Angelo A. Stekardis

2 Townsend Street, San Francisco CA 94107 · angelostekardis@gmail.com · 502-216-5684

PROFESSIONAL SKILLS

Programming Languages: Python, C++, Matlab.
Libraries and Utilities: OpenCV, PyTorch, TensorRT. CUDA, DeepStream, gstreamer, AWS, Docker, Git, Dlib, ZMQ.
Subjects: Computer Vision, Deep Learning, Object Detection and Tracking, Scene Segmentation, Sensor Calibration, Multiview Geometry. Feature Detection and Matching, Sensor Fusion.
Hardware: Image Sensors, Camera Lenses, Inertial Measurement Units, GPS, Nvidia GPUs, Nvidia Jetson Devices, WiFi/LTE.

PROFESSIONAL EXPERIENCE

curbFlow San Francisco, CA
Lead Computer Vision Engineer October 2019 - Present

- Architechted initial versions of hardware and software for curbFlow's edge AIoT device.
- Trained an object detection network (RetinaNet) on curbFlow's custom dataset and achieved an accuracy of 92.12%.
- Optimized object detection network using FP16 quantization and TensorRT to achieve a 5x improvement in inference speed.
- Mapped 2D object detections to global, GPS coordinates using camera intrinsics and extrinsics in addition to geometry.
- Built curbFlow's data collection and labeling pipelines to facilitate algorithm training/testing using tens of thousands of labeled images captured by cameras with novel perspectives on city curb and street space.
- Created a camera calibration software suite to enable non-technical device installers to calibrate up to four cameras at a time.
- Hired, trained, supervised, and managed projects, timelines, and deliverables for the computer vision team.

Iris Automation San Francisco, CA
Computer Vision Software Engineer May 2017 – October 2019

- Integrated inertial measurement unit (IMU) into the Iris hardware/software stack by performing sensor fusion on IMU, camera, and GPS data, by leveraging Kalman filtering and multiview geometry.
- Performed extensive research and testing on various concepts of epipolar geometry in order to significantly improve Iris' geometry-based 2D moving object detection pipeline.
- Researched, implemented, and evaluated real-time 2D object tracking algorithms to create a robust dynamic object tracker.
- Extended 2D object tracker into 3D to understand the trajectories of tracked objects such as planes, drones, and helicopters.
- Quantified overall performance of Iris' collision avoidance software by creating perception system evaluation software.
- Assisted in the effort to select new camera hardware for research and development and future product releases.
- Developed and maintained the internal tools for camera calibration, video labeling, and data visualization.
- Created data summaries and visualizations for distribution to aviation regulatory bodies following product demonstrations.

Faraday Future Los Angeles, CA
Advanced Technology Intern – ADAS & Self Driving May - August 2016

- Implemented image-based localization in Matlab using feature detection/matching and dense optical flow.
- Researched methods for implementing monocular visual odometry using a camera with a limited field of view.
- Created a Matlab GUI to semi-automatically find the GPS position of points of interest from a Google Maps image.
- Implemented a real-time vehicle position estimation algorithm using only yaw rate and velocity (dead reckoning).

EDUCATION

University of Kentucky: Bachelor of Science in Computer Engineering Lexington, KY
GPA: 3.97/4.00 | 7 Dean's List Awards | 2 Computer Engineering Student of the Year Awards August 2013 – May 2017

PERSONAL PROJECTS

Maskify April 2019 – June 2019

- Given two images, one of a face and one of a face mask, automatically position the mask image on the face image.
- Used Dlib to automatically detect facial landmarks on the image of the face.
- Created a "point picker" labeling tool to mark the location of specific facial landmarks on mask images.
- Calculated the homography transform between the labeled landmarks and the detected landmarks to align the two images.
- Runs in real time on an Nvidia Jetson Nano using images streamed from an RTSP security camera.

Fully Autonomous Scooter March 2019 – October 2019

- Built a data collection rig using two USB cameras, an Nvidia Jetson Nano, and my bicycle. Used collected data for qualitative analysis of model performance and to better understand the challenges of slow speed sidewalk/bike lane operation.
- Trained a scene segmentation network (U-Net) on KITTI to determine if the vehicle was operating on sidewalk or road.
- Collected simulated data using the CARLA simulator for quantitative evaluation purposes.
- Conducted interviews with existing scooter users to understand the most critical problems related to shareable scooter usage.