Abhijit Mahalle

amahalle60@gmail.com | linkedin.com/in/abhijitmahalle | Portfolio: abhijitmahalle.github.io | GitHub: github.com/abhijitmahalle

EDUCATION

Master of Engineering, Robotics | University of Maryland, College Park **GPA: 3.64** | May 2023

Coursework: Robot Perception, Path Planning, Software Development, Machine Learning, Deep Learning, Aerial Robotics, Robot Modeling, Control Systems

EXPERIENCE

Software Engineer | Stellantis **Dec 2023 - Present**

Tools & Skills: C++, Python, Software Development, Agile, CI/CD

- Developing software for charging infrastructure of autonomous electric vehicles.

Computer Vision Research Engineer | Perception and Robotics Group, UMD **Aug 2021 - Nov 2023**

Tools & Skills: C++, Python, ROS, OpenCV, NumPy, PyTorch, Docker, Computer Vision, Deep Learning

Camera Calibration

- Developed a software to calibrate event and classical cameras simultaneously by reconstructing grayscale images from the **event camera** output using deep neural network.

Dataset Generation

- Created a real-world indoor ground-truth dataset by fusing data-streams from 3 event cameras, classical camera, and 2 IMUs to train **VIO** and **SLAM** deep learning models.
- Utilized **Vicon** motion capture system to get ground-truth camera and object poses and **simulated** the recordings using **Mujuco** simulator by developing a software to get ground-truth depth and segmentation masks.

Neural Network Development and Training

- Achieved an IoU score of 0.75 for **segmentation** and Average Endpoint error of 0.15 m/s for **motion estimation** task by developing two encoder-decoder networks to estimate scene **depth**, **camera** and **object poses** from the event data-stream.
- Combined the depth and pose estimates to generate **optical flow** to jointly optimize the two networks using **unsupervised learning**.

Software Engineer | Jacobs Engineering **Aug 2018 - Jul 2021**

Tools & Skills: C++, Python, PyTorch, Computer Vision, Deep Learning, Software Development, Agile, CI/CD

3D Scene Reconstruction using Structure from Motion (SfM) | Multiple view geometry

- Reconstructed a **3D scene** and simultaneously obtained **camera poses** from a set of images using their feature point correspondence, **epipolar geometry**, triangulation, **PnP, bundle adjustment**, linear, and non-linear optimization.

Multiple object detection and tracking

- Resulted in a MOTA of 0.691 on multi-human tracking task by implementing a one-shot CNN tracker with two independent branches for **detection** and **re-identification** tasks. Jointly optimized the network for the two tasks.

Facial Recognition

- Increased the accuracy by 40% on facial recongition task by leveraging **transfer learning** to fine-tune the pre-trained VGG-16 model. Trained the model on small dataset having 10 classes and 140 images per class.

Image Segmentation

- Achieved 99% accuracy and 0.97 dice score for semantic segmentation task by training **U-Net** network on Carvana dataset.

Image Denoising

- Improved the performance by 10% over baseline for image denoising task by formulating a convolutional autoencoder with **skip connections** and training it on SID dataset.

PROJECTS

Depth Estimation using Stereo Camera | Python, OpenCV, NumPy Link

- Estimated depth and disparity from image sequences using feature-correspondence, epipolar geometry, rectification, camera calibration, and **template matching** using sum of squared differences.

Target Detection, Image Superimposition, and Augmented Reality | Python, OpenCV, NumPy Link

- Detected and tracked an April tag in a video sequence by background removal using FFT and Harris corner detection.
- Decoded the tag by identifying its position and orientation using homography.
- Superimposed a custom image and placed a **3D virtual** cube on tag using **homography, calibration**, **projection** matrices.

Voxl m500 Drone | Autonomous System, Aerial Robotics Link

- Developed a **C++** software for **real-time detection** and **tracking** of humans using monocular camera and YOLO-v5 classifier.
- Used 1D **LiDAR** sensor for depth sensing. Programmed **PD** controllers for precise yaw and depth control of the drone.
- Constructed a **ROS** node for off-board position control of drone to autonomously follow a moving April tag, move through set way-points, and to move in a figure-of-eight.

Home Organizing Robot | Software Development, SLAM Link

- Developed a **ROS** package in **C++** by **Agile** Iterative Process with GitHub Continuous Integration and test-driven development using **GTest** for Tiago mobile manipulator for indoor search and object manipulation.
- Used **MoveBase** for autonomous navigation, **MoveIt** for manipulator control, **OpenCV** for filtering and object detection.
- Maintained software version control using **Git**, checked build using **Travis CI**, and attained a code coverage of 80%.

Design, control, and simulation of Mars Rover | Robot Manipulation

- Designed a **3D CAD** model of Curiosity Rover in **SolidWorks** with a 5-DOF KUKA KR5 robotic manipulator attached to it.
- Formulated **DH parameters** to compute **Forward** and **Inverse Kinematics** of the robotic arm, used **MoveIt** for **motion planning**, and simulated the model on a Mars-like uneven terrain in **Gazebo**.
- Utilized the **ROS** control package to tune the position and velocity control for all 5 joints.

Path Planning for Autonomous Vehicles | C++, Python Link

- Implemented **BFS, Dijkstra, A***, and **Real Time-RRT*** path planning algorithms on differential-drive robot **ROS** TurtleBot.

TECHNICAL SKILLS

Programming Languages: C++, Python, MATLAB, C, Bash, HTML, UML
Softwares & Tools: Linux, Git, Docker, CMake, GCP, ROS
Libraries: OpenCV, PyTorch, TensorFlow, NumPy, SciPy, sklearn, pandas, Matplotlib, GTest, pytest
Domain Skills: Computer Vision, Motion Planning, Machine Learning, Deep Learning, Data Structures, Algorithms
Software Practices: Agile, Scrum, SDLC
Deep Learning Architectures: MLP, CNN, Autoencoder, VAE, RNN, LSTM, GAN, Transformer