# Robotics

Erwin M. Bakker| LIACS Media Lab

1-3 2021

Universiteit Leiden

Bij ons leer je de wereld kennen

# Organization and Overview

**Period:** February 1st – May 10th 2021
**Time:** Tuesday 16.15 – 18.00
**Place:** https://smart.newrow.com/#/room/qba-943
**Lecturer:** Dr Erwin M. Bakker ( erwin@liacs.nl )
**Assistant:** Erqian Tang

NB Register on Brightspace

**Schedule:**

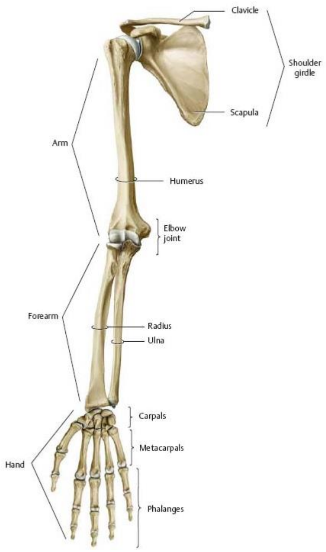
| | |
|---|---|
| 1-2 | Introduction and Overview |
| 8-2 | No Class (Dies) |
| 15-2 | Locomotion and Inverse Kinematics |
| 22-2 | Robotics Sensors and Image Processing |
| 1-3 | **Yetiborg Introduction + SLAM Workshop I** |
| 8-3 | Project Proposals (presentation by students) |
| 15-3 | Robotics Vision |
| 22-3 | Robotics Reinforcement Learning |
| 29-3 | **Yetiborg Qualification + Robotics Reinforcement Learning Workshop II** |
| 5-4 | No Class (Eastern) |
| 12-4 | Project Progress (presentations by students) |
| 19-4 | **Yetiborg Challenge** |
| 26-4 | Project Team Meetings |
| 3-5 | Project Team Meetings |
| 10-5 | **Online Project Demos** |

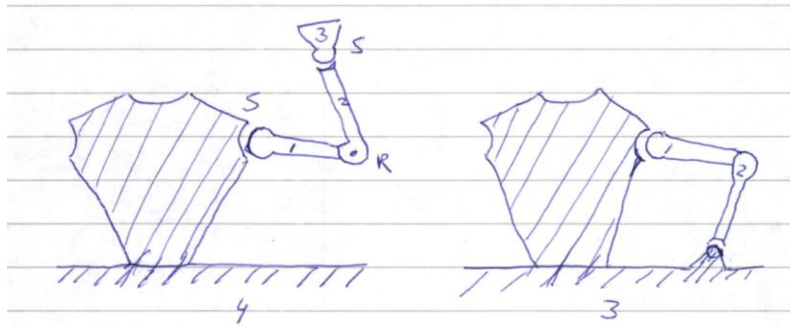
Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/

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.

Universiteit Leiden. Bij ons leer je de wereld kennen

# HomeWork II

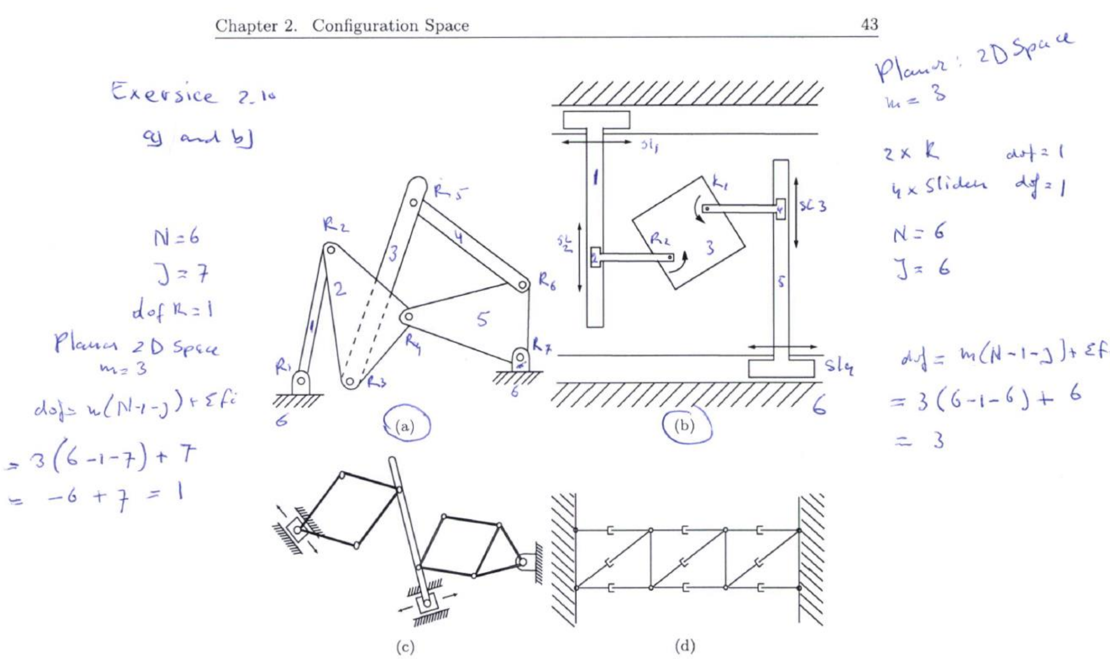



m = 3D Space dof = 6
N = #links = 4
J = #joints = 3
2x Spherical: dof 2x 3
1x Revolute: dof 1x 1
dof = m(N-1-J) + $\sum f_i$
= 6(4-1-3) + 2x3 + 1x1 = 7

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






Universiteit Leiden. Bij ons leer je de wereld kennen

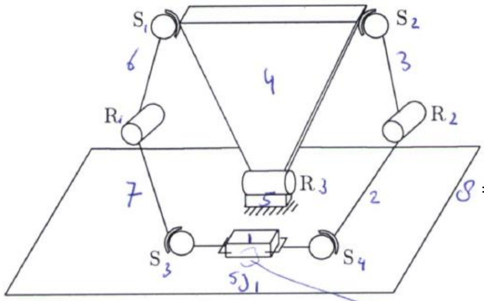



Figure 2.28: Dual-arm robot.

m = 6

#Links N = 7

#Joints:

| | | dof |
|---|---|---|
| 4 x S(3) | | 12 |
| 3 x R(1) | | 3 |
| 1 x SJ(2) | | ~~2~~ 3 |
| J = 8 | | 18 |

SJ: slide joint. dof = ~~2~~ 3

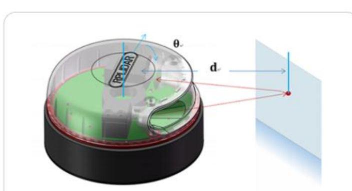
$$\Rightarrow dof = m(N-1-J) + \sum f_i$$

$$= 6(7-1-8) + 18$$

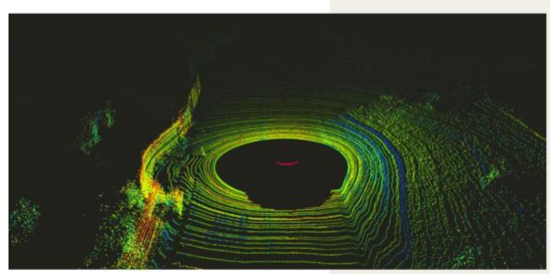
$$= -12 + 18 = 6$$

Universiteit Leiden. Bij ons leer je de wereld kennen

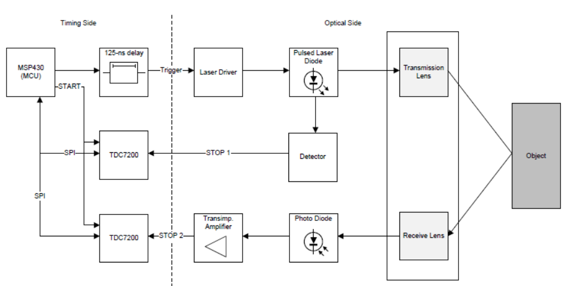
# LIDAR

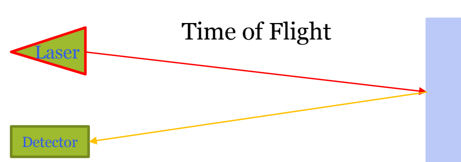

http://www.slamtec.com/en/lidar/A3



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

Universiteit Leiden. Bij ons leer je de wereld kennen

# CES2019

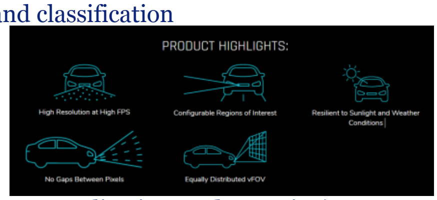
BMW Self Driving Car


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



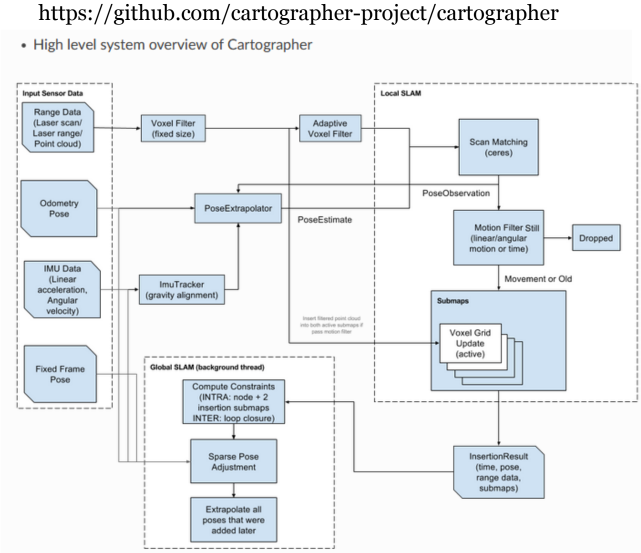


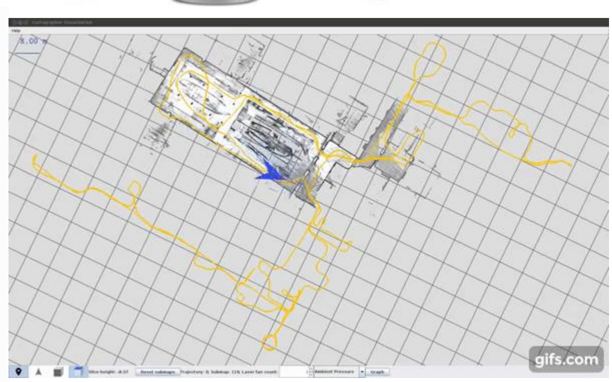
SLAM (Simultaneous Localization And Mapping)



# Cartographer

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

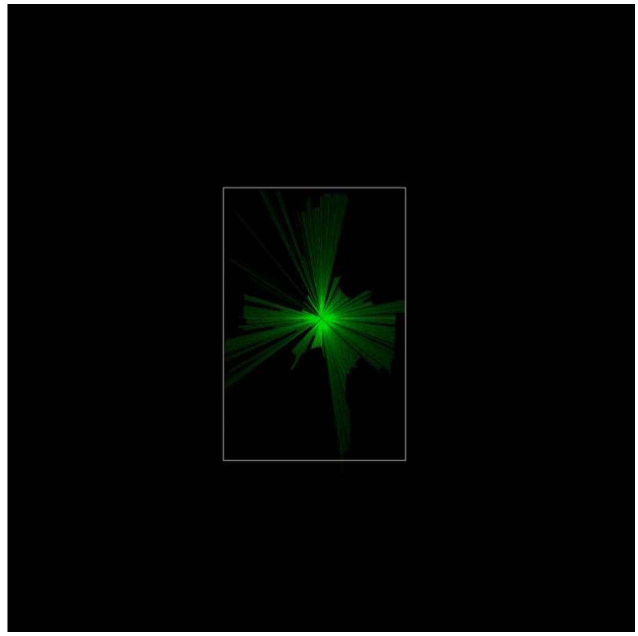



Cartographer gives real-time simultaneous localization and mapping (SLAM) in 2D and 3D.

Universiteit Leiden. Bij ons leer je de wereld kennen

# MonsterBorg SLAM
## E. van der Zande



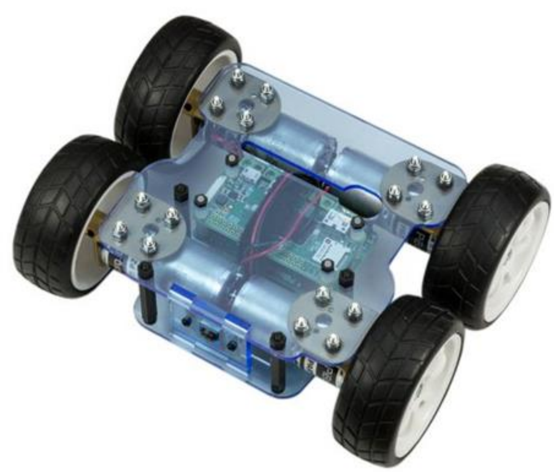


# Yeti Borg Introduction

**YetiBorg**

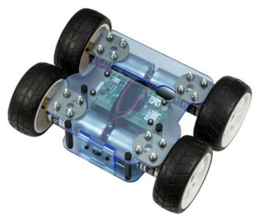
- Raspberry Pi Zero W
- 4 motors
- 1 Camera
- No odometry

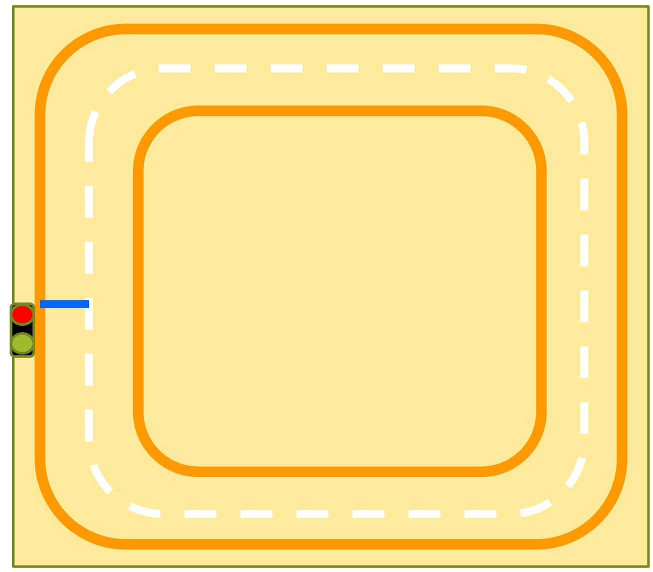
- Programming and control through SSH
- Python and OpenCV
- No live camera footage from the YetiBorg
- Live view of environment through Kaltura



# YetiBorg URBAN Challenge

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





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



# YetiBorg Urban Challenge Teams

- Form YetiBorg Urban Challenge Teams of max 4-5 members.
- Determine a Team Name and 1 contact person.
- Each contact person sends the Team Name and names and emails of the members to erwin@liacs.nl before March 7th 2021, 23.59h.

Universiteit Leiden. Bij ons leer je de wereld kennen

# Organization and Overview

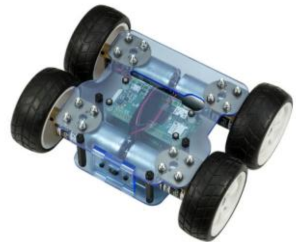
**Period:** February 1st – May 10th 2021
**Time:** Tuesday 16.15 – 18.00
**Place:** https://smart.newrow.com/#/room/qba-943
**Lecturer:** Dr Erwin M. Bakker ( erwin@liacs.nl )
**Assistant:** Erqian Tang

NB Register on Brightspace

**Schedule:**

| | |
|---|---|
| 1-2 | Introduction and Overview |
| 8-2 | No Class (Dies) |
| 15-2 | Locomotion and Inverse Kinematics |
| 22-2 | Robotics Sensors and Image Processing |
| 1-3 | **Yetiborg Introduction + SLAM Workshop I** |
| 8-3 | **Project Proposals (presentation by students)** |
| 15-3 | Robotics Vision |
| 22-3 | Robotics Reinforcement Learning |
| 29-3 | **Yetiborg Qualification + Robotics Reinforcement Learning Workshop II** |
| 5-4 | No Class (Eastern) |
| 12-4 | Project Progress (presentations by students) |
| 19-4 | **Yetiborg Challenge** |
| 26-4 | Project Team Meetings |
| 3-5 | Project Team Meetings |
| 10-5 | **Online Project Demos** |

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.

Universiteit Leiden. Bij ons leer je de wereld kennen

# Robotics Project Proposals Presentations

## Monday 8-3 2021

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 day on May 10th 2021.

Note: Groups of 4-5 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 day on May 10th 2021.

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. 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



# Robotics Discussion Session

## Wednesday 3-3 at 13.00
## Robotics Kaltura Room

During this session we discuss practical aspects and any further questions of robotics in an informal and interactive setting.

Universiteit Leiden. Bij ons leer je de wereld kennen

# 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).

Yetiborg Challenge

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



# Robotics

Universiteit Leiden

# Image Processing using OpenCV

**Core module:** the basic building blocks of this library for manipulating the images on a pixel level.

**Imgproc module:** the image processing (manipulation) functions inside OpenCV.

High Level GUI and Media (highgui module)

Image Input and Output (imgcodecs module)

Video Input and Output (videoio module)

**Camera calibration** and 3D reconstruction (calib3d module)

**2D Features framework** (feature2d module): feature points detectors, descriptors and matching framework found inside OpenCV.

Video analysis (video module) algorithms usable on your video streams like motion extraction, feature tracking and foreground extractions.

**Object Detection** (objdetect module) face detectors, etc.

**Deep Neural Networks** (dnn module)

**Machine Learning** (ml module) machine learning classes for statistical classification, regression and clustering of data.

Graph API (gapi module)

Computational photography (photo module) for advanced photo processing.

Images stitching (stitching module) create photo panoramas and more with OpenCV stitching pipeline.

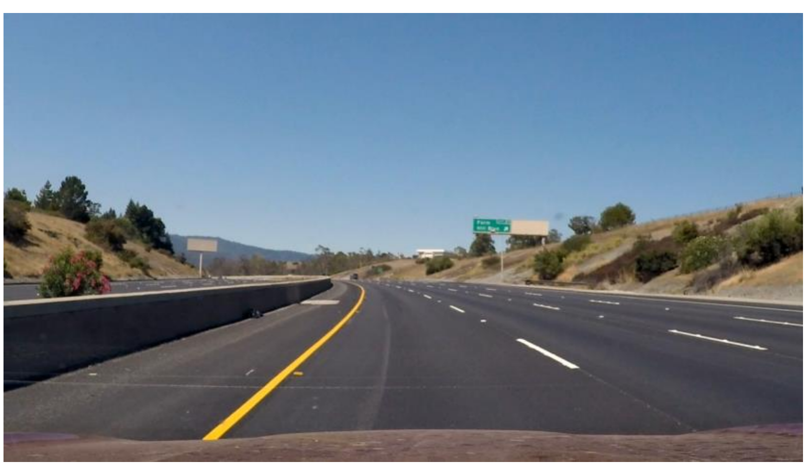
GPU-Accelerated Computer Vision (cuda module); OpenCV iOS:

Universiteit Leiden. Bij ons leer je de wereld kennen

# Lane Tracking

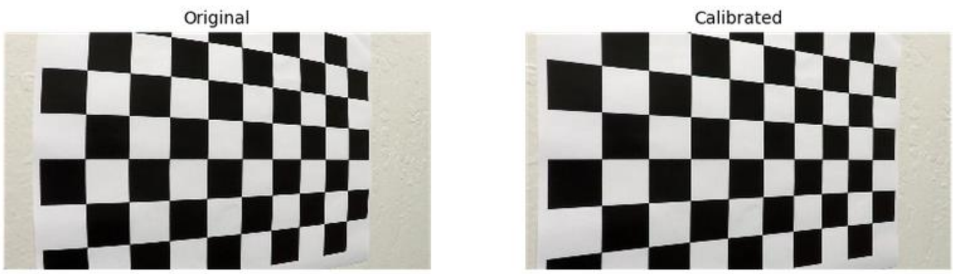
Some example project for detecting road features using OpenCV:

https://navoshta.com/detecting-road-features/ by Alex Staravoitau



Universiteit Leiden. Bij ons leer je de wereld kennen

# Processing Pipeline: Camera Calibration




Original vs. calibrated images

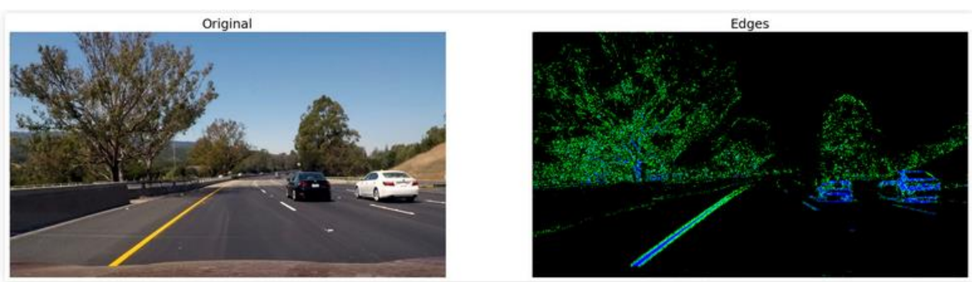
... cv2.findChessboardCorners(image, (9, 6), None) // Inner corners 9x6

... cv2.calibrateCamera( pattern_points, image_points, (image.shape[1], image.shape[0]), None, None)

corrected_image = cv2.undistort(image, self.camera_matrix, self.dist_coefficients, None, self.camera_matrix)



# Processing Pipeline: Edge Detection




Original vs. highlighted edges

Gradient Absolute Values, Ranges within certain magnitudes, Gradient Directions

- Sobel Operator (using a convolutional Kernel)

Color Ranges

- HLS Color Space: Hue, Lightness, and Saturation (for road detection, etc.)

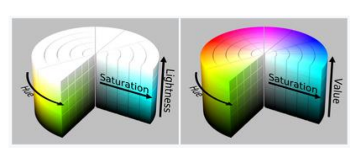



Fig. 2a. HSL cylinder. Fig. 2b. HSV cylinder.

# Processing Pipeline: Perspective Transformation


Boundary detection pipeline

Left: The *original* image after the camera calibration and perspective transform.

Right: After edge detection with edges highlighted in **green** and **blue**.

Scanning windows boundaries for areas with pixel that may belong to lines are highlighted in **yellow**,

A second order polynomial approximation of the collected points in **red**.



# Lane and Vehicle Tracking