Uday Sankar

☎ +1 (774) 519-8248 ✉ udaysankar.ambadi@gmail.com Website LinkedIn GitHub

EDUCATION

Worcester Polytechnic Institute (WPI) **Aug 2022 - May 2024**
Master of Science (MS) in Robotics Worcester, MA
Relevant Coursework: Computer Vision, Deep Learning, Machine Learning, Motion Planning, Robot Control

Cochin University of Science and Technology, India **Jul 2018 - May 2022**
Bachelor of Technology (B. Tech) in Electrical and Electronics Engineering Kochi, India
Relevant Coursework: Industrial Automation

SKILLS

Programming Languages: Python, C++, MATLAB, Bash.
Packages: PyTorch, TensorFlow, Keras, OpenCV, Pandas, Point Cloud Library (PCL), OpenGL, g2o, GTSAM.
Technologies: CMake, ROS/ROS2, Isaac Sim, Gazebo, ONNXruntime, TensorRT, Docker, Image Processing, Camera Calibration, Git, Blender.
Deep Learning Architectures: CNN, RNN, Autoencoder, VGG16, ResNet, DenseNet, NeRF, Transformers, LSTM, GAN.

EXPERIENCE

Onki Robotics **August 2024 – Present**
Robotic Perception Intern *New York City, NY*

- Currently working on compiling a series of computer vision models, including Depth Anything, YOLOv10, DeepSort, etc. for the NVIDIA Jetson Orin platform.
- Utilizing TensorRT and other techniques to enhance model performance and increase inference speed.
- Improved the frame rate from less than 1FPS to about 6FPS on the Orin platform, with ongoing efforts to further enhance model efficiency.

UVeye **February 2024 – April 2024**
Computer Vision Algorithm Intern *Boston, MA*

- Worked on an end-to-end computer vision project aimed at measuring the toe angle of vehicle wheels to detect misalignment for an autonomous vehicle inspection system.
- Utilized multiple cameras, curated datasets using CVAT and ClearML, and fine-tuned Segment Anything Model (SAM) to detect the wheel rims of the vehicles.
- Applied post-processing techniques on the SAM outputs to accurately calculate the total toe angle of the wheels.

Worcester Polytechnic Institute **August 2023 – December 2023**
Graduate Student Researcher under Dr. Andre Rosendo *Worcester, MA*

- Set up and simulated a Franka Emika Robot in NVIDIA Isaac Sim environment with a overhead camera.
- Added capabilities to the robot to perform simple manipulation tasks like pick and place, rotating, and stacking objects.
- Implemented a state machine based system using both ROS2 and Isaac Sim's API to control the robot.
- Research work on how Large Language Models can be used to break down complex robotic tasks.

Orangewood Labs **June 2023 – August 2023**
Robotics Intern *San Francisco, CA*

- Involved in the development of **RoboGPT**, an LLM based interface for a **6DOF Robotic Arm** that converts simple natural language instructions into robotic action.
- Developed and tested perception stacks for robot using **CLIP** and **Multimodal InstructBLIP** model.
- Set up a multi-camera system using Realsense D435 and D405 RGB-D cameras installed at the workspace and the end-effector.

PROJECTS

xfeatSLAM

- Integrated **XFeat architecture** with ORB SLAM3 pipeline to create a powerful Visual SLAM (Simultaneous Localization and Mapping) system, that leverages deep learning-based image feature descriptors.
- Independently developed the **C++ Implementation** of the XFeat (accelerated features) model for feature detection and description, drawing from methodologies outlined in the XFeat paper.
- Utilized libtorch (PyTorch C++ API) and OpenCV to optimize the model's post processing, focusing on efficient feature computation and real-time processing with GPU acceleration.

Structure from Motion (SfM) and Neural Radiance Fields(NeRFs)

- Simultaneously performed 3D reconstruction and extracted camera pose from given camera correspondences using concepts from multi-view geometry.
- Implemented a pipeline with algorithms like (Non)Linear triangulation, (Non)Linear PnP and Bundle Adjustment(BA) from scratch using Python.
- Leveraged Neural Radiance Fields to generate novel views of intricate scenes by optimizing a continuous volumetric scene function with a limited set of input views.

Simple SLAM

- Implemented a sparse monocular visual SLAM system from scratch using python, with ORB feature detection, brute force matching, and 3D point triangulation for real time mapping.
- Enhanced the quality of features extracted using a number of techniques like RANSAC, ratio tests, etc.
- Integrated OpenGL and Pangolin for 3D visualization of the generated point cloud, and improved the map accuracy by performing pose graph optimization with g2o-python.

Perception Stack for an Autonomous Vehicle

- Developed a perception stack for an autonomous driving, based on monocular video data obtained from a Tesla Model S.
- The 3D world perceived through computer vision models (YOLO, MiDAS, etc.) for object detection, human pose estimation, text recognition, and 6D vehicle pose estimation.
- The data obtained is then reconstructed for visualization using 3D graphics in Blender.

Image Boundary Detection

- Detected edges in images using simplified probability based boundary detection algorithm.
- Created different types of filter banks including Oriented Derivative of Gaussian (ODoG), Leung-Malik (LM) and Gabor Filters for creating a collection of filter responses for the input image.
- The filter responses are then clustered using K-Means Clustering, followed by post processing to produce boundary detection of the image.

AutoPano Panorama Stitching

- Created a pipeline using python that takes in RGB images as input and stitches them to create a panorama image.
- Extracted features, matched them, and calculated the homography matrix by building traditional methods like ANMS, RANSAC outlier rejection and Homography estimation from scratch.
- Optimized the algorithm to be robust to stitching a sequence of images.

Image Classification using ResNet, ResNeXt and DenseNet

- Classification of CIFAR-10 dataset using PyTorch implementation of ResNet, ResNeXt and DenseNet architectures.
- Designed simplified version of these models directly from the paper and implemented, trained and tested them using PyTorch.
- Also conducted a comparison study between the models based on their accuracy of preditions and inference time.