SAKSHI KAKDE

sakshi@terpmail.umd.edu | +1 240-714-0043 | GitHub: *sakshikakde* | *LinkedIn: Sakshi Kakde* | Website: *sakshikakde* | United States

Domain skills: Robot Perception, Computer Vision, Event Vision, Sensor Fusion, Mapping, Localization, Deep Learning, Artificial Intelligence

---

EDUCATION

University of Maryland, College Park   *Jan 2021 - Dec 2022*
M.Eng. Robotics | CGPA: 3.87/4 | *Roles: Teaching Assistant for CMSC426 (Computer Vision) and ENPM673 (Perception for Robotics).*
Courses: Foundations of Deep Learning, Numerical Optimization, Perception, Planning and Controls for Robotics, Software Development.
Visvesvaraya National Institute of Technology, India   *July 2014 - May 2018*
B. Tech. Electrical and Electronics | CGPA: 8/10 | *Roles: Core team member of IVLABS.*

SKILLS

Programming Languages: Python, C++, MATLAB
Libraries and Tools: OpenCV, PyTorch, TensorFlow, ROS, pytest, gtest, Git, AWS, Docker, DynamoDB, Postman, Jira
Deep Learning Architectures: VGG16, ResNet, DenseNet, HomographNet, SfMLearner, LSTM, LSTM with Attention, SIREN, Transformers, NeRF.

WORK EXPERIENCE

Quidient | *AI/ML Engineer*
Columbia, Maryland   *Feb 2023 - Present*
- Working on **Ne**ural **R**adiance **F**ield based approaches to estimate light fields for an indoor environment.
- Researching the fusion of **physics based** rendering along with NeRF, to accurately reconstruct non-lambertian surfaces.

Apple, Technology Development Group | *Research Intern*
Sunnyvale, California   *May 2022 - Aug 2022*
- Estimated camera pose using **semantic** information with classical and deep learning methods.
- Used **p**erspective-**3**-**p**oints method with **distance transform** for loss function.
- Trained a **Ne**ural **R**adiance **F**ield (NeRF) model on semantic images to learn scene geometry.
- Used **i**nverted **Ne**ural **R**adiance **F**ield approach to estimate pose given an initial guess and a trained NeRF model.

Perception and Robotics Group (PRG) | *Research Assistant*
University of Maryland   *May 2021 - Jan 2022*
- Estimated camera motion using a **D**ynamic **V**ision **S**ensor (event camera) in a static environment.
- Derived a loss function using the brightness constraints to simultaneously estimate optical flow and image gradients.
- Posed this as an optimization problem to estimate a fixed set of parameters that define the flow.

The Hi-tech Robotic Systemz Ltd. | *Research Engineer*
Gurugram, India   *June 2018 - May 2020*
- Implemented **N**ormal **D**istribution **T**ransform matching algorithm for localization on 3D maps.
- Estimated vehicle pose by fusing GPS and localizer pose with IMU data using a **Kalman** filter. (*Video*)
- Developed a server to load maps depending on the vehicle coordinates to increase the computational efficiency by aprox. **50%.**
- Developed and maintained **C++** drivers for Trimble and AN Spatial GPS modules.
- Fused GPS and IMU data to estimate vehicle velocity and reduced the noise by approx. **40%.** (*Video*)

PROJECTS

NeRF++ - Implemented NeRF++ using **PyTorch** to render a 360-degree view. The implementation involves utilizing two NeRF models: one for the background using **inverse spherical coordinates** and another for the foreground. *Details*
Interpretability of video classifier - Studied the problem of identifying both temporally and spatially salient features for a video classification problem. Used *Temporal Saliency Rescaling* methods to get better results. *GitHub*
SfMLearner - Used unsupervised learning approach to estimate depth and egomotion from image sequences. *Github*
Structure from Motion - Reconstructed a **3D scene** and simultaneously obtained the camera poses from a given set of images using their feature points correspondence **(triangulation** and **nonlinear optimization).** *GitHub*
Face Swap - Swapped faces in videos using traditional (Delaunay Triangulation and Thin Plate Spline) and deep learning (Position Map Regression Network) methods. *GitHub*
Auto Pano - Stitched images to create a **panorama** using traditional (Homography estimation using feature points) and deep learning (HomographyNet: Supervised and unsupervised) methods. *GitHub*
Auto Calib - Implemented Zhang's camera **calibration** technique with **non-linear optimization**. *GitHub*
Edge Detection - Detected edges using a simplified version of the probability of boundary detection algorithm. *GitHub*
Lane Detection and Turn Prediction - Detected lanes using curve fitting approach and estimated road curvature. *GitHub*
Depth using stereo vision- Estimated pixel-wise depth using a **sliding window approach**. *GitHub*
Robot Path Planning - Implemented **BFS, DFS, Dijkstra**, **A***, **RRT, RRT*** and **bi-RRT** for holonomic and non-holonomic robots.