Ali Younis

asyounis.github.io | linkedin.com/in/ali-younis-28702973

Graduating PhD Student focused on computer vision for 3D perception and world understanding with extensive industry experience in computer vision, robotics, deployable machine learning, embedded devices and general software development.

Education

Ph.D. Computer Science (Machine Learning & Computer Vision) — University of California, Irvine
Thesis: Learnable particle systems for computer vision. | Advisor: Erik Sudderth — *Oct. 2020 – June 2025 (expected)*

M.S. Computer Science — University of California, Irvine
Focus: Machine Learning and Artificial Intelligence | GPA: 3.882 — *Sept. 2017 – June 2018*

B.S. Computer Science and Engineering — University of California, Irvine
Deans Honor List (All Quarters) | Magna Cum Laude | GPA: 3.851 — *Oct. 2013 – June 2017*

Experience

Graduate Student Researcher - University of California, Irvine — *Oct. 2020 – June 2025 (Expected)*

Focus: Computer vision for perception, AR/VR, autonomous driving, human mesh reconstruction, tracking, ext. Creating end-to-end learnable particle-based state estimation systems which utilize the latest in neural architectures within Bayesian frameworks for superior accuracy and inference speed on multi-modal time-series data.

Notable Works:

- **Differentiable Particle Belief Propagation** (Current work): Creating end-to-end learnable belief propagation systems for 3D human mesh reconstruction from video. By combining learnable networks with a novel end-to-end differentiable particle based belief propagation algorithm, we can exploit the graphical model structure of the system to enable more accurate 3D human skeleton and shape predictions that maintain multiple modes and faithfully capture uncertainty.
- **3D Gaussian Splatting (3DGS) Simultaneous Localization and Mapping (SLAM)** (Current work): Developing novel end-to-end learnable particle filter based SLAM system using 3DGS as the mapping back-end. Enables precise real-time 3D localization with maps designed for novel view synthesis and 3D world/geometry understanding.
- **Differentiable Particle Filters** [2]: Created first end-to-end differentiable particle filter algorithm with unbiased gradients for computer vision tasks on time-varying (time series) systems with multi-modal data. Application: Real-time indoor robot localization in unseen environments using only RGB camera data and odometry information. Robust to multiple true modes while fusing multi-modal data from various sources.
- **Differentiable Particle Smoothers** [1]: Created the first end-to-end differentiable particle smoother for computer vision tasks on time-varying (time series) systems with multi-modal data. Application: Real-time vehicle localization in large real-world city-scale environments using only RGB camera and odometry sensor data. Achieved 2x accuracy and 3x performance improvements over existing state-of-the-art methods.

Software Engineering Intern (Machine Learning, PhD) - Qualcomm Inc. — *June 2024 – Sept. 2024*

Computer Vision Algorithms Research Group. Developed a novel machine learning based depth-from-stereo method based on differentiable non-parametric loopy belief propagation, trained end-to-end with neural networks tailored for mobile deployment. Assisted in commercialization (mobile deployment) of past work. Implemented training and evaluation pipelines in PyTorch.

Software Engineering Intern (Machine Learning, PhD) - Qualcomm Inc. — *June. 2023 – Sept. 2023*

Computer Vision Algorithms Research Group. Developed a proprietary depth-from-stereo method which blend classical and neural computer vision techniques while leveraging existing custom on-chip hardware accelerators. Method significantly out-performed existing real-time stereo methods (1.5x accuracy improvement). Implemented training and evaluation pipelines in PyTorch.

Software Engineer / Machine Learning Engineer (Part time) - Modal AI — *June 2020 – June 2023*

Developed, trained and deployed (to embedded) real-time depth-from-stereo algorithms (classic and machine learning based) for applications in mapping and collision avoidance. Algorithm development for GPU using OpenCL. Implemented data collection, training, validation, model quantization and deployment pipelines. Integrated various Visual Inertial Odometry systems and created custom camera drivers. Android development. Built custom Linux images. Used C. C++, Python, Java and OpenCL

Software Engineer - Tyvak Nano-Satellite Systems, Inc. — *Aug. 2018 – June 2020*

Designed and implemented mission critical embedded software systems for satellites. Engineer responsible for core software libraries and several spacecraft subsystems (sensors, actuators, control). **Software lead** for all computer vision based camera payloads and for mission with first consumer GPU in space. Platform bring up/Linux OS customization. Continuous integration maintainer. Facilitated inter-team cooperation by holding conflict resolution/sync-up meetings for various high stakes missions. Used C, C++, Java and Python.

Software Engineering Intern - Qualcomm Inc. Corporate R&D *4 Summers from 2014 – 2017*

Designed collision avoidance system for drones using only on-board sensors (RGB/RGB-D/IMU) and processing; 3D mapping, state estimation and planning path to goal region (prototype for 2016 CES demo). Implemented customer facing UART driver. Created continuous integration testing tools. Developed new features for the drone flight stack; safety, GPS navigation, ground station communications and visualizations. Used C, C++, Java, Node.js, Python and Matlab.

Undergraduate Research Asst. - University of California, Irvine *Nov. 2013 – June 2017*

Researcher in the Programming Language Research Group under Professor Brian Demsky. Developed decentralized secure communications system for IoT devices where communication channel is hostile and intermittent [4]. Researched security of IoT devices [5]. Created a bare-metal ARMv7 test platform (real hardware and emulation) for research in robust software execution on unreliable hardware. Implemented lock-free concurrent data structures in C++11 using weak memory models. Wrote device drivers and Android applications. Used C, C++ and Java.

iOS Application Developer – Self Employed *2010 – 2013*

Skills

- **Languages** C, C++, Java, Python, Go, Bash, Matlab, LaTeX, Node.js, SQL, ARMv7 Assembly
- **Frameworks/Tools:** PyTorch, TensorFlow, 3D Gaussian Splats, Keras, TFlite, ONNX, CUDA, OpenCL, OpenCV, Make, CMake, Buildroot, Linux, Android, iOS, Google Cloud, MySQL, Postgres, SLURM, LSF
- **Neural Architectures:** Transformers, Attention, Vision-Transformers (ViT), Convolutional Neural Networks (CNN), Reccurrent Neueal Networks (RNN), Spatial Transformers, Segment Anything (SAM), DinoV2, YOLO (multiple variants), Mono/Stereo Depth Networks, ResNets, UNets, Diffusion Models.

Publications

[1] **A. Younis**, E. Sudderth. "Learning to be Smooth: An End-to-End Differentiable Particle Smoother". Neural Information Processing Systems (Neurips) 2024.

[2] **A. Younis**, E. Sudderth. "Differentiable and Stable Long-Range Tracking of Multiple Posterior Modes". Neural Information Processing Systems (Neurips) 2023.

[3] S. Agarwal, G. Hope, **A. Younis**, E. Sudderth. "A Decoder Suffices for Query-Adaptive Variational Inference". Uncertainty in Artificial Intelligence (UAI) 2023.

[4] R. Trimananda, **A. Younis**, T. KWA, B. Demsky, G. Xu. "Securing Smart Home Devices against Compromised Cloud Servers", Poster at HotEdge, June 2020

[5] R. Trimananda, **A. Younis**, B. Wang, B. Xu, B. Demsky, G. Xu. "Vigilia: Securing smart home edge computing", Symposium of Edge Computing (SEC) October 2018.

Select Side Projects

Autonomous Car Machine Learning 3D Perception System (Ongoing)
Python pipeline to interface with the CARLA simulator to test various camera-only 3D perception methods for autonomous vehicles such as 3D point cloud generation and semantically labeled 3D point clouds. 3D point clouds are estimated using neural depth-from-stereo models (trained from scratch) and labeled via modern semantic segmentation networks to enable full 3D world understanding. Full custom simulation, training and visualization pipelines in PyTorch.

Video Game Computer Player (Game: "1010!")
Create AI algorithm for playing "1010!" (a 2D variant of Tetris on iOS). AI player used custom highly optimized CUDA engine back-end to evaluate all possible moves. Full game (including UI) replicated on desktop.

Awards and Honors

- **Roberto Padovani Scholarship:** Awarded to select interns for outstanding performance over summer.
- **Qualcomm Qualstar Award (multiple times):** Awarded for excellent team work while at Qualcomm.

Other Experience

- Reviewer at Neurips 2024.
- In charge of research labs compute cluster (administration, upgrading hardware, coordination).
- Teaching assistant for Various Machine Learning Courses: CS 178, CS 275P.