Supplementary Reading: The Camera Sensor

- Forsyth, D. A. and J. Ponce. (2003). *Computer vision: a modern approach* (2nd edition). New Jersey: Pearson. Read sections 1.1, 1.2, 2.3, 5.1, 5.2.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media. Read sections 2.1, 2.2, 2.3 (PDF available online: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)
- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. Read sections 1.1, 1.2, 2.1, 6.1, 6.2

Supplementary Reading: Camera Calibration

- Forsyth, D. A. and J. Ponce. (2003). *Computer vision: a modern approach* (2nd edition). New Jersey: Pearson. Read sections 5.3.
- Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media. Read sections 6.1, 6.2. 6.3 (PDF available online: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)
- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. Read sections 7.1, 7.2, 7.4, 8.4, 8.5
- Camera Calibration with OpenCV: https://docs.opencv.org/3.4.3/dc/dbb/tutorial_py_calibration.html

Supplementary Reading: Visual Depth Perception

- Forsyth, D.A. and J. Ponce (2003). *Computer Vision: a modern approach* (2nd edition). New Jersey: Pearson. Read sections 11.1, 12.1, 12.2.
- Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media. Read sections 11.1 (PDF available online: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)
- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. Read section 9.1, 10.1, 11.12
- Epipolar Geometry (OpenCV): https://docs.opencv.org/3.4.3/da/de9/tutorial_py_epipolar_geometry.html
- Depth Map from Stereo Images (OpenCV): https://docs.opencv.org/3.4.3/dd/d53/tutorial_py_depthmap.html

Supplementary Reading: Image Filtering

- Forsyth, D.A. and J. Ponce (2003). *Computer Vision: a modern approach* (2nd edition). New Jersey: Pearson. Read sections 7.1, 7.2.
- Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media. Read sections 3.2, 3.3 (PDF available online: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)
- Image filtering (OpenCV), Detailed Description section of the following document: https://docs.opencv.org/3.4.3/d4/d86/group__imgproc__filter.html

Recommended Textbooks

- [Forsyth] -- Forsyth, David A., and Jean Ponce. "A modern approach." Computer vision: a modern approach (2003): 88-101.
- [Goodfellow] -- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press. PDF available online: https://www.deeplearningbook.org/
- [Szeliski] -- Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media. PDF available online: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf
- [Hartley] -- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Course Prerequisites

This is an advanced course, intended for learners with a background in mechanical engineering, computer and electrical engineering, or robotics.

Knowledge Prerequisites

To succeed in this course, you should have the following knowledge prerequisites:

- a background in computer vision, deep learning or robotics
- programming experience in Python 3.0
- familiarity with Linear Algebra (matrices, vectors, matrix multiplication, rank, Eigenvalues and vectors and inverses)

It’s certainly helpful to know how to drive a car, but it's not a hard requirement for this course.

If you don’t have these necessary knowledge prerequisites, no sweat. There are excellent Robotics, AI, Deep Learning, Computer Vision, Controls and other courses that you can take on Coursera to prepare you for this Specialization.

If you don't have the necessary Python prerequisites, check out the Python for Everybody Specialization. If you have coding experience in another programming language, you should be able to complete this course (but may need to look up Python syntax as you go).

Software Requirements

For the final project in this course, you will develop algorithms that identify bounding boxes for objects in the scene, and define the boundaries of the drivable surface. You'll work with synthetic and real image data, and evaluate your performance on a realistic dataset.

You do not need any external software to complete this course. You will use Coursera's Jupyter Notebook system to complete the assignments, which you can code directly within the browser without any external software.