JhihYang Wu

520-534-6033 | jhihyangwu@arizona.edu | jhihyangwu.github.io | www.linkedin.com/in/jhihyang-wu/

Education

University of Arizona | Tucson, AZ August 2021 - May 2025

Bachelor of Science in Computer Science & Electrical and Computer Engineering (Double Major) GPA: 4.00/4.00

Skills

Areas: Deep Learning, Computer Graphics, 2D/3D Computer Vision, LLMs, Algorithms, Compilers, Computer Architecture, OS, Networking, Databases, Data Science

Languages: English, Mandarin

Programming Languages: Python, C++, C, Java, C#, HTML, CSS, JavaScript, MIPS Assembly, Verilog, Oracle SQL, Bash, GLSL, MATLAB, CUDA

Tools and Frameworks: PyTorch, Vivado, Django, Keras, Unity, Blender, NumPy, Matplotlib, OpenCV, KiCad, Fusion 360, Git, Docker, Vim, unittest, Linux, OpenGL, WebGL

Experience

xAI - Software Engineering Specialist January 2025 - Present

- Improving Grok's coding capabilities by designing and curating high-quality software
- Applying knowledge in data structures and algorithms, operating systems, math, and much more

ASML - Senior Design [Pictures] September 2024 - Present

- Developed from scratch an optics simulation software for our Shack–Hartmann wavefront sensor setup using C++ and OpenGL

Undergraduate Research Assistant February 2024 - January 2025

- Conducted research on NeRFs, Diffusion Models, 3D Reconstruction, 3D Gaussian Splatting
- Replicated and extended the GeNVS paper by Stanford and NVIDIA for SAR, Sonar, and IR applications
- Co-authored CrossModalityDiffusion accepted to WACV 2025 GeoCV: https://arxiv.org/abs/2501.09838
- Working with Professor Abhijit Mahalanobis and PhD students

Projects

miniRT [GitHub] [Pictures] May 2024 - May 2024

- Powerful ray tracer I built from scratch using just C++ and math
- Supports advanced features (Möller–Trumbore intersection, octree, multithreading, BRDFs, Snell's law, etc) to generate realistic images efficiently

minigrad [GitHub] June 2023 - July 2023

- PyTorch (very popular deep learning framework) clone from scratch using just Python and NumPy
- Implemented an autograd engine so users only have to define the forward pass and my library will handle backpropagation automatically by creating a computation graph and storing operations and contexts
- Used my own library to train a CNN to classify MNIST digits, a LSTM to classify which language a last name is in, and an Attention model to translate dates into standardized format

C Compiler January 2024 - May 2024

- Compiler coded in C that translates simplified version of C language into MIPS assembly
- Implemented both the front-end (lexical analysis, syntax analysis, semantic analysis, AST, three-address code) and back-end (code generation) of the compiler