Robotics

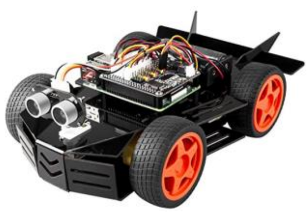
Erwin M. Bakker| LIACS Media Lab

28-2 2022

Universiteit Leiden

Bij ons leer je de wereld kennen

Organization and Overview



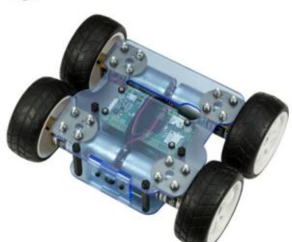
Period: February 7^{th} – May 23^{rd} 2022
Time: Monday 16.15 – 18.00
Place: Room 407 - 409
Lecturer: Erwin M. Bakker (erwin@liacs.nl)
Assistant: Hainan Yu (h.yu@liacs.leidenuniv.nl)

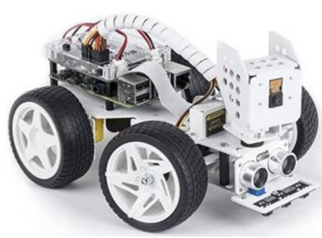
NB Register on Brightspace

Schedule:

7-2	Introduction and Overview
14-2	Locomotion and Inverse Kinematics
21-2	Robotics Sensors and Image Processing
28-2	SLAM + SLAM Workshop
7-3	**Mobile Robot Challenge Introduction**
14-3	**Project Proposals I (presentation by students)**
21-3	**Project Proposals II (presentation by students)**
28-3	Robotics Vision
4-4	Robotics Reinforcement Learning
11-4	Robotics Reinforcement Learning Workshop II
18-4	No Class (Eastern)
25-4	Project Progress I (presentations by students)
2-5	Project Progress II (presentations by students)
9-5	**Mobile Robot Challenge**
16-5	**Project Demos I**
23-5	**Project Demos II**

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/





Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

Universiteit Leiden. Bij ons leer je de wereld kennen

HomeWork II

Human Arm
m = 3D Space dof = 6
N = #links = 4
J = #joints = 3
2x Spherical: dof 2x 3
1x Revolute: dof 1x 1
$dof = m(N-1-J) + \sum f_i$
$= 6(4-1-3) + 2x3 + 1x1 = 7$

Robot Arm
m = 3D Space dof = 6
N = #links = 5
J = #joints = 4
4x Revolute: dof 4x 1
$dof = m(N-1-J) + \sum f_i$
$= 6(5-1-4) + 4x1 = 4$

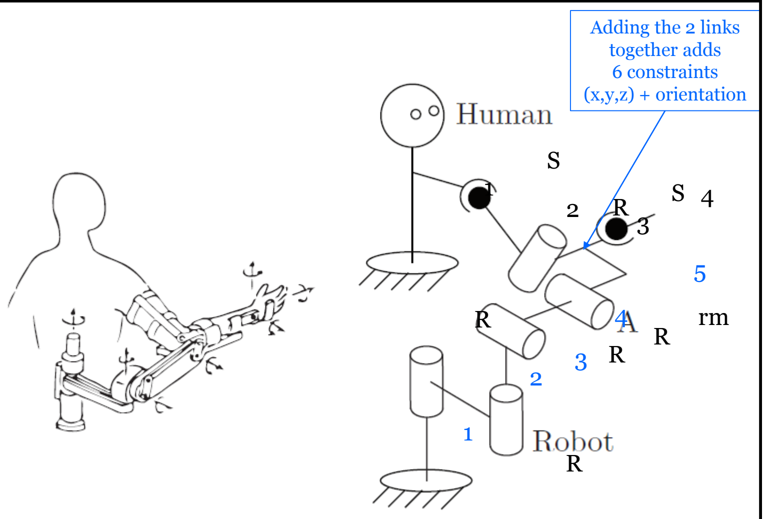


Figure 2.15: Robot used for human arm rehabilitation.

=> Human Arm connected to Robot Arm has 7 + 4 - 6 = 5 dof

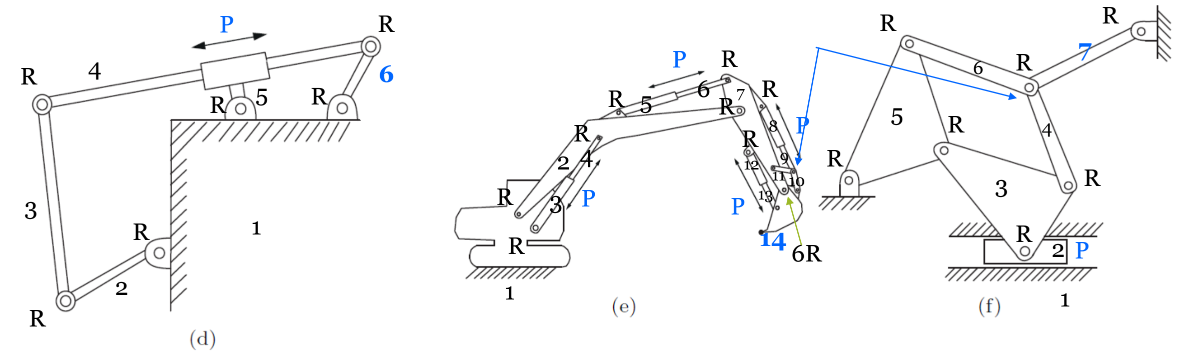


Figure 2.18: A first collection of planar mechanisms.

Fig. 1.18 d)
m = 2D Space dof = 3
N = #links = 6
J = #joints = 7
6x Revolute: dof 6x 1
1x Prismatic: dof 1x 1
$dof = m(N-1-J) + \sum f_i$
$= 3(6-1-7) + 6x1 + 1x1 = 1$

Fig. 1.18 e)
m = 2D Space dof = 3
N = #links = 14
J = #joints = 18
14x Revolute: dof 14x 1
4x Prismatic: dof 4x 1
$dof = m(N-1-J) + \sum f_i$
$= 3(14-1-18) + 14x1 + 4x1 = 3$

Fig. 1.18 f)
m = 2D Space dof = 3
N = #links = 7
J = #joints = 9
8x Revolute: dof 8x 1
1x Prismatic: dof 1x 1
$dof = m(N-1-J) + \sum f_i$
$= 3(7-1-9) + 8x1 + 1x1 = 0$

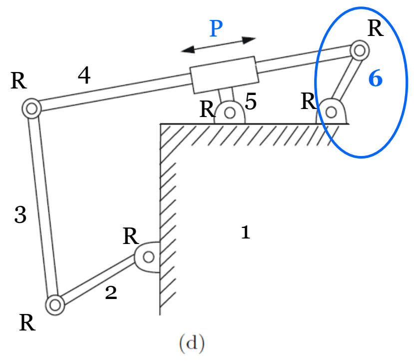


Fig. 1.18 d) without the encircled part:
m = 2D Space dof = 3
N = #links = 5
J = #joints = 5
4x Revolute: dof 4x 1
1x Prismatic: dof 1x 1
dof = m(N-1-J) + $\sum f_i$
= 3(5-1-5) + 4x1 + 1x1 = 2

- Encircled part has 2 dof.
- Joining the 2 structures adds 3 constraints.

=> dof of complete structure equals 2+2 – 3 = 1

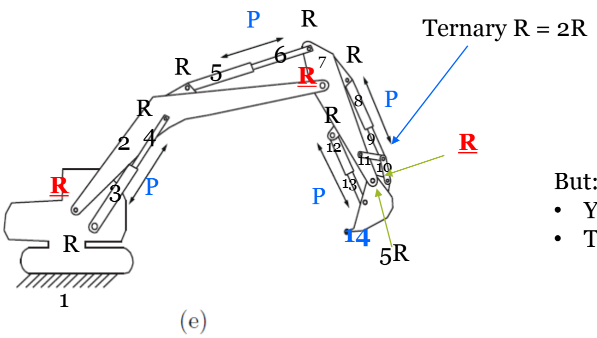


Figure 2.18: A first collection of planar mechanisms.

Fig. 1.18 e)
m = 2D Space dof = 3
N = #links = 14
J = #joints = 18
14x Revolute: dof 14x 1
4x Prismatic: dof 4x 1
dof = m(N-1-J) + $\sum f_i$
= 3(14-1-18) + 14x1 + 4x1 = 3

But: not taken any dependent joints into account:

- You can select **3 joints** such that all others have to follow.
- Thus it is clear that the dof of this structure is:

m = 2D Space dof = 3
N = #links = 4
J = #joints = 3
3x Revolute: dof 3x 1
dof = m(N-1-J) + $\sum f_i$
= 3(4-1-3) + 3x1 = 3

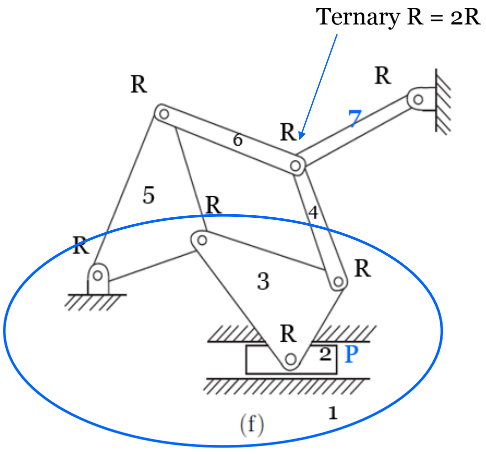


Fig. 1.18 f)
m = 2D Space dof = 3
N = #links = 7
J = #joints = 9
8x Revolute: dof 8x 1
1x Prismatic: dof 1x 1
dof = m(N-1-J) + $\sum f_i$
= 3(7-1-9) + 8x1 + 1x1 = 0

The encircled substructure has:

m = 2D Space dof = 3
N = #links = 4
J = #joints = 4
3x Revolute: dof 3x 1
1x Prismatic: dof 1x 1
dof = m(N-1-J) + $\sum f_i$
= 3(4-1-4) + 4x1 = 1 dof

- Adding the links 6 and 4 keeps this 1 dof + link 7 gives 2 dof
- Link 7 and its revolute has 1 dof.
- Combining the 2 structures gives 3 constraints => 2 + 1 - 3 = 0 dof

Fig. 2.21 d)
m = 3D Space dof = 6
N = #links = 22
J = #joints = 4 x 6 = 24
20x Revolute: dof 20x 1
4x Prismatic: dof 4x 1
dof = m(N-1-J) + $\sum f_i$
= 6(22-1-24) + 20x1 + 4x1 = 6

Fig. 2.21 d) with 1 leg on the ground
m = 3D Space dof = 6
N = #links = 24 + 1 — omit these for each leg
J = #joints = 4x 6
20x Revolute: dof 20x 1
4x Prismatic: dof 4x 1
dof = m(N-1-J) + $\sum f_i$
= 6(25-1-24) + 20x1 + 4x1 = 24

3 legs are constrained to the ground
=> dof= 24 – 3*6 = 6

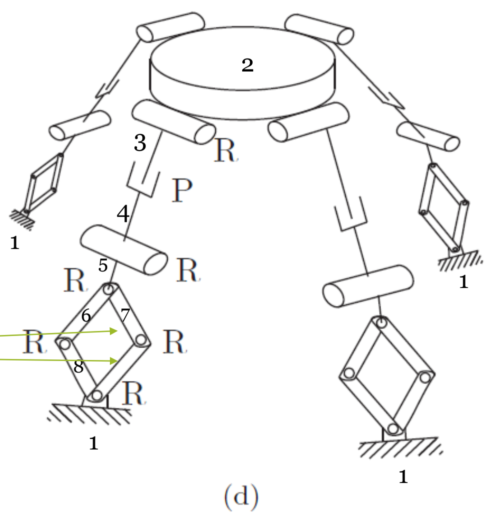
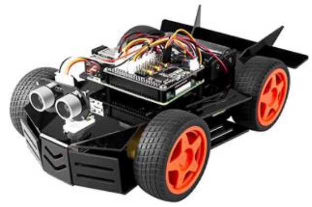


Figure 2.21: A second collection of spatial parallel mechanisms.

Organization and Overview



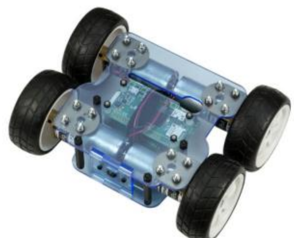
Period: February 7th – May 23rd 2022
Time: Monday 16.15 – 18.00
Place: Room 407 - 409
Lecturer: Erwin M. Bakker (erwin@liacs.nl)
Assistant: Hainan Yu (h.yu@liacs.leidenuniv.nl)

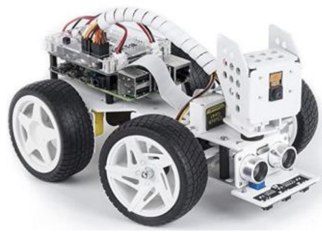
NB Register on Brightspace

Schedule:

7-2	Introduction and Overview
14-2	Locomotion and Inverse Kinematics
21-2	Robotics Sensors and Image Processing
28-2	SLAM + SLAM Workshop
7-3	**Mobile Robot Challenge Introduction**
14-3	**Project Proposals I (presentation by students)**
21-3	**Project Proposals II (presentation by students)**
28-3	Robotics Vision
4-4	Robotics Reinforcement Learning
11-4	Robotics Reinforcement Learning Workshop II
18-4	No Class (Eastern)
25-4	Project Progress I (presentations by students)
2-5	Project Progress II (presentations by students)
9-5	**Mobile Robot Challenge**
16-5	**Project Demos I**
23-5	**Project Demos II**

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/





Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

Universiteit Leiden. Bij ons leer je de wereld kennen

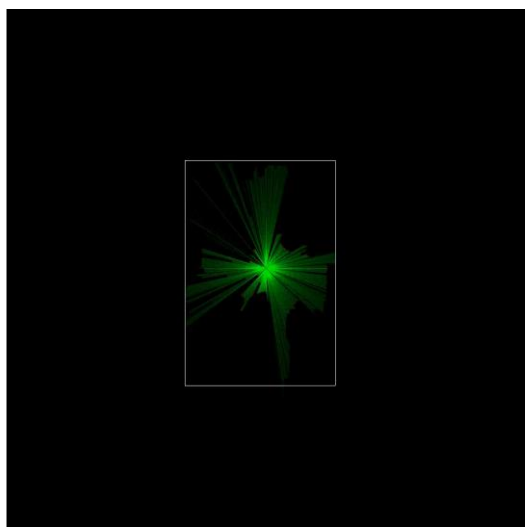
SLAM: Simultaneous Localization And Mapping

Mapping
- Use sensor data: US, LIDAR, Camera, Structured Light, etc. to make a map of the environment of the robot.

Localization
- Determine the pose of the robot relative to the map.
- Initial pose can be given: pose tracking problem
- No initial pose given: global localization problem

SLAM: do both Mapping and Localization at the same time.

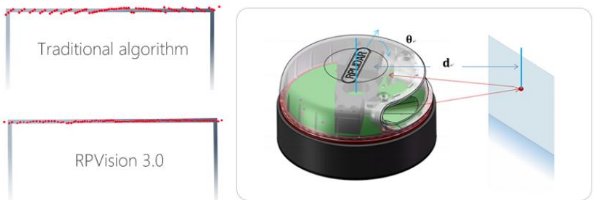
VSLAM: use camera (mono, stereo, multi-, depth, etc.) to solve SLAM



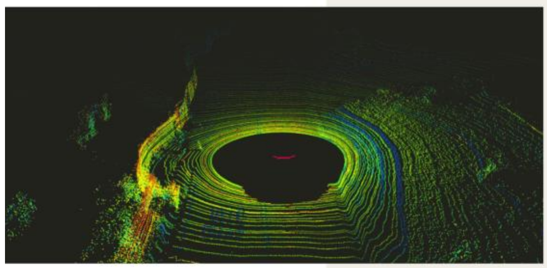
MonsterBorg SLAM by E. van der Zande

Universiteit Leiden. Bij ons leer je de wereld kennen

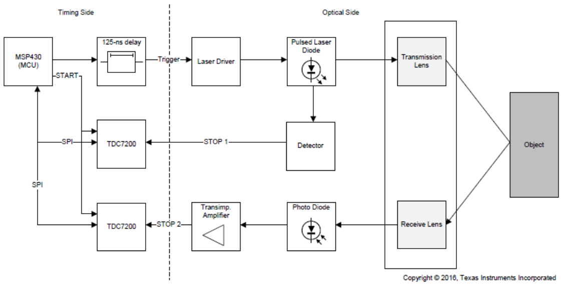
LIDAR



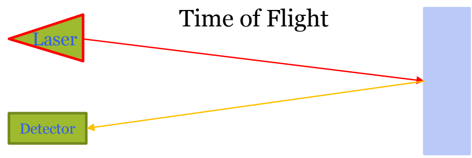
http://www.slamtec.com/en/lidar/A3



https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff



Texas Instruments LIDAR Pulsed Reference Design



- Speed of light ~ 3 x 10^8 m/s
- In 1 picosecond (= 10^{-12} sec) light travels ~ 3 x 10^{-4} m = 0.3 mm
- During 33 picoseconds light travels ~ 1cm

CES2019

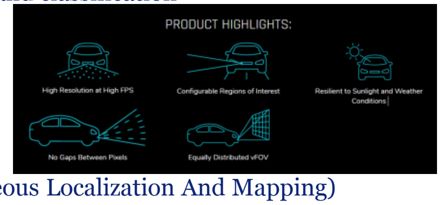
BMW Self Driving Car

InnovizOne Solid-state Lidar (goal: sub $1000 sensor)

- Angular resolution 0.1° x 0.1°
- FOV 120° x 25°
- 25 FPS
- Range 250m

Perception Capabilities

- Object detection and classification
- Lane detection
- Object Tracking
- SLAM



SLAM (Simultaneous Localization And Mapping)

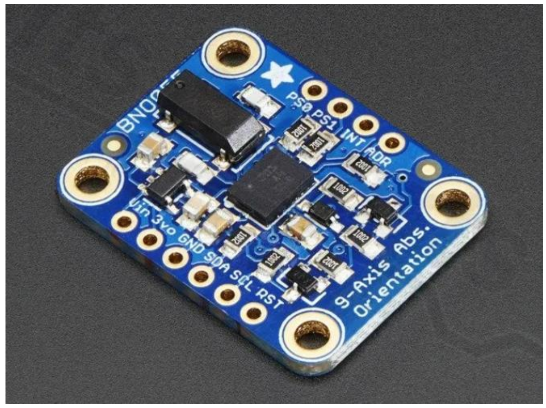
IMU, or Inertial Measurement Unit

IMU consist of Sensors for

- orientation, by measuring the earths magnetic field and a gyroscope.
- acceleration,
- (angular) velocity.

In SLAM

- Orientation is very useful for scan matching: a good initial guess.
- Acceleration is calculated by subtracting the earth gravity vector from the raw accelerometer data.
- Velocity is calculated by time integration of the acceleration (tends to accumulate large errors)

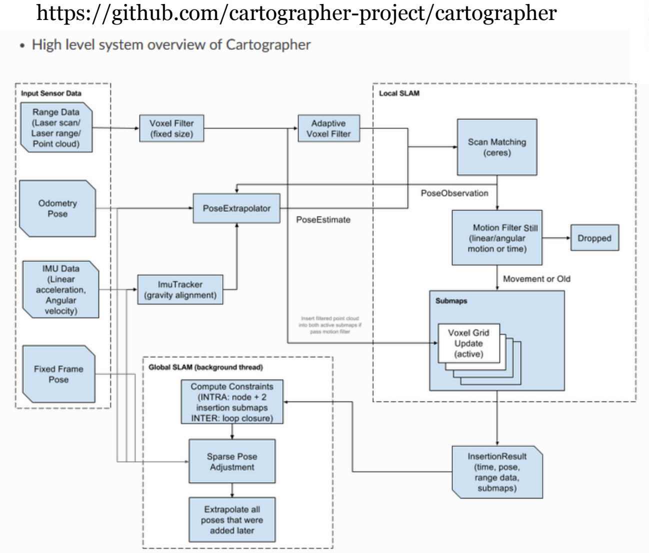


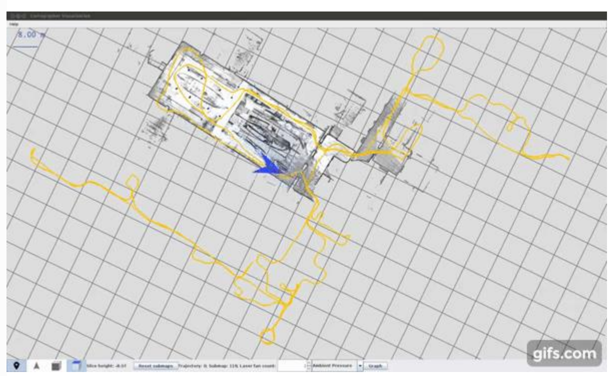
IMU Bosch BNO055 Sampling Frequencies:

- Accelerometer: 1 KHz
- Gyroscope: 523 Hz
- Magnetometer: 30Hz

Cartographer

https://github.com/cartographer-project/cartographer



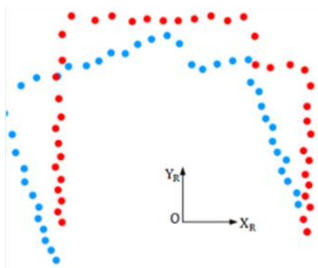


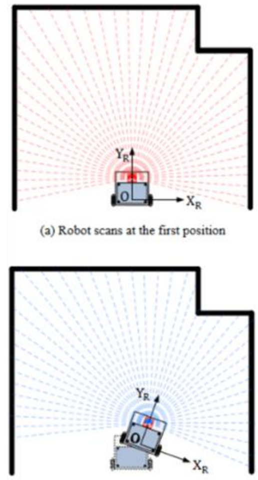
Cartographer gives real-time simultaneous localization and mapping (SLAM) in 2D and 3D.

2D and 3D Lidar SLAM

Front-end:

- Matching new measurements against the current belief of the environment
- Measurements:
  - typically point clouds of lidar data
  - Movement and heading data: IMU, wheelencoders, etc.
- Filtering of the data
  - Missing or invalid data (zero length rays)
  - Noisy data: measured too far away
  - Distorted data: e.g., as a result of robot acceleration
- Transform
  - Scan matching tries to find a rigid body transform: translation and rotation
  - IMU can give heading information: more reliable than deriving it from scan matching





Riadh Dhaoui, Ruhr-Universität Bochum

2D and 3D Lidar SLAM: Scan Matching

Point to Point Matching

- Between the 2 scans point associations are found
- The translation from points to associated points determines a rigid body transform
- E.g found by Singular Value Decomposition
- Fails in case of degenerate clouds (coplanar)
- Point association is difficult to obtain

Iterative closest point (ICP) determination

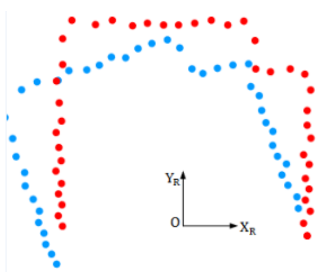
- Compute closest points
- Compute and apply the transformation
- Check if it is good enough, otherwise repeat

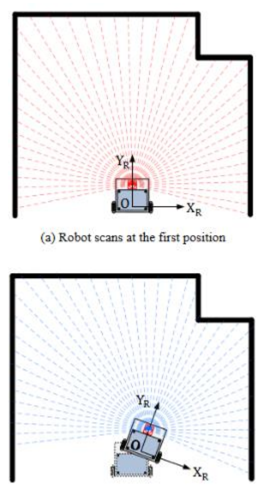
Tangent Matching

- In each of the 2 scans find Tangent lines between neighboring points
- Instead of point matching now line matching between 2 scans
- Movement and heading data: IMU, wheelencoders, etc.

Point to Grid Map

- Etc.





Riadh Dhaoui, Ruhr-Universität Bochum

SLAM

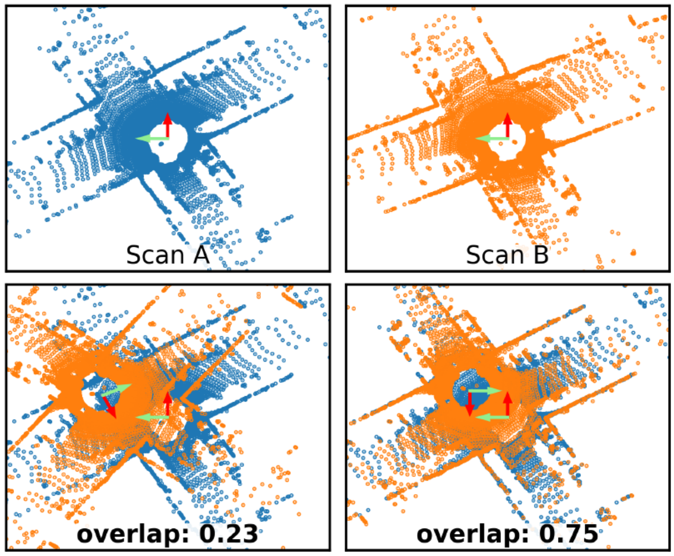
1) Pose estimation relative to recent poses: incremental scan matching, odometry
2) Loop closing: correcting accumulated drift, maintaining consistency between measurements

Loop closure detection:

- Estimate lidar point clouds overlap (note: also used in photogrammetry):
  - Yaw transformation estimate
  - Score, e.g., using Iterative Closest Point
- Can also be used for global scan matching.

OverlapNet a DNN method that does an estimate without transformation guessing, uses spherical projection of 3D Lidar data.

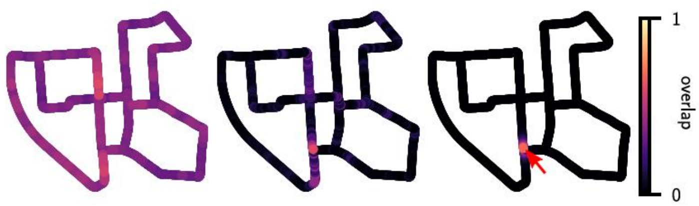
Performance on Kitty-odometry dataset.



Example: Scan A and B overlap detection: scored after transformation.

SLAM: Loop Closing OverlapNet

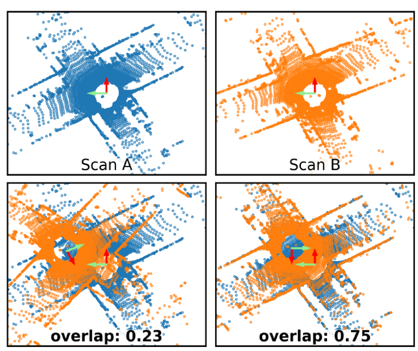
1) Pose estimation relative to recent poses: incremental scan matching, odometry
2) Loop closing: correcting accumulated drift



(a) Exhaustive evaluation of Eq. (3) (b) OverlapNet estimates (c) Ground truth

Loop closure detection:

- In general overlap scores based on point-cloud differences (see Eq. 3 of X Chen et al., May 2021) will be high at many locations of the map.
- OverlapNet a DNN method that does an estimate without transformation guessing, uses spherical projection of 3D Lidar data.



Example: Scan A and B overlap detection: scored after transformation.

Universiteit Leiden. Bij ons leer je de wereld kennen

OverlapNet

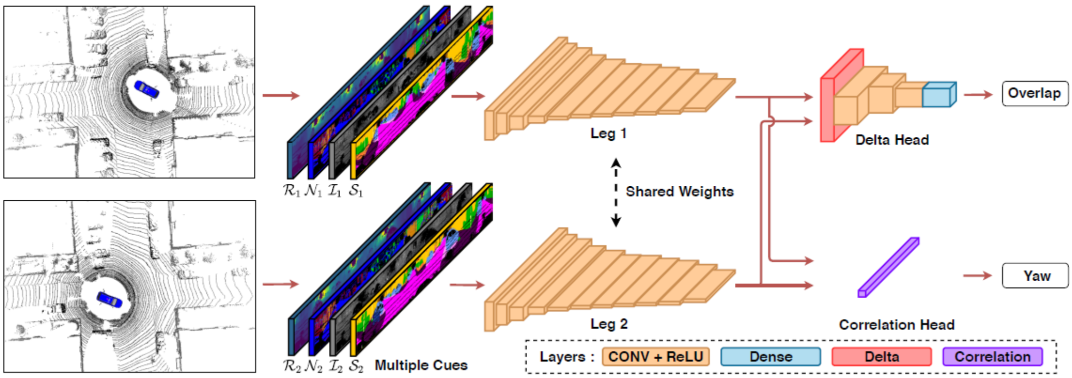


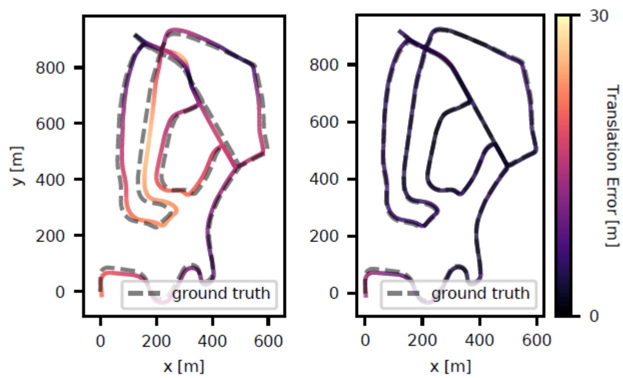
Fig. 3: Pipeline overview of our proposed approach. The left-hand side shows the preprocessing of the input data which exploits multiple cues generated from a single LiDAR scan, including range $\mathcal{R}$, normal $\mathcal{N}$, intensity $\mathcal{I}$, and semantic class probability $\mathcal{S}$ information. The right-hand side shows the proposed OverlapNet which consists of two legs sharing weights and the two heads use the same pair of feature volumes generated by the two legs. The outputs are the overlap and relative yaw angle between two LiDAR scans.

OverlapNet

X. Chen et al., OverlapNet: Loop Closing for Lidar-based SLAM, May 2021.

TABLE II: Comparison with state of the art.

Dataset	Approach	AUC	F1 score
KITTI	Histogram [26]	0.83	0.83
	M2DP [14]	0.83	0.87
	SuMa [2]	-	0.85
	Ours (AllChannel, TwoHeads)	**0.87**	**0.88**
Ford Campus	Histogram [26]	0.84	0.83
	M2DP [14]	0.84	**0.85**
	SuMa [2]	-	0.33
	Ours (GeoOnly)	**0.85**	0.84



(a) SuMa (b) Ours_CovNearestOfTop10

Fig. 6: Qualitative result on KITTI sequence 02.

Universiteit Leiden. Bij ons leer je de wereld kennen

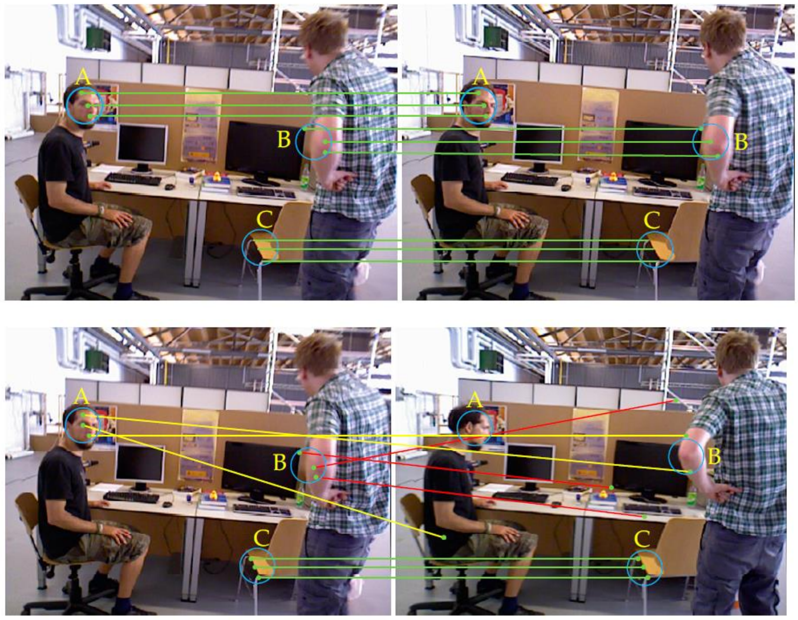
Visual SLAM

VSLAM: use camera (mono, stereo, multi-, depth, etc.) to solve SLAM

- Readily available on mobile platforms: drones, cars, mobile phones, AR/VR Glasses
- DMS-SLAM, OpenVSLAM, ORBSLAM

DMS-SLAM:

- pose tracking, closed-loop detection and relocalization based on static 3D map points of the local map
- supports monocular, stereo and RGB-D visual sensors in dynamic scenes.



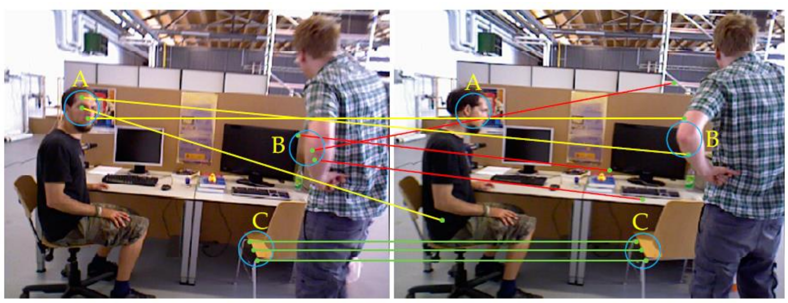
A and B partial Motion, C static => A and B matching errors, C still correct.

Universiteit Leiden. Bij ons leer je de wereld kennen

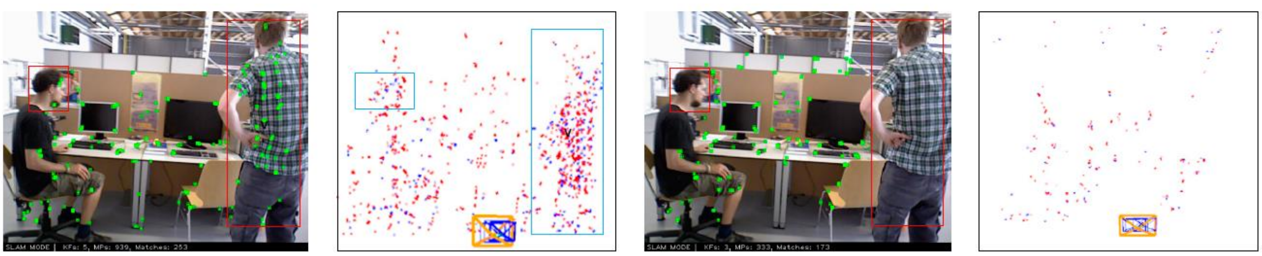
Visual SLAM

VSLAM: use camera (mono, stereo, multi-, depth, etc.) to solve SLAM

- DMS-SLAM, OpenVSLAM, ORBSLAM



A and B partial Motion, C static => A and B matching errors, C still correct.



Red-boxes moving

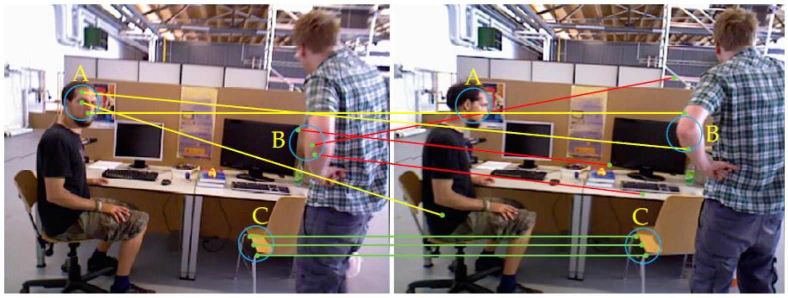
(a) (b) (c) (d)

ORB-SLAM2. DMS-SLAM

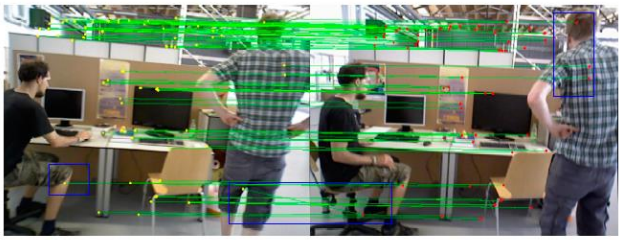
Visual SLAM

VSLAM: use camera (mono, stereo, multi-, depth, etc.) to solve SLAM

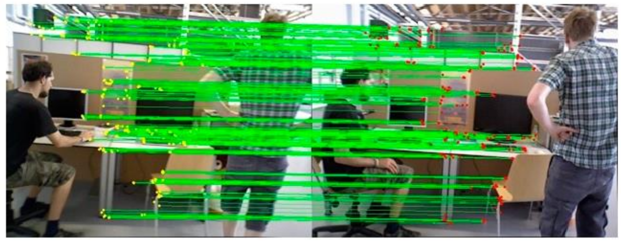
- DMS-SLAM, OpenVSLAM, ORBSLAM



A and B partial Motion, C static => A and B matching errors, C still correct.



(a) ORB-SLAM2



(b) DMS-SLAM

Blue-boxes wrong feature matching point pairs.

(a) ORB-SLAM2

(b) DMS-SLAM

Sequence 01

(c) ORB-SLAM2

(d) DMS-SLAM

Sequence 03

(e) ORB-SLAM2

(f) DMS-SLAM

Sequence 09

Figure 15. The pose tracking experiments of ORB-SLAM2 and DMS-SLAM on the 01, 03, 09 sequences in the KITTI dataset. The rectangular box represents the moving object in the scene, and the others are static areas.

OpenVSLAM

A Versatile Visual SLAM Framework. (S. Sumikura et al., 2019)

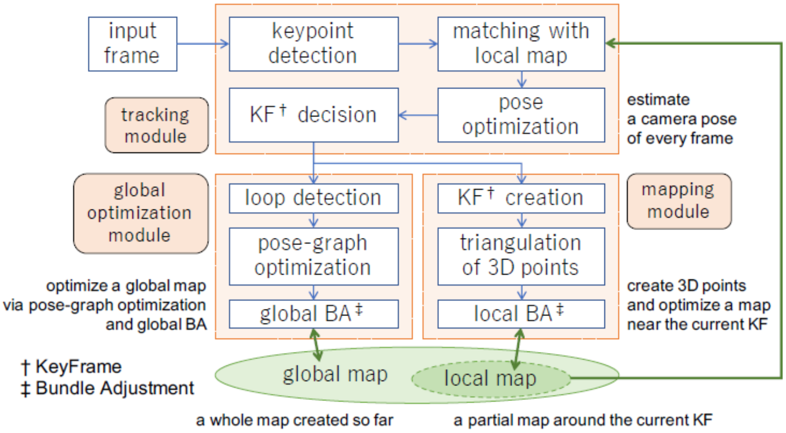
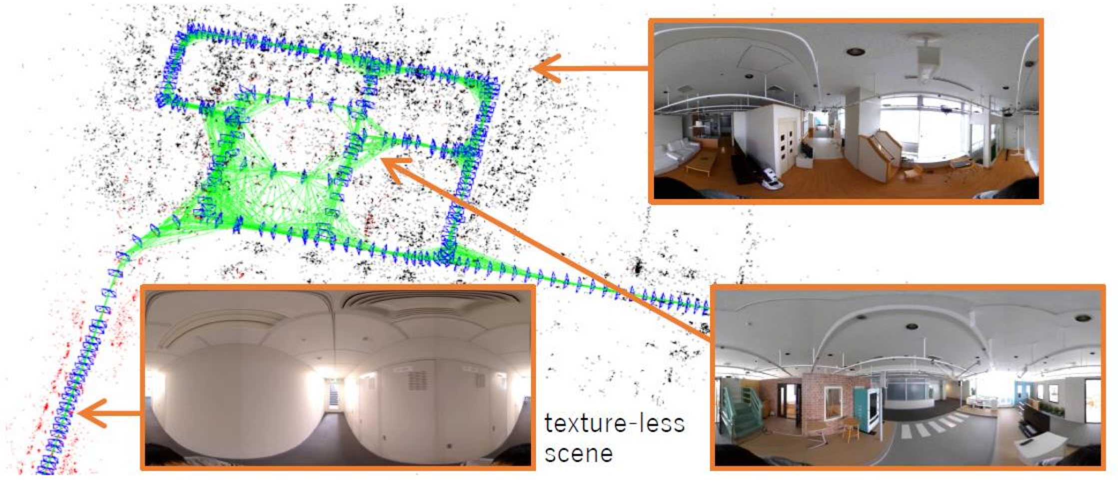
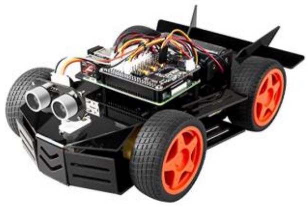


Figure 2: Main modules of OpenVSLAM: tracking, mapping, and global optimization modules.

OpenVSLAM



Organization and Overview



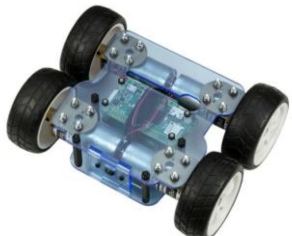
Period: February 7th – May 23rd 2022
Time: Monday 16.15 – 18.00
Place: Room 407 - 409
Lecturer: Erwin M. Bakker (erwin@liacs.nl)
Assistant: Hainan Yu (h.yu@liacs.leidenuniv.nl)

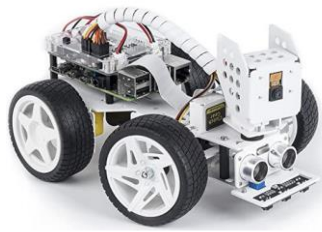
NB Register on Brightspace

Schedule:

7-2	Introduction and Overview
14-2	Locomotion and Inverse Kinematics
21-2	Robotics Sensors and Image Processing
28-2	SLAM + SLAM Workshop
7-3	**Mobile Robot Challenge Introduction**
14-3	**Project Proposals I (presentation by students)**
21-3	**Project Proposals II (presentation by students)**
28-3	Robotics Vision
4-4	Robotics Reinforcement Learning
11-4	Robotics Reinforcement Learning Workshop II
18-4	No Class (Eastern)
25-4	Project Progress I (presentations by students)
2-5	Project Progress II (presentations by students)
9-5	**Mobile Robot Challenge**
16-5	**Project Demos I**
23-5	**Project Demos II**

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/





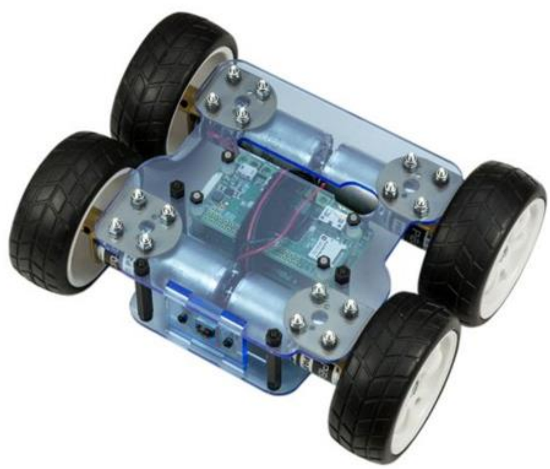
Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

Universiteit Leiden. Bij ons leer je de wereld kennen

Yeti Borg Introduction

YetiBorg
- Raspberry Pi Zero W
- 4 motors
- 1 Camera
- No odometry

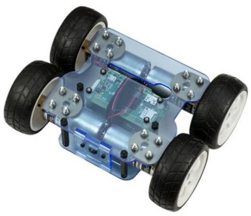
- Programming and control through SSH
- Python and OpenCV
- No live camera footage from the YetiBorg
- VNC can be used to get live camera footage

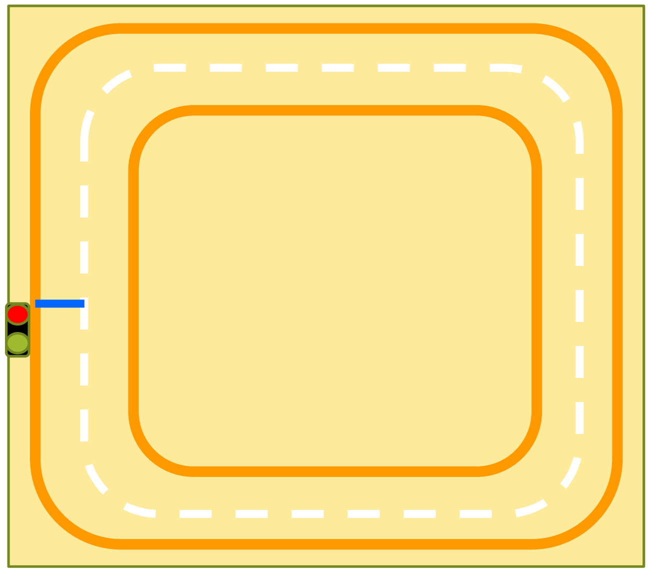


Universiteit Leiden. Bij ons leer je de wereld kennen

YetiBorg URBAN Challenge (previously: 2021)

Urban Challenge
- Keep right
- Traffic Lights
- Traffic Signs
- Obstacles



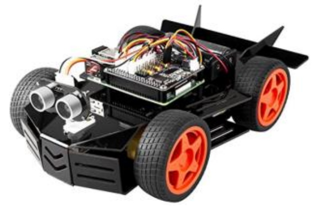


Qualification Track
- Keep Right
- Traffic Lights
- Stop as close as possible before the blue line

Mobile Robot Challenge Teams

- Form Mobile Robot Challenge Teams of 4-5 members.
- Determine a Team Name and 1 contact person.
- Further details will follow on Brightspace.

Organization and Overview



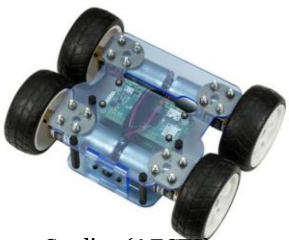
Period: February 7th – May 23rd 2022
Time: Monday 16.15 – 18.00
Place: Room 407 - 409
Lecturer: Erwin M. Bakker (erwin@liacs.nl)
Assistant: Hainan Yu (h.yu@liacs.leidenuniv.nl)

NB Register on Brightspace

Schedule:

7-2	Introduction and Overview
14-2	Locomotion and Inverse Kinematics
21-2	Robotics Sensors and Image Processing
28-2	SLAM + SLAM Workshop
7-3	**Mobile Robot Challenge Introduction**
14-3	**Project Proposals I (presentation by students)**
21-3	**Project Proposals II (presentation by students)**
28-3	Robotics Vision
4-4	Robotics Reinforcement Learning
11-4	Robotics Reinforcement Learning Workshop II
18-4	No Class (Eastern)
25-4	Project Progress I (presentations by students)
2-5	Project Progress II (presentations by students)
9-5	**Mobile Robot Challenge**
16-5	**Project Demos I**
23-5	**Project Demos II**

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/



Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

Robotics Project Proposals Presentations

Monday 14-3 2022 and 21-3 2022

Present your Robotics Project Proposal during a **5 minute (max)** talk. Clearly state the title of your project, the team members, your goals, how you will pursue them, what are the challenges and what at least can and should be delivered on the demo days: May 16th 2022 and May 23rd 2022.

Note: Groups of 1-5 members are allowed.

The presentation should contain slides for:
1. Title and group members.
2. Goal of the project.
3. How will you pursue these goals.
4. What are the challenges.
5. What at least can and should be delivered on the demo days: May 16th 2022 and May 23rd 2022.

The LIACS Media Lab can support your project with some materials for your project. Please clearly state any materials that you would need for your proposal.
Please note that these materials are limited so project goals may need to be adjusted accordingly.

Each presentation will be followed by a short class discussion.

Universiteit Leiden. Bij ons leer je de wereld kennen

Previous Projects

1. Evolutionary Locomotion
2. Nao plays Tic-Tac-Toe
3. Slam Robot Project.
4. Dolphin Drone: Drone Recognition and Maneuvering with Hoops.
5. Delivery Drone.
6. Programming a NAO to play a tune using a xylophone.
7. Floor mapping with Swarm Robotics
8. Tootballing Yetiborg
9. Cat Flap Opening Based on Audio/Video/RFID
10. DrawBot
11. Traffic coordination (simulation).
12. Plane filling curves (simulation).

1. AimBot
2. Artificial Muscles
3. Ball Tracking Car
4. BorrelBot
5. Fetch Bot
6. Floor Mapping Robot
7. Gesture Control Pachenko
8. Hexapod
9. Nao Pose
10. Position Estimation
11. Race Car Training
12. Face Touch

References

SLAM

1. M. Sualeh, G.-W. Kim, Simultaneous Localization and Mapping in the Epoch of Semantics: A Survey, International Journal of Control, Automation and Systems 17(3) (2019) 729-742.
2. W. Hess, D. Kohler, H. Rapp, D. Andor, Real-Time Loop Closure in 2D LIDAR SLAM, Published in: 2016 IEEE International Conference on Robotics and Automation (ICRA).
3. X. Chen et al., OverlapNet: Loop Closing for Lidar-based SLAM, May 2021.
4. E.R. van der Zande, An Introduction to 2D Lidar SLAM, December 2021.
5. S. Sumikura et al., OpenVSLAM: A Versatile Visual SLAM Framework, 2019.
6. G. Liu et al. DMS-SLAM: A General Visual SLAM System for Dynamic Scenes with Multiple Sensors, Sensors, 2019.

Yetiborg Challenge

1. https://navoshta.com/detecting-road-features/ by Alex Staravoitau
2. OpenCV.org

Universiteit Leiden. Bij ons leer je de wereld kennen