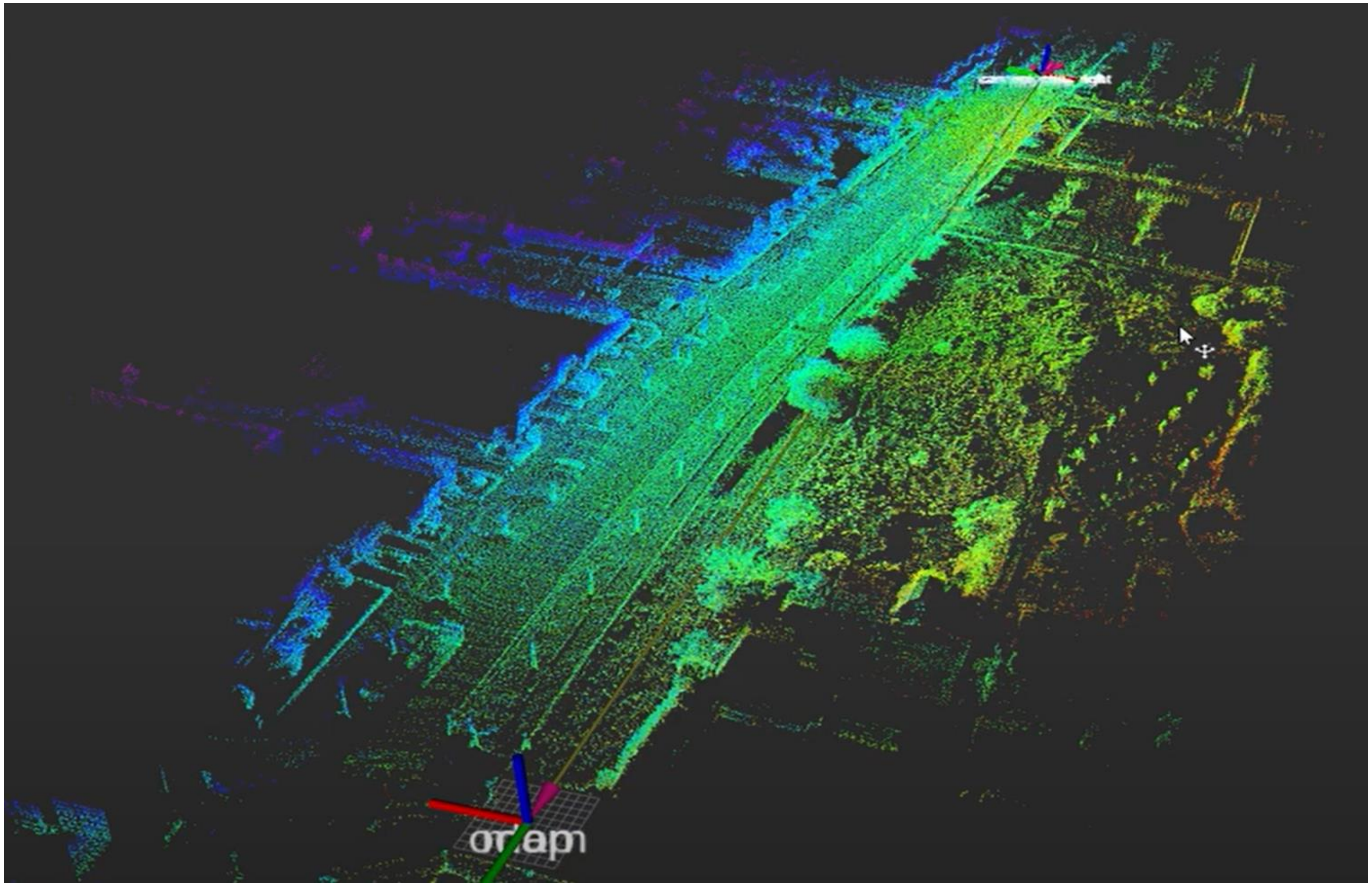
Use of 3D point clouds combined in one map

Map representations – Example Setup



Map representations

Point Cloud Maps



https://www.youtube.com/watch?v=W3gOJlgFfKg

Map representations

Point Cloud Maps: Pros and Cons

+ Easy understandable
+ Directly usable for path planning
+ Relatively easy to create

- Limited precision of localization
- Expensive 3D LiDAR
- Amount of data
- Inclusion of irrelevant information

Map representations

Feature Maps

Problem

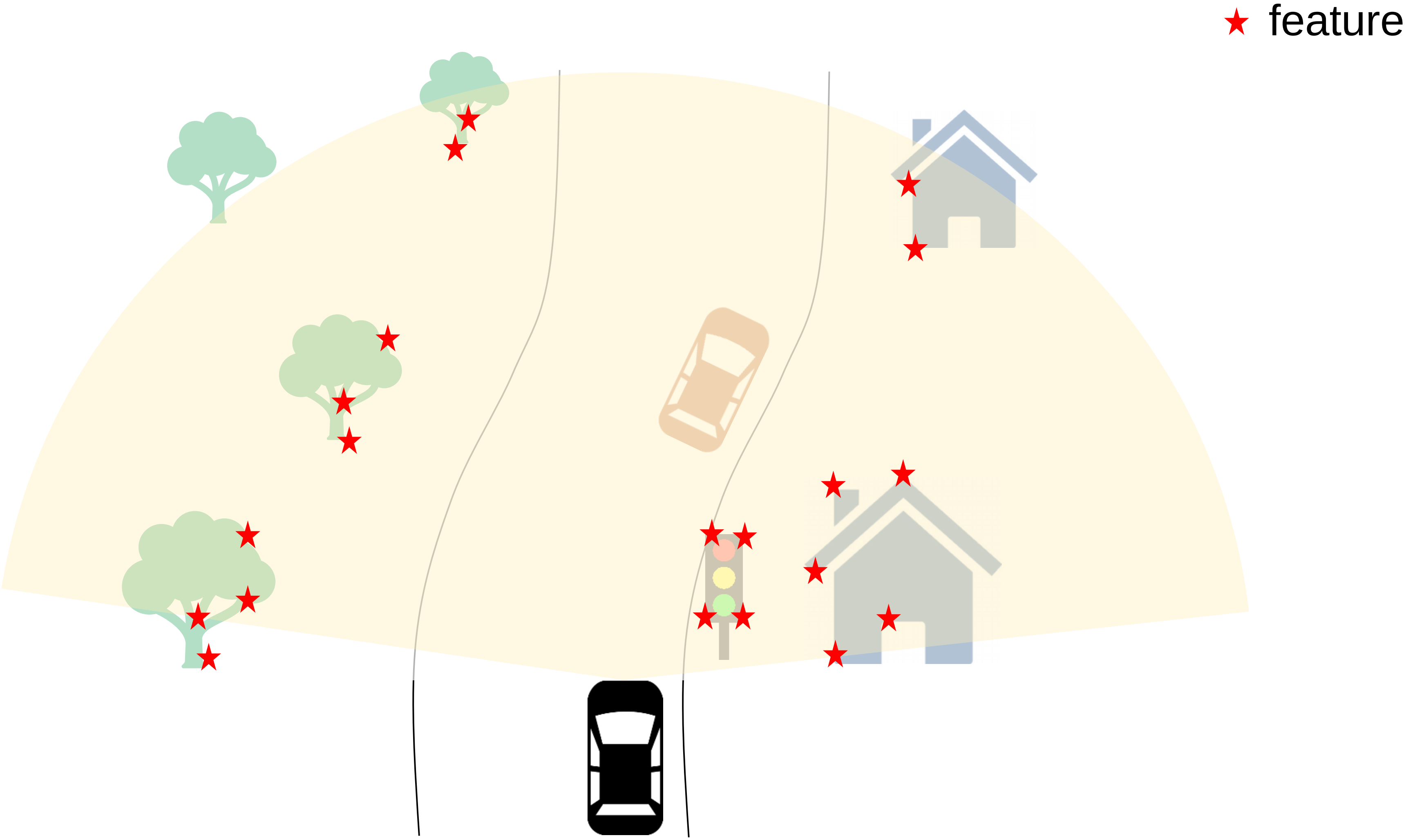
How to build a map based on camera images?

Solution

Extract features from images and save them in a 3D map

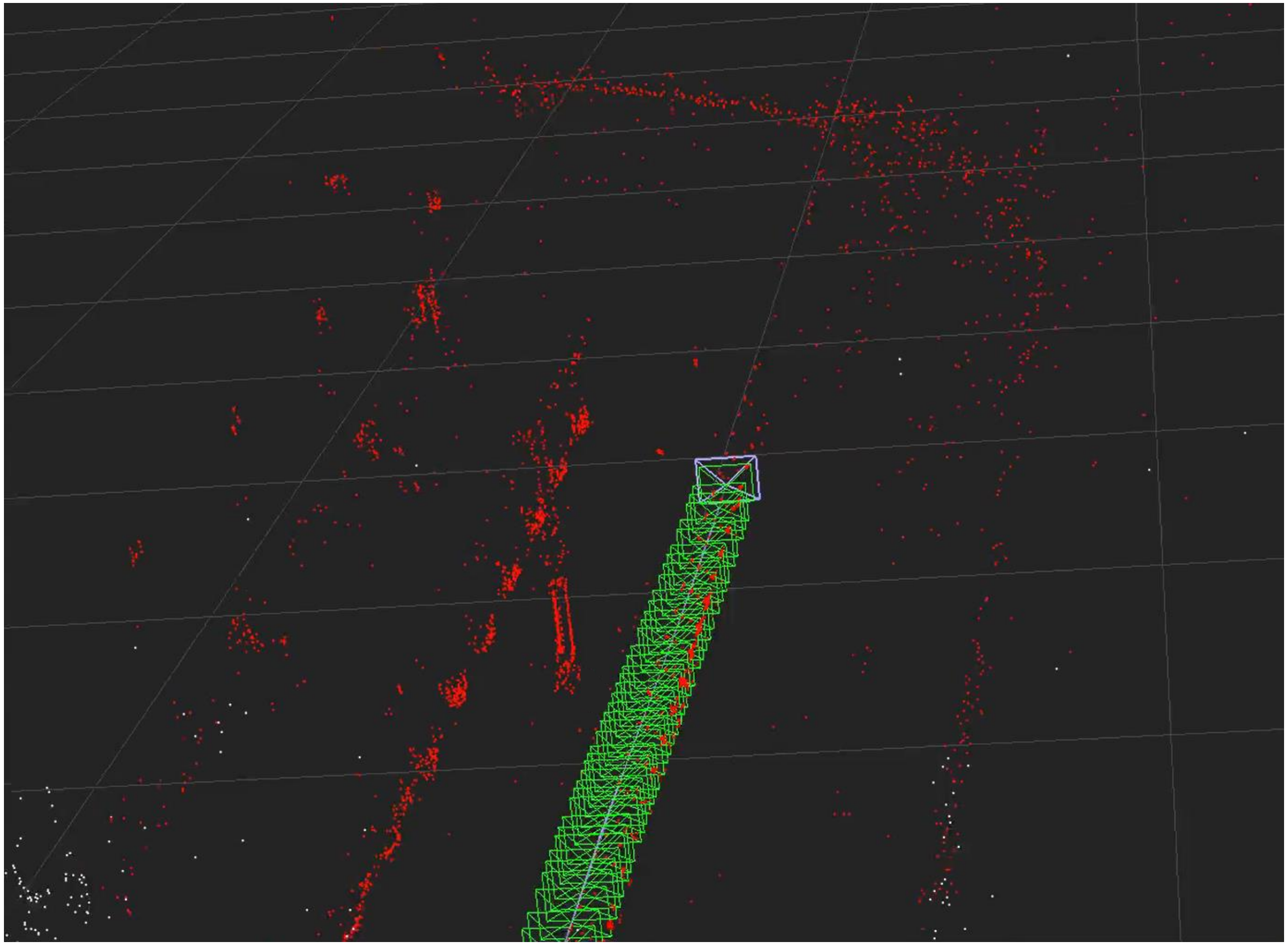
- Features extracted from environment as fix-points
- 3D representation possible
- Features can be extracted from LiDAR and cameras

Map representations – Feature Maps



Map representations

Feature Maps



Map representations

Feature Maps: Pros and Cons

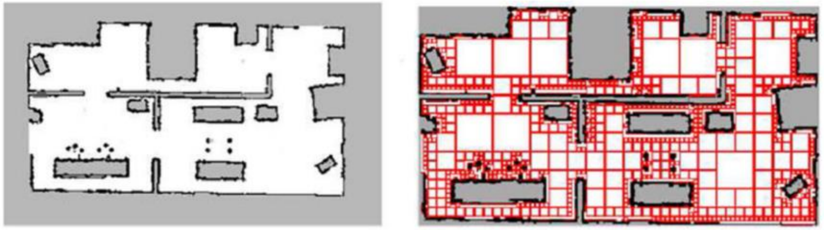
+ Applicable in 2D and 3D
+ No discretization needed
+ Compatible with different sensors
+ Low memory consumption

- Not understandable for humans
- No information on occupancy

Map representations

Semantic Maps

- Detection and position estimate of objects
- Inclusion of semantic information
- Usually combined with other map representations

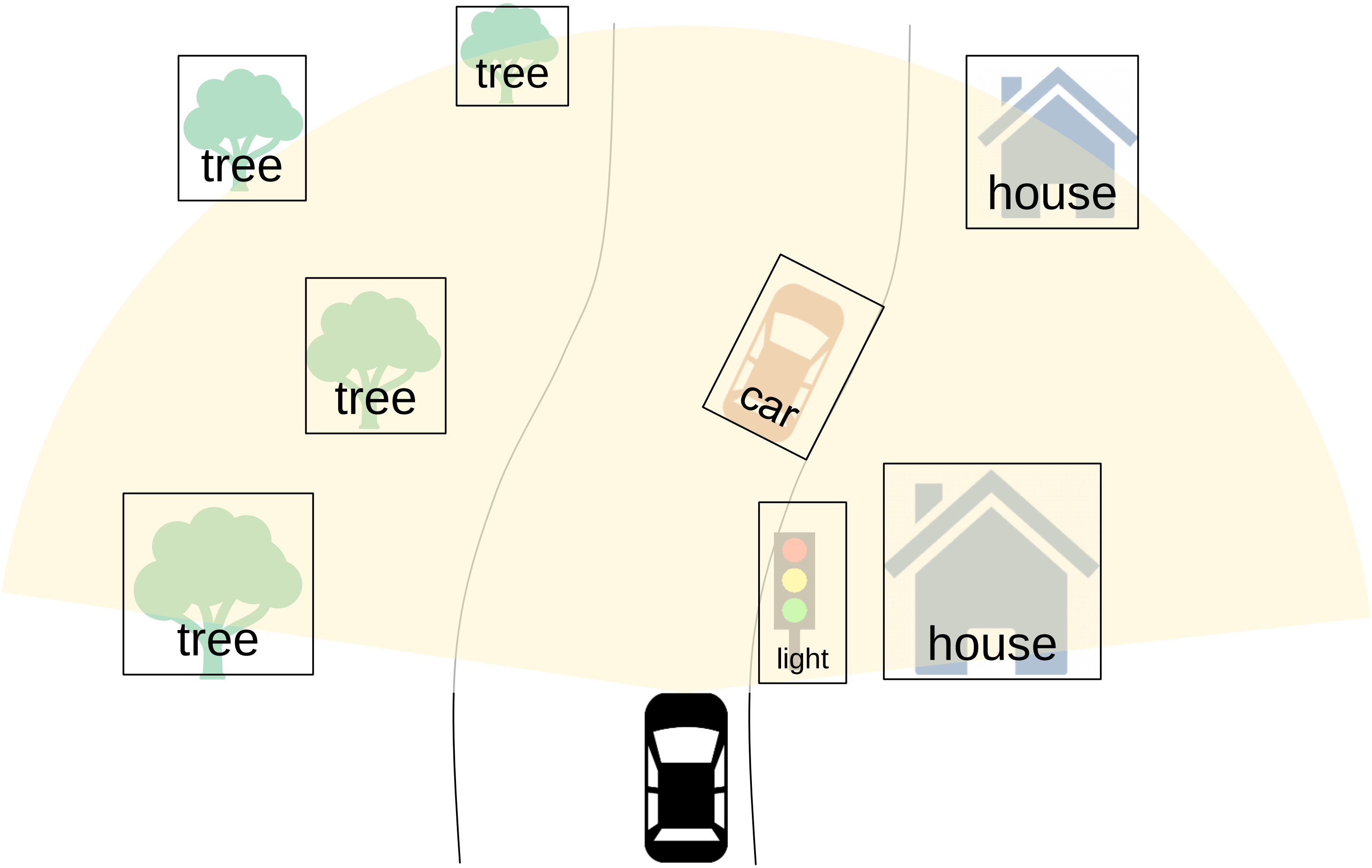


2D grid map

3D map with semantic information

Zhao, Z., Chen, X. Building 3D semantic maps for mobile robots using RGB-D camera. *Intel Serv Robotics* **9,** 297–309 (2016). https://doi.org/10.1007/s11370-016-0201-x

Map representations – Example Setup



Map representations

Semantic Maps: Pros and Cons

+ Include semantic information
+ Different object classes instead of abstract features
+ Additional information for path planning

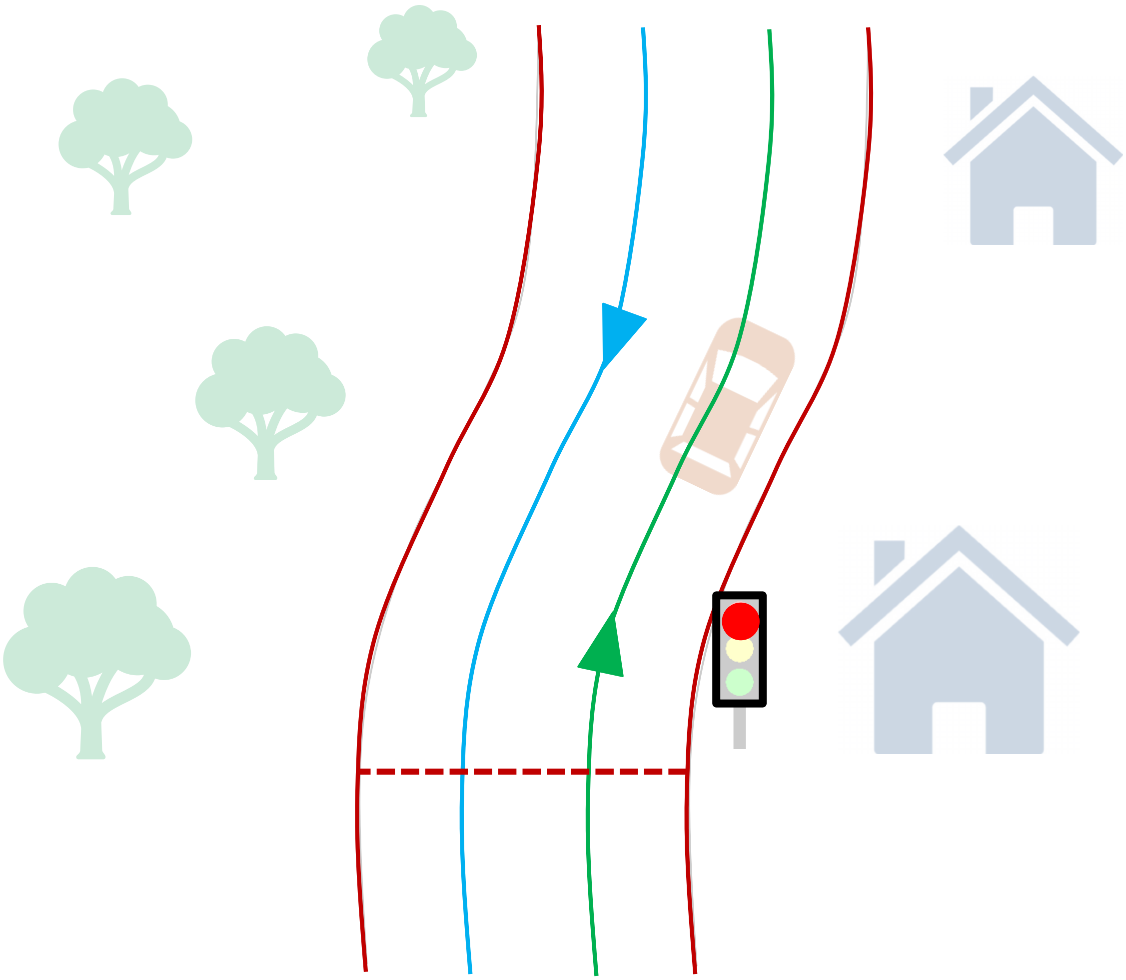
- Need additional information on surroundings
- Dependent on object detection
- Compute ressources and computation time

Map representations

High Definition Maps

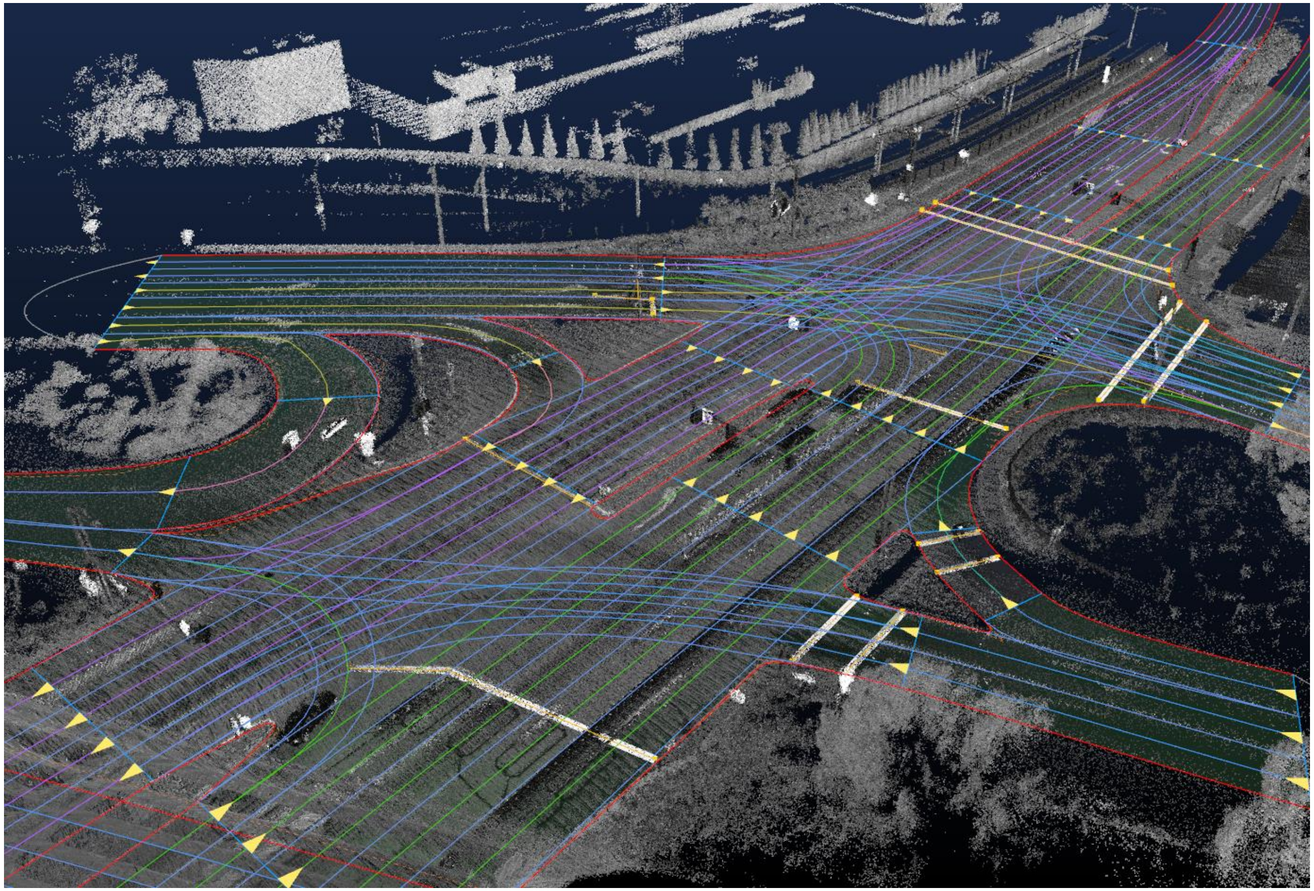
- Offline generated, online localization
- Combination of many different sources
- Includes additional information like street lanes, speed limits, traffic lights etc.

Map representations – Example Setup



Map representations

High Definition Maps



https://www.geoawesomeness.com/wp-content/uploads/2020/04/HD-Live-Map.png

Map representations

HD Maps: Pros and Cons

+ High information content
+ Perfectly adjustable for application
+ Integration of many sensors

- Expensive to create
- Difficult to keep up-to-date
- High demands on memory and bandwidth

Map representations – Outlook

Elon Musk Declares Precision Maps A "Really Bad Idea" -- Here's Why Others Disagree

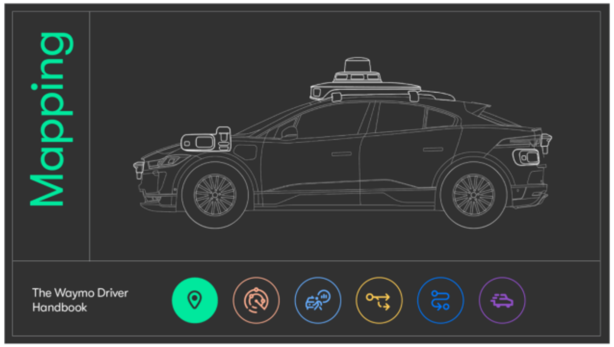
Brad Templeton Senior Contributor
Transportation
I cover robocar technology & previously worked on Google's car team.

https://www.forbes.com/sites/bradtempleton/2019/05/20/elon-musk-declares-precision-maps-a-really-bad-idea-heres-why-others-disagree/

September 21, 2020

The Waymo Driver Handbook: How our highly-detailed maps help unlock new locations for autonomous driving

TECHNOLOGY
The Waymo Team



https://blog.waymo.com/2020/09/the-waymo-driver-handbook-mapping.html

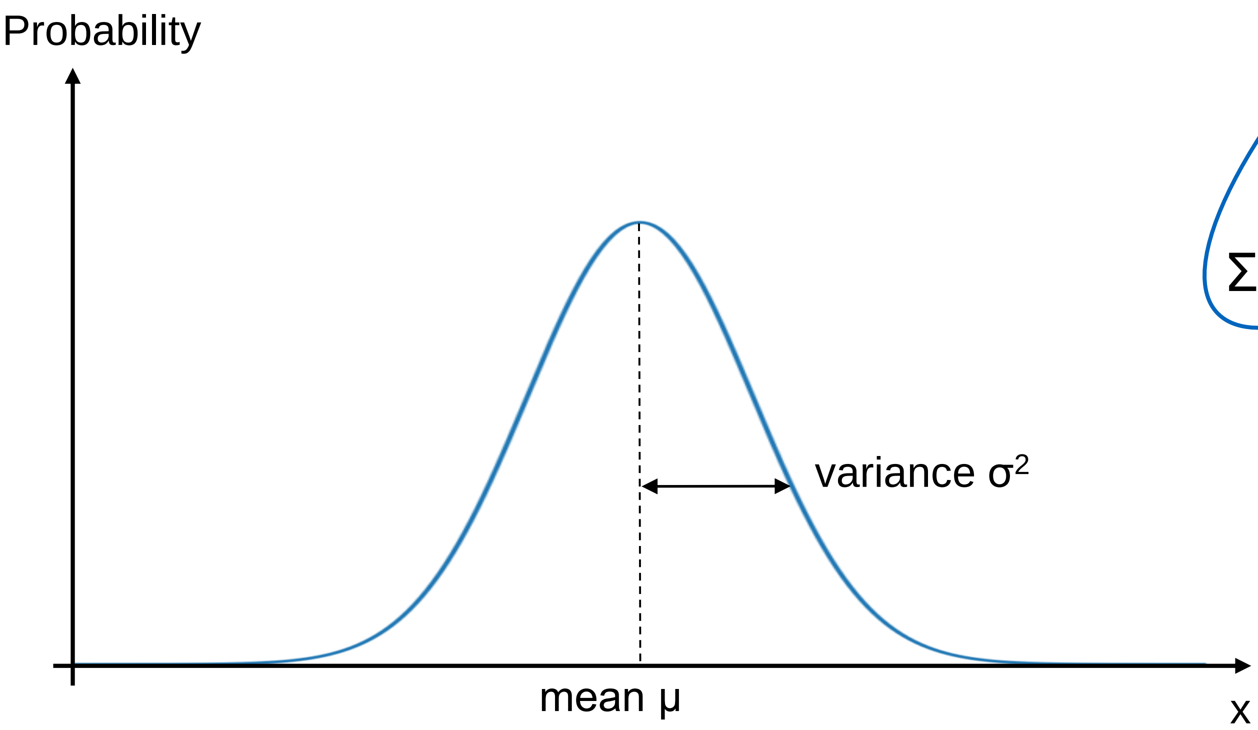
Localization & Mapping I
Prof. Dr. Markus Lienkamp

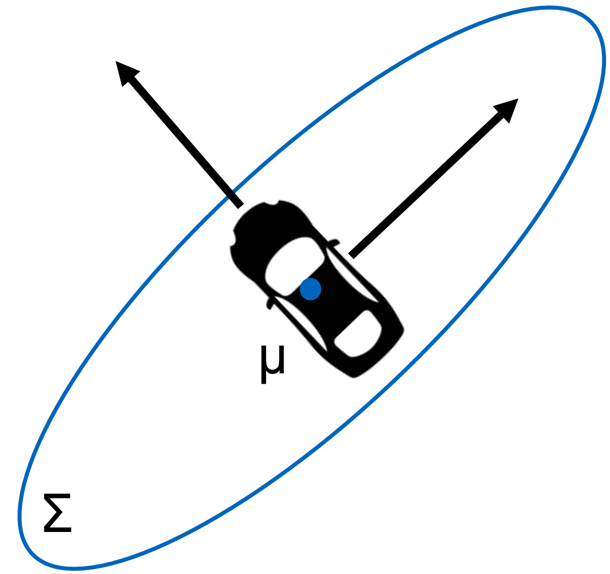
Florian Sauerbeck, M. Sc.

Agenda

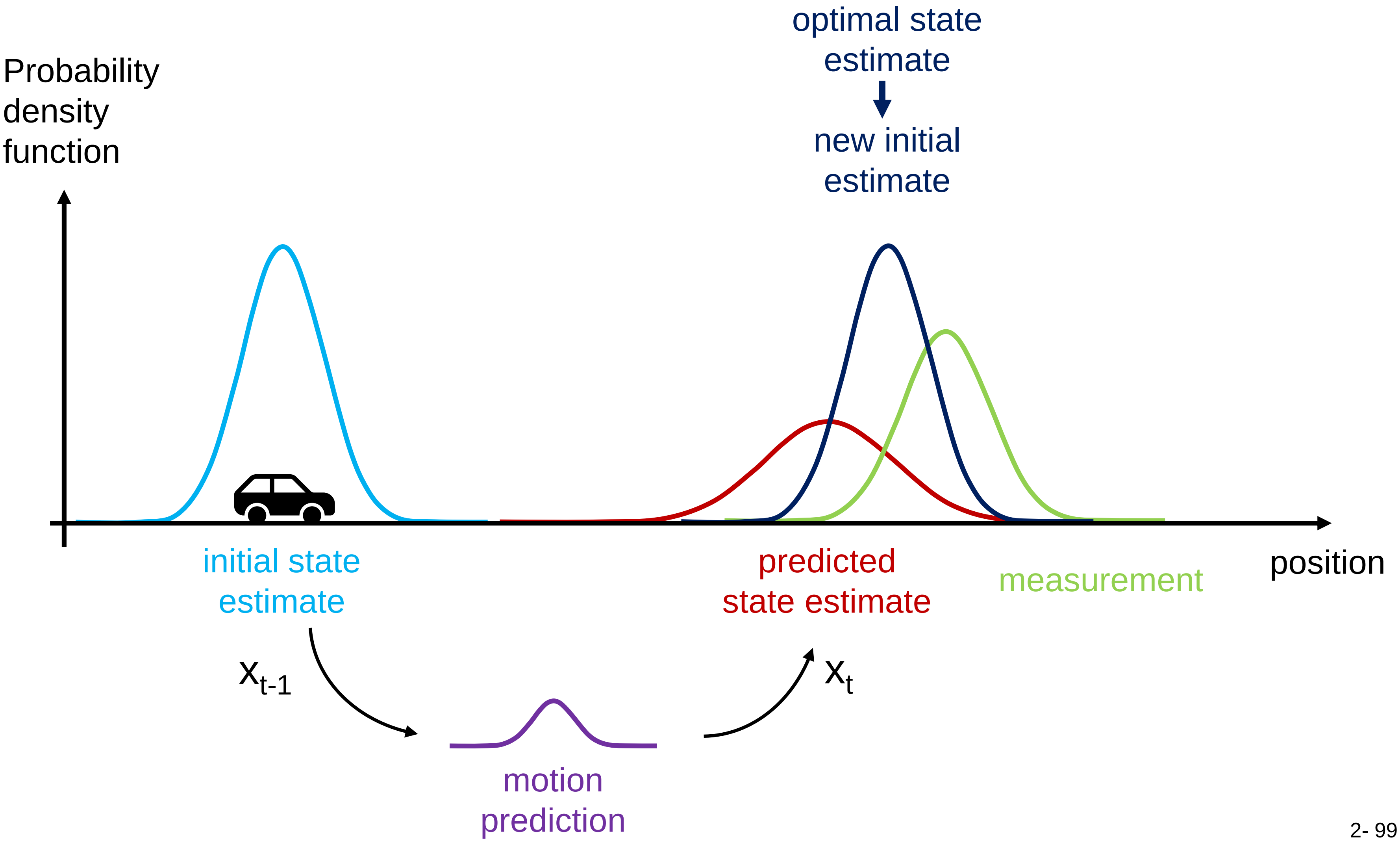
1. Motivation
2. Introduction to Probabilistics
3. Bayesian Filtering
4. Probabilistic Localization
5. Map representations
6. **Summary**

Summary – Probabilistic

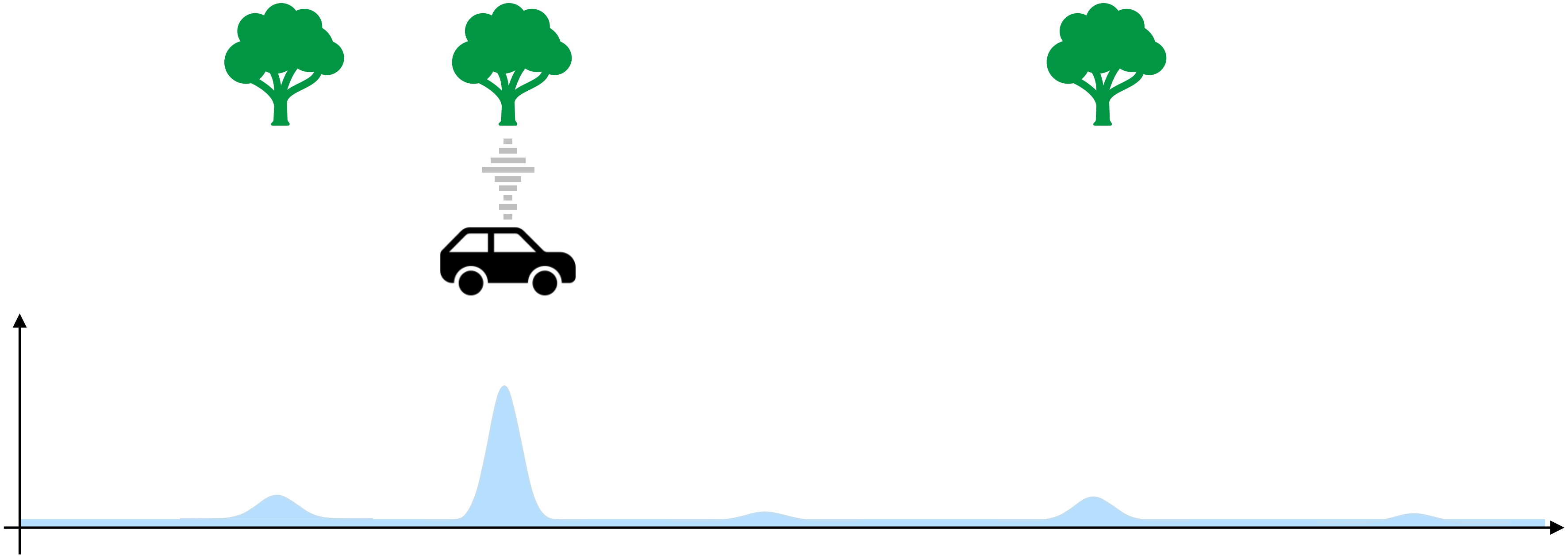




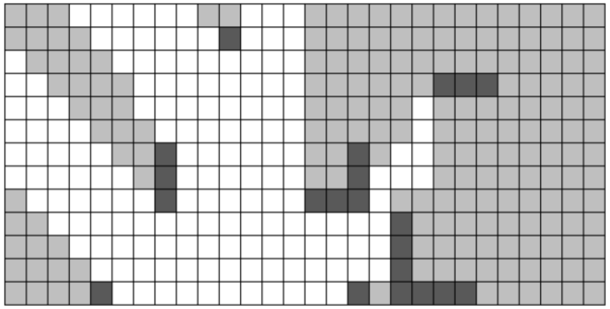
Summary – Kalman Filters

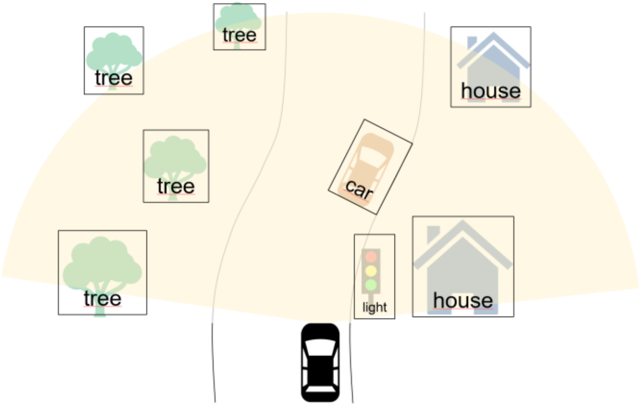


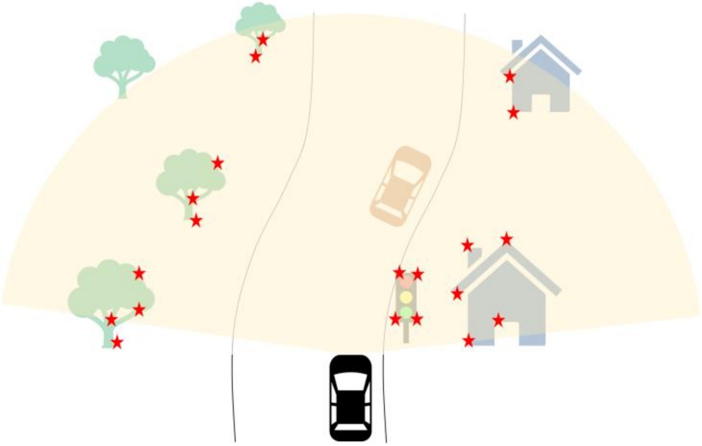
Summary – Probabilistic Localization

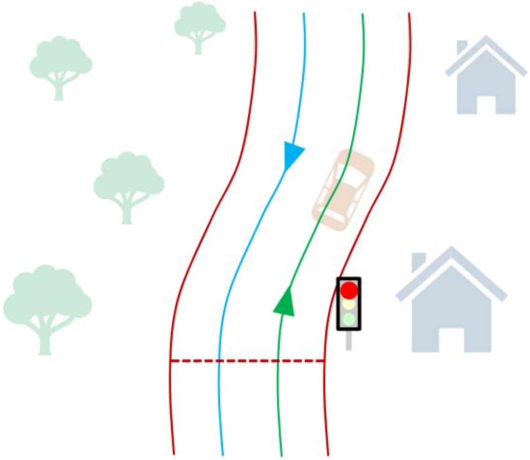


Summary – Map Representations









https://cdn.mos.cms.futurecdn.net/izXa9hiiSTbnNtwSp2ixKC-970-80.jpg.webp

Summary – What did we learn today

- Mapping and localization crucial for the whole autonomous software
- Probabilistic robotics
  - Sensor measurements can be handled as Gaussians with mean and variance
  - Addition of Gaussians
  - Multiplication of Gaussians
- Kalman Filter for Odometry
  - Fusion of sensors
  - Update step: Multiplication
  - Prediction step: Addition

Summary – What did we learn today

- Probabilistic Localization
  - Use of environmental information for estimating the ego position
- Map representations
  - Different map representations for different algorithms/applications
  - (Volu-)metric map
  - Occupancy grid map
  - Feature map
  - Semantic map
  - HD map