

Probabilistic Robotics Course

INTRODUCTION

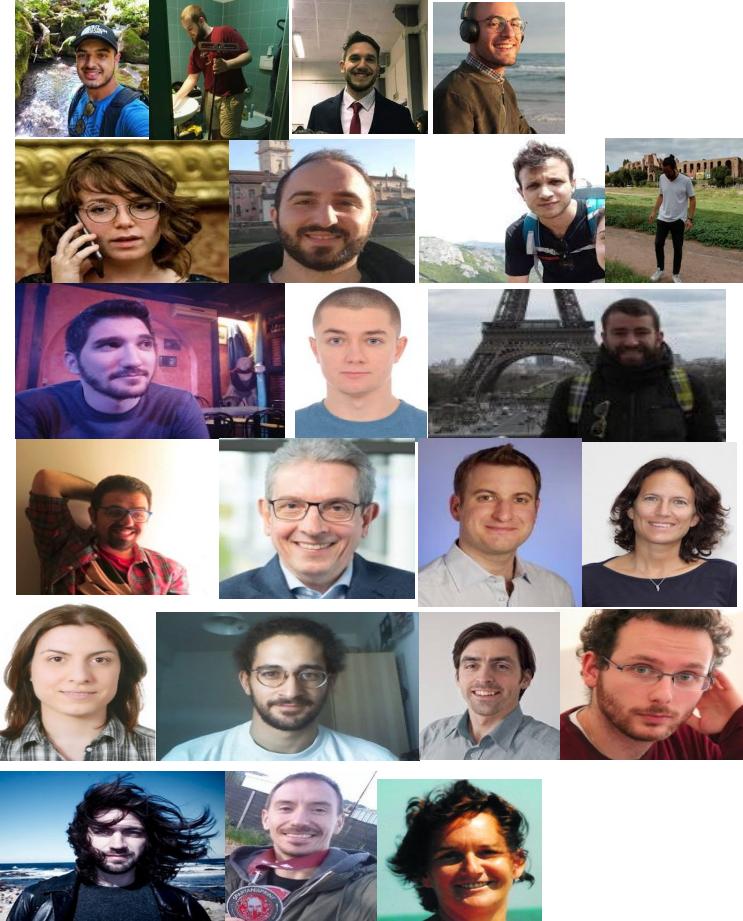
Giorgio Grisetti

grisetti@diag.uniroma1.it

Department of Computer Control and Management Engineering
Sapienza University of Rome

Thanks to

- Lorenzo De Rebotti
- Simone Ferrari
- Leonardo Brizi
- Emanuele Giacomini
- Omar Salem
- Barbara Bazzana
- Luca Di Giammarino
- Tiziano Guadagnino
- Bartolomeo Della Corte
- Dominik Schlegel
- Irvin Aloise
- Mirco Colosi
- Wolfram Burgard
- Cyrill Stachniss
- Maren Bennewitz
- Kai Arras
- Mayte Lazaro
- Federico Nardi
- Jacopo Serafin
- Maurilio Di Cicco
- Taigo Bonanni



Contacts

Teacher:

Giorgio Grisetti

Tutors:

Lorenzo DeRebotti

Omar Salem

Leonardo Brizi

Email:

- {lastname}@diag.uniroma1.it
- all emails you send us about the course should have **[prob-rob]** as first string in the subject.

Teaching Material

- The primary source of information for the course is this web page

<https://sites.google.com/diag.uniroma1.it/probabilistic-robotics-2023-24/home>

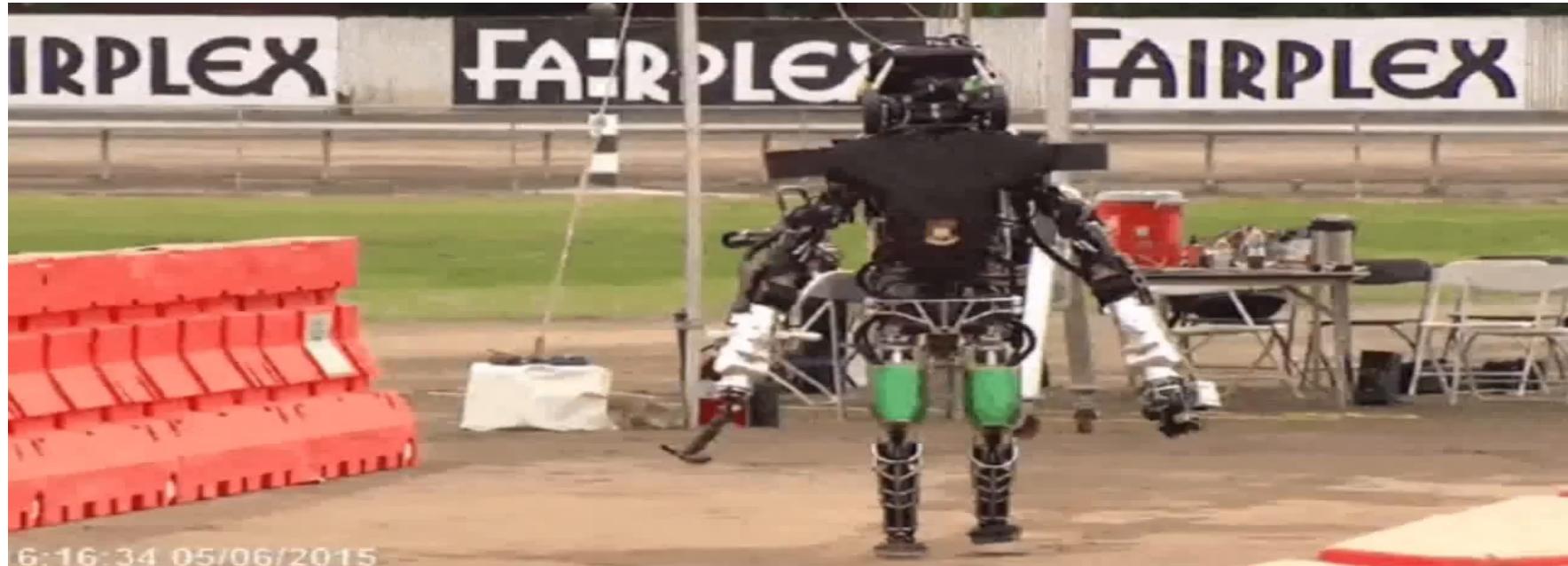
- The material

- Slides
- Source code for practicals

Is also available at the following repository

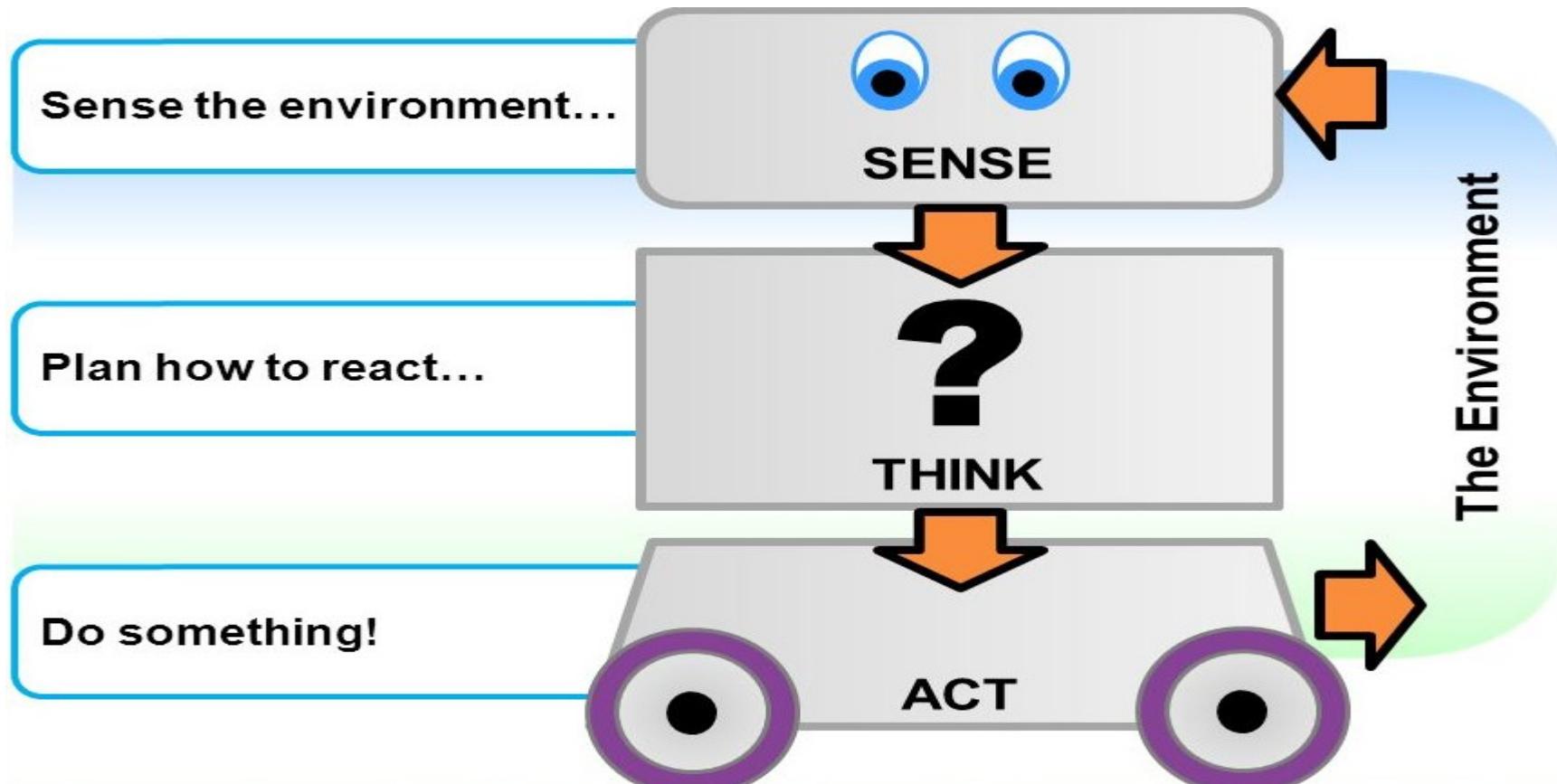
https://gitlab.com/grisetti/probabilistic_robotics_2023_24

Robots Have to Work



<https://www.youtube.com/watch?v=wX0KagJ1du8>

Sense-Plan-Act



State in Robotics

Model of the world

- Geometry
- Traversability
- Other moving objects
- ...

Robot configuration

- Kinematics
- Dynamics
- State of batteries

...



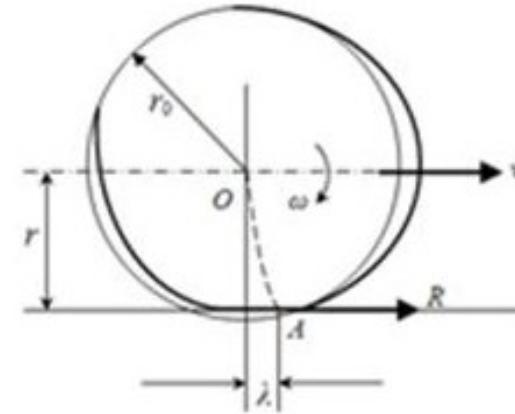
Probability and Robotics

The model of a system is typically an abstraction of a more complex real entity

Disturbances affect the measurements and the system

Measurement might be poorly informative

The predicted behavior is different from the real one



Probability and Robotics

The model of a system is typically an abstraction of a more complex real entity

Disturbances affect the measurements and the system

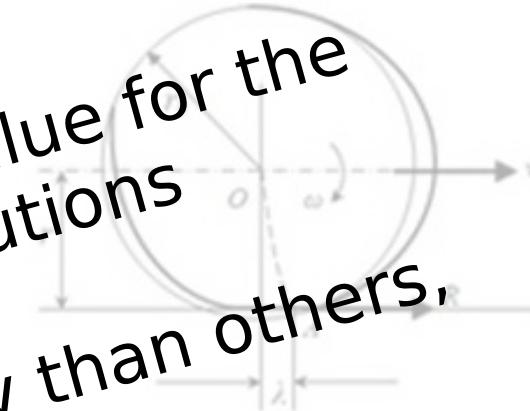
Measurement might be poorly informative

The predicted behavior is different from the observed one

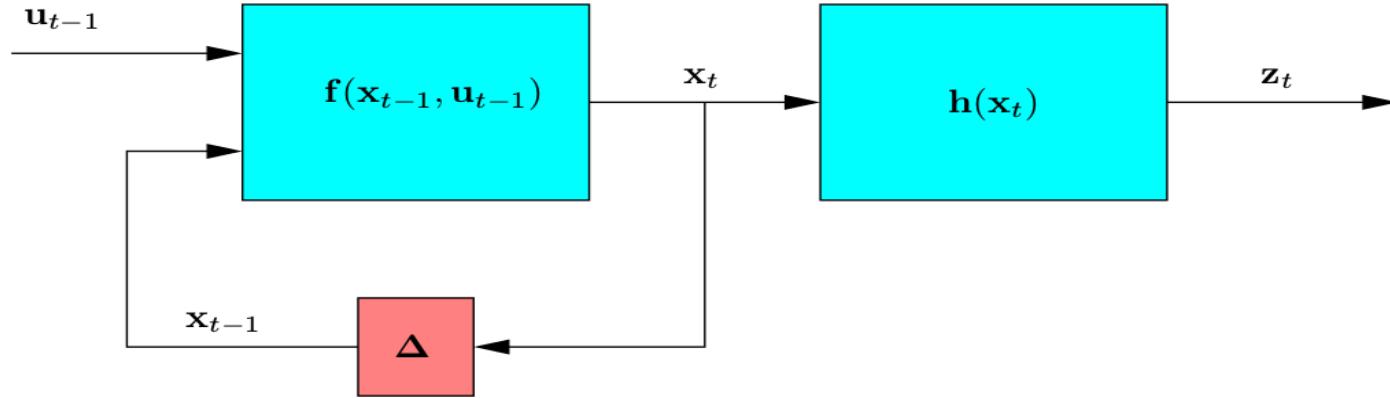
There is not a single possible value for the state, but a set of potential solutions

Some solutions are more likely than others, given the current measurements

We will use probabilistic inference



System Model



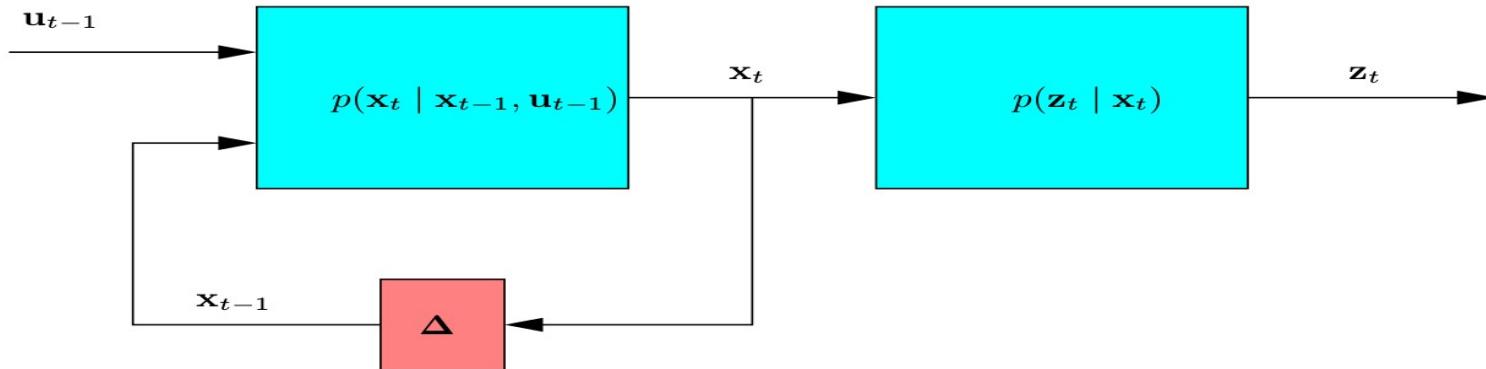
Notation:

- \mathbf{u} : controls
- \mathbf{x} : state
- \mathbf{z} : measurement
- \mathbf{f} : transition function
- \mathbf{h} : observation function

Perfect knowledge of

- inputs
- measurements
- transition model
- observation model

System Model



Probabilistic model

- The variables become stochastic (uncertain)
- The transition function is replaced by a transition model, that describes a conditional probability distribution over the next states as a function of the previous states
- The observation function is replaced by a conditional probability describing the distribution of possible observations as a function of the current state
- If no uncertainty is present the probabilistic models degenerates to the deterministic model

What will I learn?

Tools

- Filtering tools
 - Discrete
 - Gaussian
 - Particle
- Maximum Likelihood Estimation
 - Gauss-Newton
 - Sparse Least Squares
- Data Association
 - Greedy
 - Voting Schemes
 - Spectral Methods

Applications

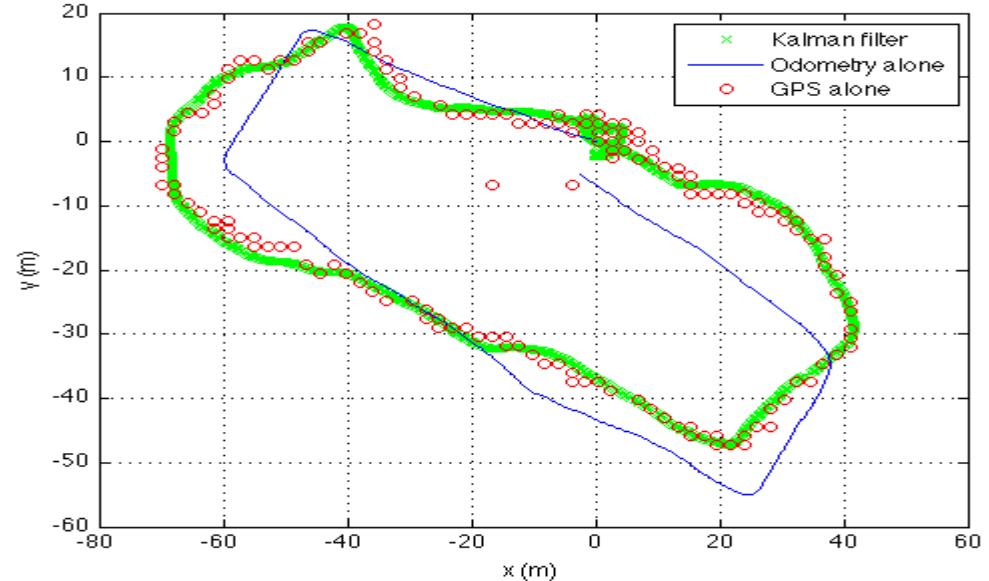
- Calibration
- Tracking
- Localization
- Mapping
- SLAM

Filtering

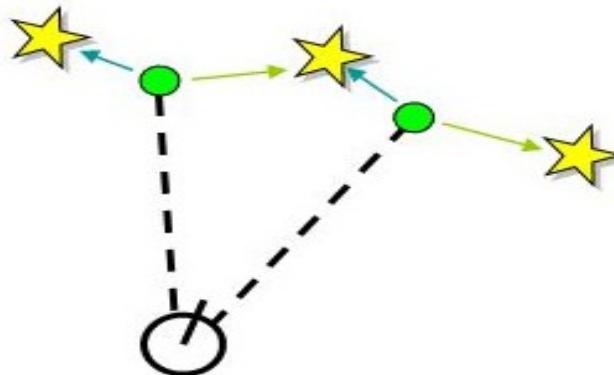
Filtering is the process of estimating the distribution over the possible **current** states of a dynamic system

In doing so we have access to

- the controls
- the measurements



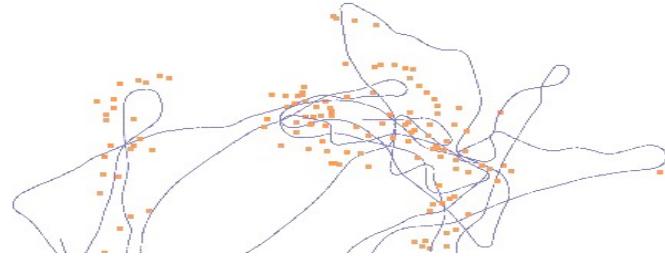
Data Association



Determine which state variable (if any) is responsible of a measurement, based on:

- the current estimate of the state
- the history of measurements

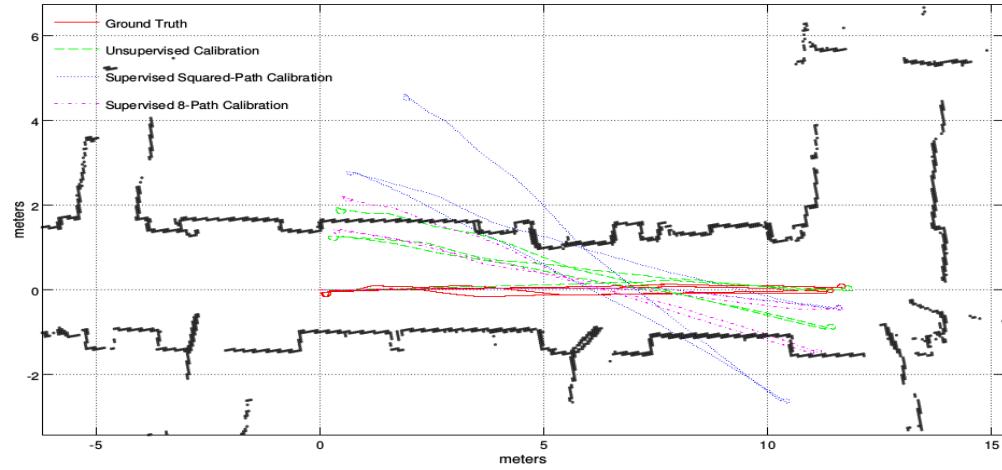
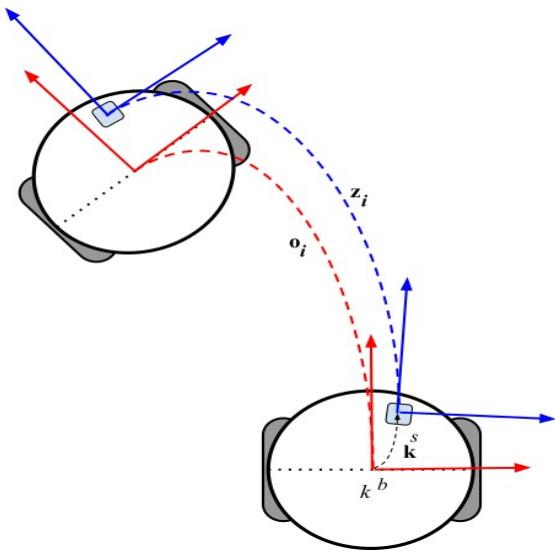
Maximum Likelihood Estimation



Estimate the most likely trajectory of the system's state given all measurement so far

In case of stationary system, the trajectory degenerates to a single state

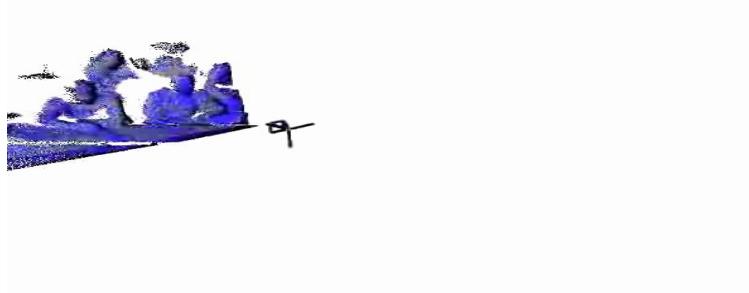
Calibration



Determining the kinematic parameters of a robotic system given

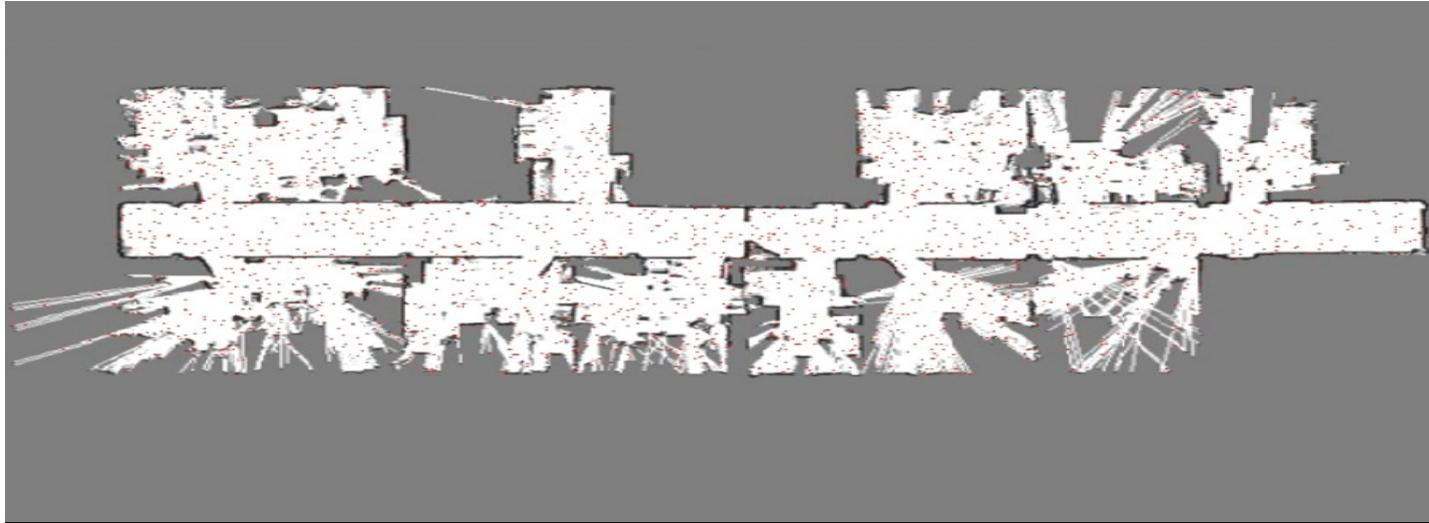
- trajectory of the sensors (externally measured)
- encoder readings

Tracking



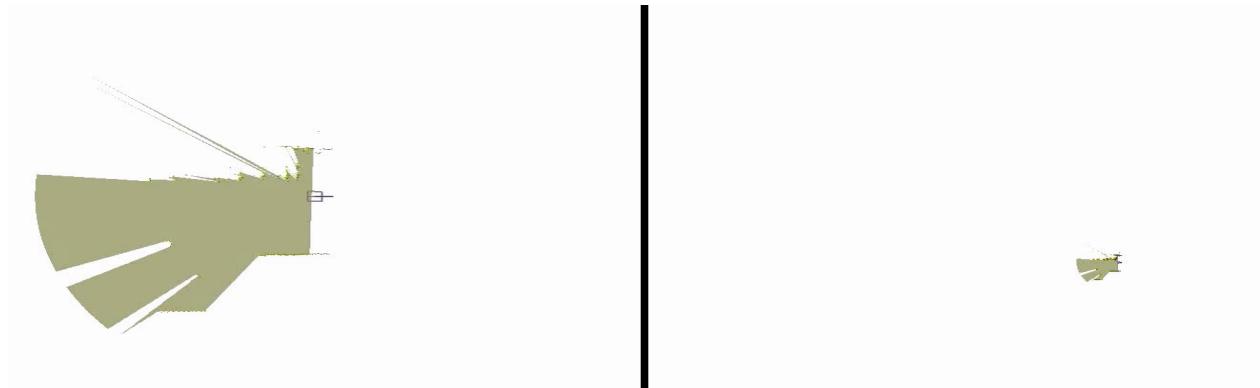
Track the position of a moving sensor by estimating the motion between the previous and the current views

Localization



- Determining the current position of a robot, given
 - The knowledge of the map
 - All sensor measurements up to the current time

SLAM



Determine the robot position AND the map, based on the sensor measurements

Tools

- Linux (Ubuntu)
- C/C++ (gcc)
- g2o (<https://github.com/RainerKuemmerle/g2o>)
- ROS (www.ros.org)
- Octave (sudo apt-get install octave)
- V-REP (<http://www.coppeliarobotics.com/index.html>)

Exam

Individual project

AND

Written exam

- theory
- exercise
- modeling

Oral question possible at discretion of the teachers

We will provide example exercises throughout the course, and publish exam samples on the course repo