Ali Younis

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

Education

Ph.D. Computer Science (Machine Learning & Computer Vision) — University of California, Irvine
Focus: Learned particle based 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 – Present*

Focus: End-to-end learnable (deep learning, neural networks) particle based belief propagation systems that enable world understanding for computer vision applications in robotics, AR/VR, autonomous driving, tracking, human mesh reconstruction and other state estimation tasks. Used recent neural networks architectures: convolutional neural networks (CNN), recurrent neural networks (RNN, LSTM, GRU), Transformers/Attention, Resnet, VGG, Spatial Transformers, Segment Anything (SAM), ext.

Notable Work and Projects:

- **Differentiable Particle Loopy Belief Propagation** (Current Work): Creating an end-to-end learnable (loopy) belief propagation system with multi-modal predictions using RGB camera data for various computer vision tasks (human mesh reconstruction, tracking, depth estimation, ext). Used deep learning (neural networks) within the belief propagation algorithm and trained end-to-end on collected data. Applications include computer vision, robotics, AR/VR, ext.
- **Differentiable Particle Filters** [2]: An end-to-end learnable state estimation technique for computer vision based state localization and tracking for time varying systems. By combining classical particle filtering, neural networks and a novel differentiable particle resampling method, we created the first unbiased, end-to-end learnable particle filter method. Produces multi-modal predictions for various tasks using image data (RGB) and other sensors (odometry, angles).
- **Differentiable Particle Smoothers** [1]: An end-to-end learnable computer vision based state estimation technique for time varying systems. Created a the first end-to-end differentiable particle smoother and applied it to localization in a city-scale environment using only hand-held RGB images and semantic maps (like Google/Apple maps). Method estimates a Birds-Eye-View from hand-held RGB images before applying differentiable particle smoothing.

Machine Learning Intern - Qualcomm Inc. (Computer Vision Algorithms Research) — *June 2024 – Sept. 2024*

Worked on mobile deployable computer vision. Developed novel machine learning based depth-from-stereo methods tailored for mobile deployment with existing hardware accelerators. Designed neural network based non-parametric belief propagation systems. Assisted in commercialization of past work. Implemented fully custom training and evaluation pipelines in PyTorch.

Machine Learning Intern - Qualcomm Inc. (Computer Vision Algorithms Research) — *June. 2023 – Sept. 2023*

Developed proprietary depth-from-stereo methods which blend classical and neural computer vision techniques while leveraging existing custom on-chip hardware accelerators. Method significantly out-performed existing internal and public real-time stereo methods. Implemented fully custom 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 computer vision) for applications in mapping and collision avoidance. Implemented data collection, training, validation, model quantization and deployment pipelines. Integrated various Visual Inertial Odometry systems. Integrated custom cameras using the low level USB stack. Android development. Built custom Linux images. Used Python, C, C++ and custom OpenCL.

Software Engineer - Tyvak Nano-Satellite Systems, Inc. — *Aug. 2018 – June 2023*

Designed and Implemented mission critical embedded software systems for satellites. Lead software engineer for core software libraries and several spacecraft subsystem software (sensors, actuators, control software). Software lead for all computer vision based camera payloads and primary software engineer for mission with first commercial GPU in space. Platform bring up and 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 and suites for drones. Developed new features for Qualcomm's flight stack (Safety/GPS navigation/Ground Station Communications/Sensing). Developed interface for remote drone control and live data visualization using Node.js back-ends. 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*

Designed and developed various iOS apps with over 100K downloads worldwide. Developed a arcade style zombie game with custom physics engine. Developed app that syncs music playback between various devices using only audible cues.

Skills

- **Languages** C, C++, Java, Python, Go, Bash, Matlab, LaTeX, Node.js, SQL, ARMv7 Assembly
- **Frameworks/Tools:** PyTorch, TensorFlow, Keras, TFlite, ONNX, CUDA, OpenCL, OpenCV, Make, CMake, Buildroot, Linux, Android, iOS, Google Cloud, MySQL, Postgres, SLURM, LSF

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

Automatic AI TikTok Video Generator

Developed python pipeline to generate TikTok videos automatically using various AI tools. Used various APIs to integrated data sources to get content (background images/videos and script material). Integrated AI voice generators APIs to create voice over and AI speech-to-text tools to sync subtitles to audio. Video stitched together using python video libraries. (Never uploaded videos to respect copyright).

3D Gaussian Splatting Depth from Stereo

Attempted to extract depth-from-stereo using 3D Gaussian splatting to allow for a temporally consistent depth estimate. Modified existing Gaussian splatting frameworks and implemented custom pipelines.

Video Game Computer Player (Game: "1010!")

Created a computer player that can play the game "1010!" (a 2D variant of Tetris on iOS). Replicated the iOS game on desktop, including graphics. Computer player used various heuristics to score moves and a custom CUDA based game engine back-end to quickly evaluate all possible moves before selecting best one. CUDA implementation heavily optimized.

Awards and Honors

- **Roberto Padovani Scholarship:** Awarded to select interns for outstanding performance over summer.
- **Qualcomm Qualstar Award (2 times):** Awarded for excellent team work while at Qualcomm.