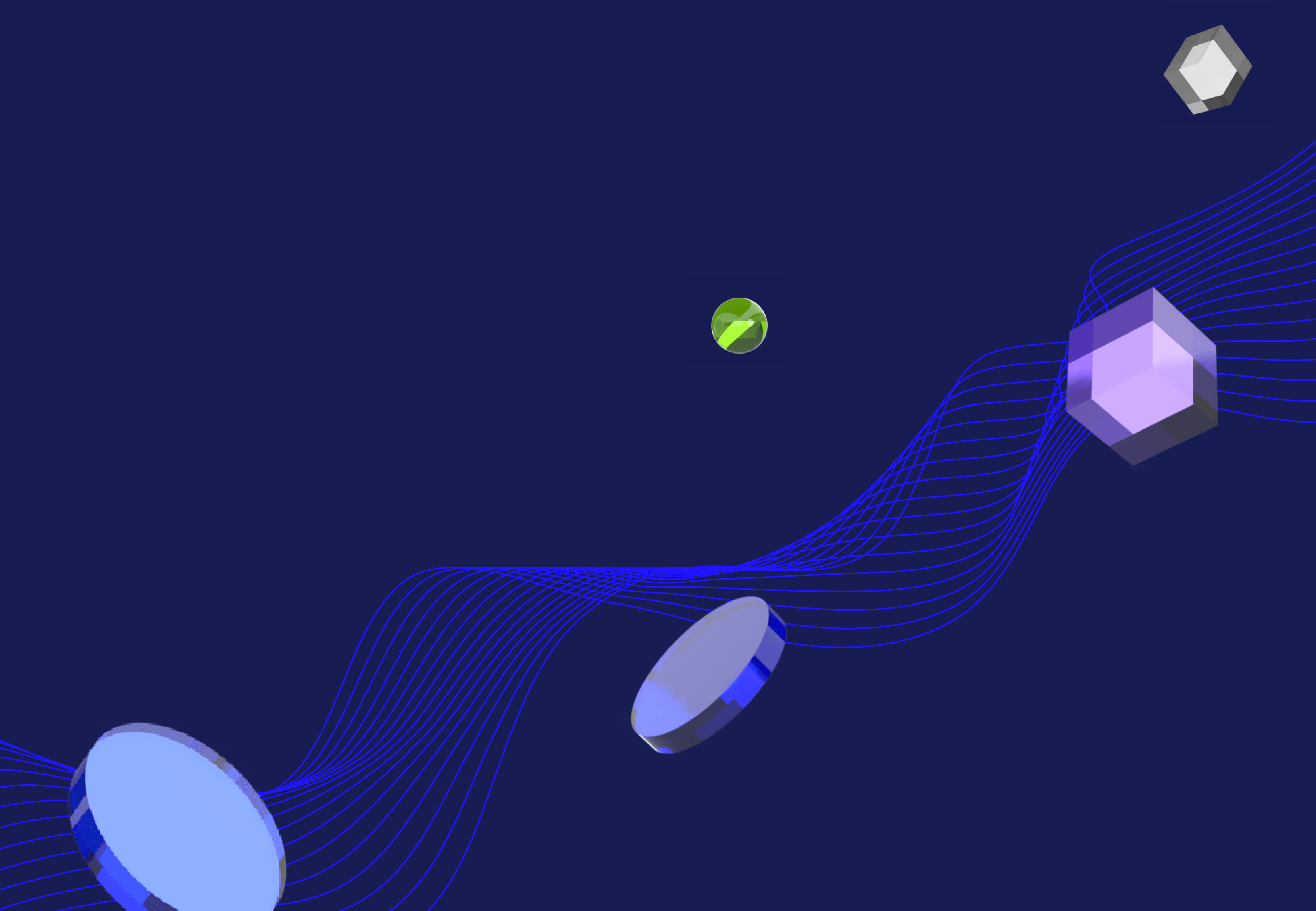
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF AUTONOMOUS SYSTEMS

Self-Driving Car Engineer

Nanodegree Program Syllabus



Overview

This program will cover the techniques that power self-driving cars across the full stack of a vehicle's autonomous capabilities. To begin, learners will find out how to apply computer vision and deep learning toward perception problems like lane finding and classifying traffic signs, as well as a full end-to-end algorithm for driving with behavioral cloning. Learners will also learn how to track objects from radar and lidar data with sensor fusion. From there, they'll learn and implement the concepts behind localization, path planning and control, making sure their vehicle knows where it is in the environment and how to navigate through it.

Learning Objectives

A graduate of this program will be able to:

- Train a neural network to classify images of cars, cyclists and pedestrians.
- Correspond objects in camera images to objects in lidar point clouds.
- Train a model to predict an object's trajectory.
- Predict the behavior of other agents in the environment.
- Program a finite state machine to plan a vehicle's motion in an urban environment.
- Implement a Proportional Integral Derivative (PID) controller to execute a non-linear trajectory using steering, acceleration, and brake.

Built in collaboration with:

Mercedes-Benz

WAYMO

Program information

Estimated Time

5 months at 10hrs/week*

Skill Level

Advanced

Prerequisites

A well-prepared learner should have knowledge of:

- Python
- C++
- Linear algebra
- Calculus

Required Hardware/Software

Learners need access to one of the following:

- PC: Windows 7 or higher with the latest updates installed (note: Internet Explorer is not supported).
- Mac: OS X 10.11 or higher with the latest updates installed.
- Linux: Any recent distribution that has the supported browsers installed.
- Ubuntu: 17.10+ or 14.04 LTS+.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Computer Vision

In this course, learners will develop critical machine learning skills commonly leveraged in autonomous vehicle engineering. They will learn about the life cycle of a machine learning project, from framing the problem and choosing metrics to training and improving models. This course will focus on the camera sensor, and learners will process raw digital images before feeding them into different algorithms, such as neural networks. They will build convolutional neural networks using TensorFlow and learn how to classify and detect objects in images. With this course, learners will be exposed to the entire machine learning workflow to have a good understanding of the work of a machine learning engineer and how it translates to autonomous vehicle engineering.

Course Project

Object Detection in an Urban Environment

In this project, learners will create a convolutional neural network to detect and classify objects using data from the Waymo Open Dataset. Learners are provided a dataset containing images of urban environments with annotated cyclists, pedestrians, and vehicles. First, they will perform an extensive data analysis, including the computation of label distributions, displaying sample images and checking for object occlusions. This analysis will inform learners to decide what augmentations are meaningful for the project. Next, they will train a neural network to detect and classify objects. Then, learners will monitor the training with TensorBoard and decide when to end it. Finally, they will experiment with different hyperparameters to improve performance.

Lesson 1

The Machine Learning Workflow

- Identify the key stakeholders in a ML problem.
- Frame the ML problem.
- Perform exploratory data analysis on an image dataset.
- Pick the most adequate model for a particular ML task.
- Choose the correct metric.
- Select and visualize the data.

Lesson 2

Sensor & Camera Calibration

- Manipulate image data.
- Calibrate an image using checkerboard images.
- Perform geometric transformation of an image.
- Perform pixel level transformation of an image.

Lesson 3

From Linear Regression to Feedforward Neural Networks

- Implement a logistic regression model in TensorFlow.
- Implement backpropagation.
- Implement gradient descent.
- Build a custom neural network for a classification task.

Lesson 4

Image Classification with Convolutional Neural Networks

- Write a custom classification architecture using TensorFlow.
- Choose the right augmentations to increase a dataset variability.
- Use regularization techniques to prevent overfitting.
- Calculate the output shape of a convolutional layer.
- Count the number of parameters in a convolutional network.

Lesson 5

Object Detection in Images

- Use the TensorFlow object detection API.
- Choose the best object detection model for a given problem.
- Optimize training processes to maximize resource usage.
- Implement non-maximum suppression.
- Calculate mean average precision.
- Choose hyperparameters to optimize a neural network.

Course 2

Sensor Fusion

Learn about a key enabler for self-driving cars: sensor fusion. Besides cameras, self-driving cars rely on other sensors with complementary measurement principles to improve robustness and reliability. Therefore, learners will explore lidar sensors and their role in the autonomous vehicle sensor suite. They will learn about the lidar working principle, get an overview of currently available lidar types and their differences, and look at relevant criteria for sensor selection. Also, learners will find out how to detect object such as vehicles in a 3D lidar point cloud using a deep learning approach and evaluating detection performance using a set of state-of-the-art metrics.

In the second half of the course, learners will find out how to fuse camera and lidar detections and track objects over time with an extended Kalman filter. They will get hands-on experience with multi-target tracking, where they will learn how to initialize, update and delete tracks, assign measurements to tracks with data association techniques, managing several simultaneously. After completing the course, they will have a solid foundation to work as a sensor fusion engineer on self-driving cars.

Course Project

3D Object Detection

Learners will first load and preprocess 3D lidar point clouds and then use a deep learning approach to detect and classify objects (e.g., vehicles, pedestrians). They will then evaluate and visualize the objects, including calculating key performance metrics. This project combines with the Sensor Fusion project to form an entire detection pipeline.

Lesson 1

Introduction to Sensor Fusion & Perception

- Distinguish strengths and weaknesses of each sensor.

Lesson 2

The Lidar Sensor

- Explain the role of lidar in autonomous driving.
- Extract lidar data from the Waymo dataset.
- Extract lidar technical properties such as coordinates.
- Visualize lidar data.

Lesson 3

Detecting Objects in Lidar

- Describe the state-of-the-art in 3D object detection.
- Transform a point cloud into a birds-eye view (BEV).
- Perform model inference using BEV images.
- Visualize detection results.
- Evaluate object detection performance with metrics.
- Evaluate object detection performance between models.

Course Project

Sensor Fusion

In this project, learners will solve a challenging multi-target tracking task by fusing camera and lidar detections. They will implement an extended Kalman filter to track several vehicles over time, including the different measurement models for camera and lidar. This task also requires a track management module for track initialization and deletion and a data association module to decide which measurement originated from which track. Finally, learners will evaluate and visualize the tracked objects. To complete this project, they will use a real-world dataset, exposing them to the everyday challenges of a sensor fusion engineer.

Lesson 1

Kalman Filters

- Track objects over time with a linear Kalman filter.

Lesson 2

Extended Kalman Filters

- Track objects over time with an extended Kalman filter.
- Implement motion and measurement models.
- Derive a Jacobian for nonlinear models.
- Apply appropriate coordinate transforms (e.g. sensor, vehicle coordinates).
- Fuse lidar measurements with camera detections with appropriate camera models.

Lesson 3

Multi-Tracking Tracking

- Initialize, update, and delete tracks.
- Define and implement a track score and track state.
- Calculate a simple detection probability/visibility reasoning.
- Associate measurements to tracks for multi-target tracking.
- Reduce association complexity through a gating method.
- Evaluate tracking performance through RMSE.

Course 3

Localization

Learn about robotic localization, from one dimensional motion models up to three-dimensional point cloud maps obtained from lidar sensors. Learners will begin by learning about the bicycle motion model, an approach to use simple motion to estimate location at the next time step, before gathering sensor data. Next, they'll move on using Markov localization to perform 1D object tracking, as well as further leveraging motion models. From there, they will learn how to implement two different scan matching algorithms, Iterative Closest Point (ICP) and Normal Distributions Transform (NDP), working with 2D and 3D data. Finally, utilizing these scan matching algorithms in the Point Cloud Library (PCL), learners will localize a simulated car with lidar sensing, using a 3D point cloud map obtained from the CARLA simulator.

Course Project

Scan Matching Localization

In this project, learners will recover the position of a simulated car using lidar with either ICP or NDT, two scan matching algorithms, aligning point cloud scans from the CARLA simulator. Learners will need to achieve sufficient accuracy for the entirety of a drive within the simulated environment, updating the vehicle's location appropriately as it moves and obtains new lidar data.

Lesson 1

Introduction to Localization

- Explain how a self-driving car might use GPS or detected objects to localize itself in an environment.
- Predict motion to estimate location in a future time step using the bicycle motion model.

Lesson 2

Markov Localization

- Apply the law of total probability to robotic motion.
- Derive the general Bayes/Markov filter.
- Implement 1D localization in C++.

Lesson 3

Creating Scan Matching Algorithms

- Explain ICP for localization.
- Explain NDT for localization.
- Implement ICP and NDT for 2D localization in C++.

Lesson 4

Utilizing Scan Matching in 3D

- Align 3D point cloud maps with ICP.
- Align 3D point cloud maps with NDT.
- Create point cloud maps in the CARLA simulator.

Course 4

Planning

Path planning routes a vehicle from one point to another, handling reactions when emergencies arise. The Mercedes-Benz Vehicle Intelligence team will take learners through the three stages of path planning. First, learners will apply model-driven and data-driven approaches to predict how other vehicles on the road will behave. Then they'll construct a finite state machine to decide which one of several different maneuvers your vehicle should perform. Finally, they'll generate a safe and comfortable trajectory to execute the maneuver.

Course Project

Motion Planning & Decision Making for Autonomous Vehicles

In this project, learners will implement two of the main components of a traditional hierarchical planner: the behavior planner and the motion planner. Both will work in unison to avoid static objects parked on the side of the road, preventing collision with these objects by executing either a "nudge" or a "lane change" maneuver, navigate intersections, and track the centerline on the traveling lane.

Lesson 1

Behavior Planning

- Learn how to think about high level behavior planning in a self-driving car.

Lesson 2

Trajectory Generation

- Use C++ and the Eigen linear algebra library to build candidate trajectories for the vehicle to follow.

Lesson 3

Motion Planning

- Program a decision making framework to plan a vehicle's motion in an urban environment.
- Incorporate environmental information into the motion planning algorithm.
- Generate an "optimal," feasible, and collision free path.
- Navigate the vehicle through an urban driving scenario in simulation following the rules of the road, in a human-like fashion.

Course 5

Control

This course will teach learners how to control a car once they have the desired trajectory. In other words, how to activate the throttle and the steering wheel of the car to move it following a trajectory described by coordinates. The course will cover the most basic and most common controller: the Proportional Integral Derivative or PID controller. You will understand the basic principle of feedback controls and how they apply to autonomous driving techniques.

Course Project

Control & Trajectory Tracking for Autonomous Vehicles

In this project, applying the skills acquired in previous projects, learners will design a PID controller to perform vehicle trajectory tracking. Given a trajectory as an array of locations and a simulation environment, they will design and code a PID controller and test its efficiency on the CARLA simulator. This project will help learners understand the power and the limitations of the PID controller while utilizing feedback control. This project is good training for C++ coding, which is the standard language used in the industry.

Lesson 1

PID Control

- Recognize the observation of the state of the vehicle (position, velocity), the action (steering, accelerator, brake) and the possible perturbations.
- Design and code the feedback controller (PID and MPC) for trajectory tracking, using he PID controller, and understanding how to choose the parameters to guarantee stability, and then with the MPC, a more general controller with non-linear dynamics.
- Test the controllers and evaluate their robustness to real-world perturbations.
- Analyze the differences between the two controllers.

Meet your instructors.

Thomas Hossler

Senior Deep Learning Engineer

Thomas is originally a geophysicist but his passion for computer vision led him to become a deep learning engineer at various startups. By creating online courses, he is hoping to make education more accessible. When he is not coding, Thomas can be found in the mountains skiing or climbing.

Antje Muntzinger

Self-Driving Car Engineer

Antje is a self-driving car engineer and a technical lead for sensor fusion at Mercedes-Benz. She wrote her PhD about sensor fusion for advanced driver assistance systems. By educating more self-driving car engineers, she hopes to realize the dream of fully autonomous driving together in the future.

Andreas Haja

Professor

Andreas Haja is an engineer, educator and autonomous vehicle enthusiast with a PhD in computer science. Andreas now works as a professor, where he focuses on project-based learning in engineering. During his career with Volkswagen and Bosch he developed camera technology and autonomous vehicle prototypes.

Aaron Brown

Senior AV Software Engineer

Aaron Brown has a background in electrical engineering, robotics, and deep learning. Currently working with Mercedes-Benz Research & Development as a senior autonomous vehicle software engineer, Aaron has worked as a content developer and simulation engineer at Udacity focusing on developing projects for self-driving cars.

Munir Jojo Verge

Lead Autonomous & AI Systems Developer At Mitre

Previously, Munir was a motion planning and decision-making manager at Amazon. He also worked for a 2 self-driving car companies and for Walt Disney Shanghai building TronLightcycle. Munir holds a BS in aerospace, an MS in physics, and an MS in space studies.

David Silver

Curriculum Lead

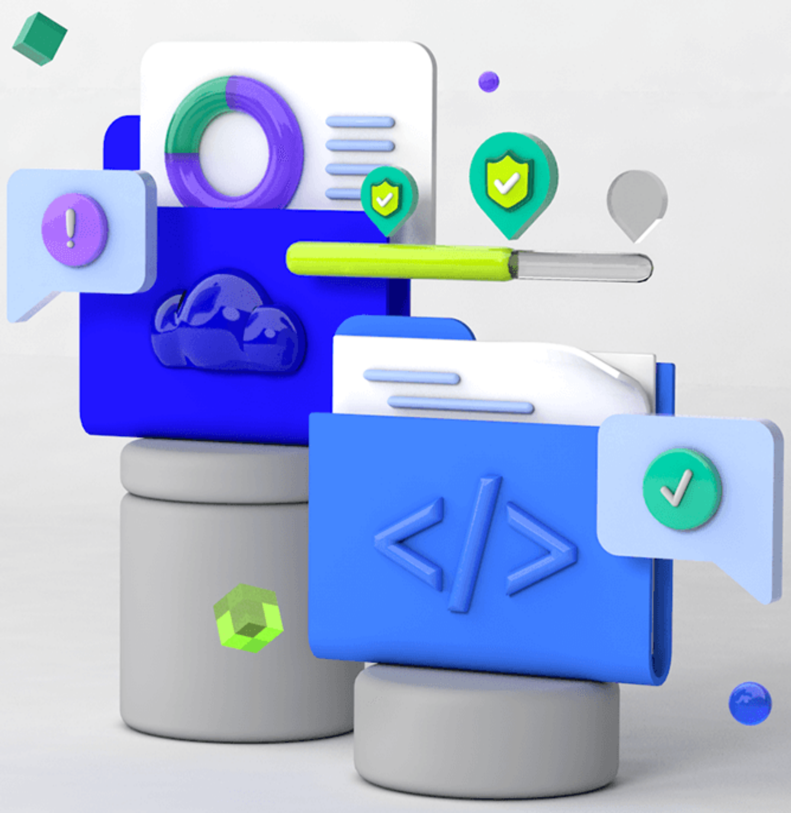
David Silver leads the School of Autonomous Systems at Udacity. Before Udacity, David was a research engineer on the autonomous vehicle team at Ford. He has an MBA from Stanford, and a BSE in computer science from Princeton.

Mathilde Badoual

Fifth Year Phd Student at UC Berkeley

Mathilde has a strong background in optimization and control, including reinforcement learning and has an engineering diploma from the electrical engineering school Supelec, in France. Previously she worked at Tesla in the energy and optimization team.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

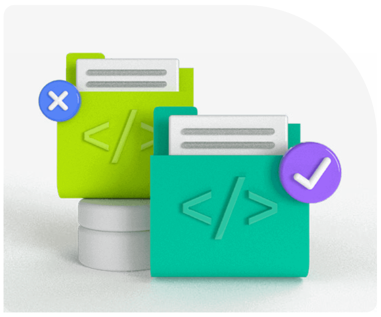
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

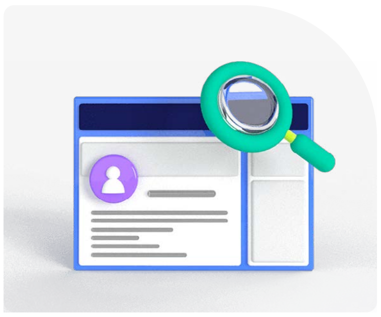
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

12.16.22 | V1.0