

# Roumen Guha (perception engineer)

+1 (469) 328 3442  
roumen.guha@gmail.com  
roumenguha.github.io

## Experience

- Oct 2020 - **Graduate Teaching Assistant**, *Department of Computer Science & Engineering, UCSD.*  
Dec 2020
  - Assisted **Professor Ryan Kastner** in teaching two introductory High-Level Synthesis (HLS) courses.
  - Created assignments and labs, graded student submissions, held office hours while taking three technical courses myself.
- Sept 2019 - **Graduate Research Assistant**, *CHEI Drone Lab, Robotics Institute, UCSD.*  
September 2020
  - Created an HLS FPGA debayering + smoothing image processing pipeline for upstream feature extraction.
  - Assisted in the hardware development of DevCAM, an open-source multi-camera FPGA-powered machine vision system developed at UCSD.
- Dec 2016 - **Controls & Electrical Team Lead**, *Wisconsin Hybrid SAE Vehicle Team, Madison, WI.*  
May 2018
  - Managed team to completion of converting a stock SUV into an **electric vehicle** with 35-mile range, in **under 4 months**.
  - Developed motor control code for **driving and regenerative braking**, improving range on a single charge.
  - Managed student team to **integrate a small ethanol engine** into our electric vehicle to achieve a range-extended hybrid.
  - Planned, designed, built, wired and wrote code for a dynamometer over summer. Used by several vehicle teams for testing.
  - Performed high-voltage wiring, built and debugged vehicle wiring-harness, and integrated power electronics.
  - Worked with **Simulink**, **MotoHawk**, **MotoTune**, **CANoe**, **CANdb++** and other Woodward and Vector development tools and software.

## Education

- 2019–2021 **M.S., Robotics Engineering**, *University of California, San Diego, Advanced courses: Robot Sensing & Estimation, Reinforcement Learning, Image Understanding, Computer Vision, FPGA High-Level Synthesis, GPU Programming.*  
**GPA: 3.3**
- 2014–2018 **B.S., Electrical Engineering & Mathematics**, *University of Wisconsin-Madison, Dean's Honors, AMCHAM Scholarship, Advanced courses: Robotics, Machine Learning, Image Processing, Optimization, Artificial Intelligence.*  
**GPA: 3.4**

## Favorite Projects

Most of these (and more) can be found on my GitHub page.

- December 2020 **Ensemble Learning: Training Agents to Walk.**
  - Implemented an ensemble learning method called Multi-TD3 to teach agents in the OpenAI gym to walk.
  - Evaluated this method against SUNRISE and TD3 against a suite of OpenAI agents.
- July 2020 **Real-time American Sign-Language Recognition.**
  - Implemented an ASL detector in PyTorch via transfer learning on ResNet50 and VGG16.
  - Used the generated model to implement a live detector in OpenCV.
- March 2020 **Visual-Inertial SLAM via the Extended Kalman Filter.**
  - Wrote an EKF routine to filter noisy **IMU** data using **visual-keypoint** data extracted via **Harris Corners**.
  - Achieved beautiful results. Animations compared to ground truth available on GitHub.
- February 2020 **Particle Filter SLAM.**
  - Wrote a particle filter routine to filter noisy **IMU** data and build an occupancy grid of the environment.
  - Achieved passable results. Animations available on GitHub.
- January 2020 **Stop Sign Detection Redux.**
  - Made a red color classifier using **logistic regression** color classifier, and added various **shape heuristics** to make a stop-sign detector in **Python** and **OpenCV**.
- Dec 2017 **Dancing Robot.**
  - Built a dancing robot arm with a robotic arm, utilizing **inverse kinematics** and a **DSP-based beat-tracker**.
  - Programmed using **ROS** in **Python** on a **Raspberry Pi**. Video demo available on GitHub.
- Oct 2017 **Stop Sign Detection.**
  - Coded a **Mathematica** image processing routine that **detected stop signs** in a class-provided dataset with 98% accuracy.
  - Utilized classical techniques such as **segmentation, filtering, dilation and erosion, opening and closing**.
  - Won Silver in class competition.**

## Skills

Advanced Python, OpenCV, Vivado HLS  
Intermediate MATLAB, PyTorch, ROS, C, Git