AJ Chandrasekaran

U.S. Citizen • (248) 719-2472 • ajaay@umich.edu • Novi, MI 48375 • ajaayc.github.io

Experience (7 Years)

Stellantis Dec 2019 - Mar 2024

Senior Automated Driving Software Engineer – AI Motion Planning Team, Core Development Team Auburn Hills, MI

- Collaborated in global organization of ~700+ software, systems, calibration, and product engineers to build L2 autonomous driving platform in C++ and Simulink with Bazel build system, Docker development environment, and Git version control
- Dissected and documented 10,000+ lines of esoteric BMW C++ codebase obtained from BMW-Stellantis partnership to help team adapt primary BMW codebase for Stellantis vehicles while developing a secondary custom-Stellantis codebase
- Conducted dozens of in-vehicle tests to diagnose and resolve problems across AUTOSAR/ROS publisher-subscriber stack
- Validated system requirements through simulation of AUTOSAR software components via Google Test (GTest) framework
- Built emergency failover motion planning module to deaccelerate vehicle to a stop in case of primary planning module failure
- Facilitated migration of data visualization plugins from RViz to Foxglove Studio (TypeScript) to enhance software diagnostics
- Prototyped and validated trajectory planning algorithms in Python and visualized algorithm execution via Jupyter Notebook
- Trained ~10 new team members and pioneered new training curriculum used by dozens of new software developers

Epic Jul 2018 - Aug 2019

Software Developer – Epic ASAP Team Madison, WI

- Migrated several features of electronic medical record software to JavaScript and C# given existing Visual Basic codebase
- Fixed security vulnerabilities and various bugs in SVN version-controlled codebase touched by hundreds of developers

University of Michigan Transportation Research Institute May 2017 - Jun 2018

Graduate Student Research Assistant Ann Arbor, MI

- Constructed vehicle perception system to monitor the safety and quality of construction materials along the US-23 highway over a 2 year period, merging data from a Novatel GPS, Quanergy M-8 Lidar, and a Mobileye unit
- Developed multi-threaded C++ code utilizing VTK and PCL libraries to extract, interpret, and visualize Lidar readings
- Interfaced with a Mobileye unit via vehicle CAN message network to measure lane marking quality over time

Education

University of Michigan Sept 2013 - Apr 2018

MSE in Electrical and Computer Engineering, Robotics — GPA: 3.730/4.000 Ann Arbor, MI

BSE in Computer Science — GPA: 3.759/4.000

Relevant Coursework: Autonomous Robotics Laboratory, Self-Driving Cars: Perception & Control, Machine Learning, Motion Planning, Mobile Robotics Algorithms, Robot Kinematics & Dynamics, Software Engineering, Control Systems Design, Data Structures & Algorithms, Operating Systems, Computer Architecture

Academic Projects

Collision Estimation for Safe Planning • Motion Planning Course Project Mar 2018 - Apr 2018

- Devised and implemented a Gaussian Mixture Model-based algorithm in C++ to expand upon existing Monte Carlo simulation methods to estimate the probability that a robot following a motion plan would collide with an obstacle
- Simulated a linear feedback controller and a Kalman Filter estimate for robot position via OpenRAVE and Armadillo

Deep Neural Network Vehicle Detection System • Self-Driving Cars Course Project Nov 2017 - Dec 2017

- Collaborated with a team to develop a classifier to detect cars in various images captured from GTA simulation
- Utilized TensorFlow Python Object Detector API and NVIDIA CUDA library on AWS EC2 GPU instances to train convolutional neural network (faster RCNN) on ~6,000 training images with known bounding box locations of cars
- Ranked 5th place among 35 teams in class competition by achieving 63% mean absolute error on instructor test set

TrashBot • Senior Undergraduate Capstone – Autonomous Robotics Laboratory Course Project Mar 2017 - Apr 2017

- Collaborated with a team to build a robot that would navigate and detect and remove trash in urban environments
- Implemented forward and inverse kinematics as well as trajectory planning algorithms in Python for movement of a robotic arm to efficiently pick up and dispose of objects detected via 3D point cloud data from a Microsoft Kinect

Tools

Languages: C++/C, Python, Matlab/Simulink, Java, BASH, SQL, JavaScript, C#, CSS

Software & Libraries: Bazel, CMake/Make, Git, Emacs, VSCode, Docker, Linux terminal, GTest, ROS, Scikit-Learn

Subject Knowledge: Machine learning, Planning algorithms, Kalman filtering, Multithreading, Sockets, AUTOSAR, CAN