ADHEESH CHATTERJEE

adheeshchat@gmail.com | +1 (240) 784-7779 | 3405 Tulane Drive, Hyattsville, MD 20783 | adheeshc.github.io

EXPERIENCE

Computer Vision Engineer, Vidalign Inc. *Aug 2020 - Present*
- Developed a precise facial feature detection and tracking module from scratch with dlib and OpenCV to be used for real-time generation of a 3D mesh of the face.
- Led a team of engineers to implement, optimize, and deploy deep learning and vision pipelines for different upgrades and versions of the VFX software.
- Designed and implemented a parametric model of the facial wrinkles to improve detection and make the model robust

Teaching Assistant, University of Maryland *Jan 2020 - May 2020*
- Provided course support and assisted in the development of new course material for the Robot Learning course covering topics focused on Reinforcement Learning, Control through Machine Learning, and Evolutionary Robotics
- Performed all assistant teaching duties including mentoring, lecturing, researching, and clerical help

Research Assistant, University of Maryland *Sep 2019 - May 2020*
- Developed a Multi-Agent Cooperative Reinforcement Learning solution to the frontier exploration problem
- Decentralized a system of drones and mobile robots while working with a modified PPO/Rainbow algorithm

Summer Research Assistant, University of Maryland *May 2019 - Sep 2019*
- Created an integrated Semantic Segmentation and Depth Estimation (RGB-D) network working primarily on the Cityscapes and Kitti datasets
- Designed an encoder-decoder CNN architecture (VGG and Resnet backend) by performing sensor fusion of image and LIDAR data

PROJECTS

SLAM (Simulataneous Localization and Mapping) Projects
- Localization - Extended Kalman Filter, Unscented Kalman Filter, and Particle Filter (Monte Carlo)
- Mapping - 2D Gaussian grid, Ray Casting, K-means Clustering and Rectangle Fitting using LIDARs
- Complete Frameworks - FastSLAM, GraphSLAM, LSD-SLAM, RTab-SLAM

ROS Projects
- Developed a ROS interface using raw sensor data (IMU, Camera, Magnetic Encoders) for tracking the robot using a Raspberry Pi and an Arduino Nano. Performed EKF-SLAM to map out the UMD Robotics Realization Lab on Rviz
- Simulated an assembly line of Pick and Place robots to sift through objects and separate out individual components using ROS Packages

Computer Vision Projects for Autonomous Driving
- Visual Odometry, Lane Detection, Traffic Sign Recognition and Classification using HOG feature descriptors and SVM, Lucas Kanade Object Tracker, RCNN object detector using Selective Search and Region Proposal.

Structure From Motion
- Used RANSAC based Outlier Rejection, PnP Estimation and Bundle Adjustment to reconstruct a 3D point cloud of surrounding structures and environment in C++ using OpenGL and 6DOF camera pose calibration

Sensor Fusion
- Processed Lidar point cloud, Radar, and Camera data to calculate total time to collision from preceding vehicles and 3D object tracking in C++ (using Point Cloud Library)

Motion Planning Projects
- BFS, DFS, Dijkstra, A*, RRT, RRT*, PRM, B-Spline, CubicSpline, Dubins Path to find a collision-free path
- Kruskal, Prim, Boruvka and Nearest Neighbour algorithm to form a Minimum Spanning Tree to solve the TSP

EDUCATION

Nanodegree, Robotics Software Engineer *Mar 2020 - Apr 2020*
Udacity **Grade: A**

Specialization, Deep Learning *Dec 2019 - Jan 2020*
Coursera **Grade: A**

Masters of Engineering, Robotics *Aug 2018 - May 2020*
University of Maryland **GPA: 3.63**

Bachelor of Technology, Mechanical Engineering w/ Minors in Computer Science *Aug 2014 - May 2018*
Vellore Institute of Technology **GPA: 3.6**

SKILLS

Interests: SLAM, Computer Vision, Sensor Fusion, 3D Mapping, Camera Calibration, Object Classification, 3D Reconstruction, Object Tracking, Multi View 3D geometry, Reinforcement Learning, Controller Design
Engineering: SolidWorks, Gazebo, VREP, Raspberry Pi, Arduino, ANSYS Workbench, ANSYS Mechanical
Programming: Python, C/C++, ROS, Matlab, Rust, Git, OpenCV, OpenGL, Pytorch, CUDA, Tensorflow, Scikit-Learn, Numpy, Matplotlib, Pandas, Eigen, HTML5+CSS, ROS2