

Abhijit Mahalle

abhimah@umd.edu | GitHub: [abhijitmahalle](#) | LinkedIn: [Abhijit Mahalle](#) | Website: [abhijitmahalle](#) | College Park, MD

Domain skills: Robot Perception, Computer-Vision, ROS, Deep Learning, Robotics Software Development, Path-planning, Sensor Fusion

EDUCATION

University of Maryland, College Park

Master of Engineering, Robotics

May 2023

GPA: 3.7/4.0

Courses: Perception, Path-Planning, Software Development, Machine Learning, Deep Learning, Robot Learning, Aerial Robotics, Control Systems

University of Mumbai, India

Bachelor of Engineering, Mechanical Engg.

May 2018

GPA: 8.36/10.0

TECHNICAL SKILLS

Languages and Tools: C++, Python, MATLAB, ROS, Gazebo, RViz, Git, Docker, pytest, gtest

Libraries: OpenCV, PyTorch, TensorFlow, Keras, numpy, sklearn, pandas

Planning: A*, RRT*, Real Time-RRT*, Dijkstra

Deep Learning: VGG16, ResNet, SfmLearner, LSTM, Transformers

Controls: MPC, LQG, LQR, PID

WORK EXPERIENCE

Perception and Robotics Group (PRG) | Research Assistant

May 2022 - Present

University of Maryland

- Created a ground-truth dataset for deep-learning models by fusing data-streams from **Dynamic Vision Sensor** (event camera), **Vicon** motion capture system, and monocular camera for motion, depth, scene segmentation, optical flow, and ego-motion and simulated the recorded scenes on **Mujoco**.
- Developed a pipeline that calibrates event and monocular camera simultaneously and in real-time by converting event-stream to gray-scale images using E2VID deep-learning network.
- Developed a neural network to extract depth from two image sequences and achieved an accuracy of 90%.

Worley | Piping Design Engineer

Sept 2018 - July 2021

Mumbai, India

- Designed piping systems for effective fluid transfer within the process plants considering chemical process requirements and effect of temperature, pressure, and weight on pipe material.

PROJECTS

3D Scene Reconstruction - Reconstructed a **3D** scene and simultaneously obtained camera poses from a given set of images using their feature point correspondence (**epipolar geometry**, **triangulation** and **non-linear optimization**). [GitHub](#)

Visual Odometry - Computed trajectory of a camera placed on an autonomous vehicle using the concepts of feature-matching, **RANSAC**, epipolar geometry and achieved an accuracy of over **90%**. [GitHub](#)

Panorama stitching - Stitched images to create a **panorama** using traditional (Homography estimation using feature points) and deep learning (HomographyNet: Supervised and unsupervised) methods. [GitHub](#)

Face swap - Swapped faces in videos using traditional (Delaunay Triangulation and Thin Plate Spline) and deep learning (Position Map Regression Network) methods. [GitHub](#)

Edge Detection - Detected edges using a simplified version of the probability of boundary detection algorithm. [GitHub](#)

AR Tag detection and tracking - Decoded custom Aruco marker using corner detection and fast Fourier Transform and superimposed a custom image and a virtual cube on it using projection, calibration, and homography matrices. [GitHub](#)

Lane detection and turn prediction - Detected straight and curved lanes using homography, hough lines, polynomial curve fitting, and predicted turn by calculating the radius of curvature for detected lanes. [GitHub](#)

Optical flow - Tracked a moving vehicle and determined its speed using the concepts of difference of images, contour detection, SIFT feature matching, pixel displacement, and achieved an accuracy of **99%** for different speeds. [GitHub](#)

Human Position Estimator and Detector - Developed a software by **Agile Iterative Process** that detects and tracks humans using a pre-trained **HOG** descriptor and **SVM** detector of OpenCV. [GitHub](#)

Face recognition - Implemented **Bayes' classifier**, **k-NN**, **Kernel** and **Boosted SVM** along with **PCA** and **LDA** for face recognition and achieved an accuracy of over **70%** for each of them. [GitHub](#)

Hand written digit recognition - Implemented **Linear** and **Kernel SVM**, **Logistic Regression**, and **CNN** along with **transfer learning** on MNIST dataset with an accuracy of over **90%**. [GitHub](#)

Image denoising - Implemented a convolutional auto-encoder with skip connections for general image denoising and improved the performance by **10%** over baseline.