# State estimation and localization

## Basics: Least Square

* [Interactive least squares fitting simulator https://phet.colorado.edu/sims/html/least-squares-regression/latest/least-squares-regression\_en.html](https://phet.colorado.edu/sims/html/least-squares-regression/latest/least-squares-regression_en.html)
* Georgia Tech online textbook <https://textbooks.math.gatech.edu/ila/least-squares.html>
* Chapter 3, Sections 1 and 2 of [Dan Simon, Optimal State Estimation (2006)](https://onlinelibrary.wiley.com/doi/book/10.1002/0470045345)
* Recursive least square: Chapter 3, Section 3 of [Dan Simon, Optimal State Estimation (2006)](https://onlinelibrary.wiley.com/doi/book/10.1002/0470045345)
* Interactive explanation of Central Limit Theorem <http://mfviz.com/central-limit/>
* Maximum likelihood: <https://arxiv.org/pdf/0804.2996.pdf>

## Linear and Nonlinear Kalman Filters

Linear Kalman filter

* Blog post: <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>
* Chapter 5, Sections 1 and 2 of [Dan Simon, Optimal State Estimation (2006)](https://onlinelibrary.wiley.com/doi/book/10.1002/0470045345)
* Great resources: <https://www.cs.unc.edu/~welch/kalman/>
* Original article: <https://www.cs.unc.edu/~welch/kalman/kalmanPaper.html>

Nonlinear Kalman filter

* EKF: Chapter 13, Sections 1 and 2 of [Dan Simon, Optimal State Estimation (2006)](https://onlinelibrary.wiley.com/doi/book/10.1002/0470045345)
* Error State Kalman Filter <https://ieeexplore.ieee.org/document/772597> And Section 5 of <https://arxiv.org/pdf/1711.02508.pdf>
* UKF <https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf> Tutorial <https://www.cse.sc.edu/~terejanu/files/tutorialUKF.pdf> And <https://www.cs.unc.edu/~welch/kalman/media/pdf/Julier1997_SPIE_KF.pdf>

## GNSS/INS Sensing for Pose Estimation

3D Geometry and Reference Frames

* Chapter 6, Sections 1 to 3 of  [Timothy D. Barfoot, State Estimation for Robotics (2017)](http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf)
* Online: interactive quaternion calculator <https://quaternions.online/>
* Online: 3D rotation converter <https://www.andre-gaschler.com/rotationconverter/>

IMU

* Lecture on IMUs <http://stanford.edu/class/ee267/lectures/lecture9.pdf>
* Chapter 11, Section 1 of [Jay A. Farrell, Aided Navigation (2008)](https://books.google.ca/books/about/Aided_Navigation_GPS_with_High_Rate_Sens.html?id=yNujEvIMszYC&redir_esc=y)
* Article <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.616.1248>

GNSS

* Article <https://www.geospatialworld.net/article/global-navigation-satellite-system-gnss/>
* Review overviews of the [Galileo](https://m.esa.int/Our_Activities/Navigation/Galileo/Galileo_satellites) global navigation satellite system developed by the European Union, the [GLONASS](https://gssc.esa.int/navipedia/index.php/GLONASS_General_Introduction) system developed by the Russian Federation, and [COMPASS (BeiDou-2)](https://en.wikipedia.org/wiki/BeiDou_Navigation_Satellite_System) developed by the People's Republic of China.

## LiDAR Sensing

Light Detection and Ranging Sensors

* Read Chapter 6, Section 4.3 of [Timothy D. Barfoot, State Estimation for Robotics (2017)](http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf) (available for free).
* Read the Wikipedia [article](https://en.wikipedia.org/wiki/Lidar) on LIDAR sensors.
* Read Chapter 4, Section 1.9 of [Roland Siegwart, Illah R. Nourbakhsh, Davide Scaramuzza, Introduction to Autonomous Mobile Robots (2nd ed., 2011)](https://mitpress.mit.edu/books/introduction-autonomous-mobile-robots-second-edition).

LIDAR Sensor Models and Point Clouds

* Read Chapter 6, Sections 1 and 2 of [Timothy D. Barfoot, State Estimation for Robotics (2016)](http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf) (available for free).
* Explore the functionality available in the Point Cloud Library (PCL) at <http://pointclouds.org/>.

Pose Estimation from LIDAR Data

* Read Chapter 8, Section 1.3 of [Timothy D. Barfoot, State Estimation for Robotics (2016)](http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf) (available for free).
* Read the Wikipedia articles on [point set registration](https://en.wikipedia.org/wiki/Point_set_registration) and [ICP](https://en.wikipedia.org/wiki/Iterative_closest_point).
* Examine a method to produce an [accurate closed-form estimate of ICP's covariance](https://ieeexplore.ieee.org/document/4209579) from Andrea Censi of the University of Rome "La Sapienza" (now at ETH Zurich).
* Read a research paper on [LIDAR and Inertial Fusion for Pose Estimation by Non-linear Optimization](https://arxiv.org/abs/1710.07104), available for free on arXiv.
* Review the original papers by [Yang Chen and Gerard Medioni (1991)](https://ieeexplore.ieee.org/document/132043), and [Paul Besl and Neil McKay (1992)](https://ieeexplore.ieee.org/document/121791), that first described (variations of the) iterative closest point (ICP) algorithm.

## Multisensor Fusion for State Estimation

* Read Sections 5.1-5.4 and Section 6.1 of a technical report by [Joan Solà, Quaternion kinematics for the error-state Kalman filter, 2017](https://arxiv.org/pdf/1711.02508.pdf) (available for free). Note that this is an advanced reading.
* Read a [research paper](https://www.sciencedirect.com/science/article/pii/S2405896317323674) by Jay Farrell and Paul Roysdon that provides a tutorial for autonomous driving state estimation.
* Read a [Medium article](https://medium.com/@wilburdes/sensor-fusion-algorithms-for-autonomous-driving-part-1-the-kalman-filter-and-extended-kalman-a4eab8a833dd) about sensor fusion algorithms for autonomous driving (Kalman filters and extended Kalman filters).
* Review an [article](https://www.technologyreview.com/s/608321/this-image-is-why-self-driving-cars-come-loaded-with-many-types-of-sensors/) from MIT Technology Review that explains the need for sensor fusion to enable robust autonomous driving.

## Sensor Calibration

* Read an [interesting article](https://www.rscal.com/all-you-need-to-know-about-sensor-calibration/) on why sensor calibration is necessary.
* Read a [blog post](https://aimotive.com/blog/content/1227) from AImotive about the need for sensor spatial calibration and temporal synchronization.
* Explore the [cloud-based calibration](http://apollo.auto/platform/perception.html) service for self-driving cars provided by Baidu's Apollo initiative.

# Visual Perception

## Basics of 3D Computer Vision

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

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>

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>

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>

## Visual Features - Detection, Description and Matching

Feature Detectors and Descriptors

* You can find implementation resources here: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html>
* Textbook: Forsyth, D.A. and J. Ponce (2003). *Computer Vision: a modern approach* (2nd edition). New Jersey: Pearson. Read section 9.4.
* Haris Corner Detection: <https://docs.opencv.org/4.0.0/dc/d0d/tutorial_py_features_harris.html>
* Introduction to SIFT (Scale-Invariant Feature Transform): <https://docs.opencv.org/4.0.0/da/df5/tutorial_py_sift_intro.html>

Feature Matching

* Feature Matching: <https://docs.opencv.org/4.0.0/dc/dc3/tutorial_py_matcher.html>
* Feature Matching + Homography to find Objects: <https://docs.opencv.org/4.0.0/d1/de0/tutorial_py_feature_homography.html>

Outlier Rejection

* Forsyth, D.A. and J. Ponce (2003). Computer Vision: a modern approach (2nd edition). New Jersey: Pearson. Read section 19.1-19.3.

## Feedforward Neural Networks

* Feedforward neural networks: Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning (Vol. 1). Cambridge: MIT press. Read sections 6.1, 6.3. <https://www.deeplearningbook.org/>.

# Output Layers and Loss Functions: Read sections 6.2, 6.4.

# Neural Network Training with Gradient Descent: Read sections 6.5, 8.1-8.5.

# Neural Network Regularization: Read sections 7.1, 7.8, 7.12.

# Convolutional Neural Networks: Read sections 9.1-9.3.

## 2D Object Detection

The Object Detection Problem

* Implementation Resources: <https://github.com/tensorflow/models/tree/master/research/object_detection> (Fully implemented models ready to be used, from Google team)

2D Object detection with Convolutional Neural Networks

* Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338. (For understanding the problem + the metrics)

Training vs. Inference

* Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).
* Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham. <https://arxiv.org/abs/1512.02325>
* Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence. (State of the art)

Using 2D Object Detectors for Self-Driving Cars

* Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2017). Frustum pointnets for 3d object detection from rgb-d data. arXiv preprint arXiv:1711.08488. (3D object detection from 2D)
* Forsyth, D.A. and J. Ponce (2003). *Computer Vision: a modern approach* (2nd edition). New Jersey: Pearson. Read section 18.2 (Tracking)

## Semantic Segmentation

The Semantic Segmentation Problem

* Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3223). (For understanding the problem + the metrics)
* Neuhold, G., Ollmann, T., Bulò, S. R., & Kontschieder, P. (2017, October). The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In ICCV (pp. 5000-5009).

ConvNets for Semantic Segmentation

* Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561.
* Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017, July). Pyramid scene parsing network. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (pp. 2881-2890). (State of the art)

Semantic Segmentation for Road Scene Understanding

* Forsyth, D.A. and J. Ponce (2003). *Computer Vision: a modern approach* (2nd edition). New Jersey: Pearson. Read section 8.1, 8.2, 8.3 (Edge Detection); 16.1, 16.2 (Hough transform)

# Motion Planning

## The Planning Problem

* P. Polack, F. Altche, B. Dandrea-Novel, and A. D. L. Fortelle, “[The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles](https://ieeexplore.ieee.org/abstract/document/7995816)” 2017 IEEE Intelligent Vehicles Symposium (IV), 2017. Gives an overview of the kinematic bicycle model.
* S. Karaman and E. Frazzoli, “[Sampling-based optimal motion planning for non-holonomic dynamical systems](http://amav.gatech.edu/sites/default/files/papers/icra2013.Karaman.Frazzoli.submitted.pdf),” 2013 IEEE International Conference on Robotics and Automation, 2013. Introduces the RRT\* algorithm as an example of sampling-based planning.
* N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, “[CHOMP: Gradient optimization techniques for efficient motion planning](https://kilthub.cmu.edu/articles/CHOMP_Gradient_Optimization_Techniques_for_Efficient_Motion_Planning/6552254/1),” 2009 IEEE International Conference on Robotics and Automation, 2009. Introduces the CHOMP algorithm as an example of applying calculus of variations to planning.
* M. Pivtoraiko, R. A. Knepper, and A. Kelly, “[Differentially constrained mobile robot motion planning in state lattices](https://ri.cmu.edu/pub_files/2009/3/ross.pdf),” Journal of Field Robotics, vol. 26, no. 3, pp. 308-333, 2009. Introduces the state lattice planning method.

## Mapping for Planning

* S. Thrun, W. Burgard, and D. Fox, [Probabilistic robotics](http://www.probabilistic-robotics.org/). Cambridge, MA: MIT Press, 2010. Read Chapter 9 - Occupancy Grid Mapping for an overview of how occupancy grids are generated.
* P. Bender, J. Ziegler, and C. Stiller, “[Lanelets: Efficient map representation for autonomous driving](http://static.aixpaper.com/pdf/d/f5/gs.2014.81cd3b9828.v1.pdf" \o "Publication Link" \t "_blank),” 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014. Introduces the concepts of lanelets used in mapping.

## Mission Planning in Driving Environments

* Steven M Lavalle, [Planning Algorithms](http://planning.cs.uiuc.edu/), 2006, Cambridge University Press. Chapter 2 covers discrete planning over graphs including Dijkstra's and A\*.
* N. J. Nilsson, “[Artificial intelligence: A modern approach](http://aima.cs.berkeley.edu/),” Artificial Intelligence, vol. 82, no. 1-2, pp. 369–380, 1996. Read Chapters 3.4-3.5 for an overview of search algorithms in graphs.

## Dynamic Object Interactions

* C. Urmson, C. Baker, J. Dolan, P. Rybski, B. Salesky, W. Whittaker, D. Ferguson, and M. Darms, “[Autonomous Driving in Traffic: Boss and the Urban Challenge](https://www.aaai.org/ojs/index.php/aimagazine/article/view/2238),” AI Magazine, vol. 30, no. 2, p. 17, 2009. This gives an overview of some of the methods used to handle dynamic obstacles in the DARPA Urban Challenge.

## Principles of Behaviour Planning

* J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, “[A behavioral planning framework for autonomous driving](https://ieeexplore.ieee.org/abstract/document/6856582),” 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014. This gives a nice overview of an example framework that can be used in behaviour planning.
* R. S. Sutton and A. G. Barto, [Reinforcement learning an introduction](http://incompleteideas.net/book/the-book-2nd.html). Cambridge: A Bradford Book, 1998. Gives a great introduction to reinforcement learning concepts.

## Reactive Planning in Static Environments

* Fox, D.; Burgard, W.; Thrun, S. (1997). "The dynamic window approach to collision avoidance". Robotics & Automation Magazine, IEEE. 4 (1): 23–33. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1109/100.580977](https://doi.org/10.1109%2F100.580977). This gives an overview of dynamic windowing and trajectory rollout.
* M. Pivtoraiko, R. A. Knepper, and A. Kelly, “[Differentially constrained mobile robot motion planning in state lattices](https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20285),” Journal of Field Robotics, vol. 26, no. 3, pp. 308–333, 2009. This paper is a great resource for generating state lattices under kinematic constraints.

## Putting it all together - Smooth Local Planning

* A. Kelly and B. Nagy, “[Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control,](https://journals.sagepub.com/doi/abs/10.1177/02783649030227008?casa_token=1eJaU-j-rQMAAAAA%3AkOxyZCACePcPX12nrkI9ytr-xQC0KY9nZ_TZ4m7ClMuSbHmpA8TOnlmNMDQVxa7-K_9bEtOFm820&)” The International Journal of Robotics Research, vol. 22, no. 7, pp. 583–601, 2003. This paper discusses the math behind generating spirals to desired terminal states.
* A. Piazzi and C. G. L. Bianco, “[Quintic G/sup 2/-splines for trajectory planning of autonomous vehicles](https://ieeexplore.ieee.org/abstract/document/898341),” Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511). This paper discusses the math behind generating quintic splines to desired terminal states.
* M. Mcnaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, “[Motion planning for autonomous driving with a conformal spatiotemporal lattice](https://ieeexplore.ieee.org/abstract/document/5980223),” 2011 IEEE International Conference on Robotics and Automation, 2011. This paper introduces the concepts behind generating a conformal spatiotemporal lattice for on-road motion planning.