Prabhman Dhaliwal

prabhmandhlwl@gmail.com 916-806-2838 https://prabbydd.github.io/

Work Experience

UC Davis Medical Center **Davis, CA**
Volunteer Researcher *September 2023 - Janurary 2024*

- Created potential GenAI applications to reduce time to analyze cancer in patients from days to hours

Amazon Astro **Sunnyvale, CA**
Software Engineer, Computer Vision/Deep Learning and Robotics *July 2022 - Janurary 2023*

- Reduced latency of simple object detection models by ***6-7%*** using quantization, C++ threading, delegation to robot hardware GPUs, and TFLite optimized models
- Developed ML infra (ROS/TFLite) to run models on x86/ARM-NEON/Linux/Android
- Constructed manual and unit tests (gtest and C++) to analyze perception and mobility on Astro
- Augmented robots CV algorithms to identify objects regardless of orientation using neural networks

Education

University of California, Berkeley **Berkeley, CA**
BA: Data Science, Robotics Emphasis *August 2017 - July 2021*
Minor: Electrical Engineering & Computer Science

- Relevant Coursework: Computer Vision, AI/ML, Controls, Digital Signal Processing, Linear Algebra, Graphics, Mechatronics, Data Structures and Algorithms, Operating Systems

Projects

PokeRogue Open Source Game Developer in Typescript *July 2024- Present*

- Added new gameplay features such as new moves/abilities and quality-of-life changes that improved gameplay experience of ***over 10000 daily players***
- Completed bugfixes that routinely negatively impacted the experience of ***over 10000 players***. Debugging was done via manual testing and the browser debugger, and wrote official unit tests using Jest

Custom Compiler in C++ *July 2024*

- Designed compiler in C++, with floating/int arithmetic, if-else, variables, pointers, function scopes, etc.
- Implemented precedence climbing for arithmetic ops. which is ***10-20% faster*** than base C++ precedence algorithm

Custom Raytracer in C++ *March 2024*

- Designed and created a ray tracer in C++ from scratch. Implements basic materials (dielectric, metal, matte, etc.)
- Implemented antialiasing, threading, and BVH which sped up the simple raytracer by ***over 90x*** for scenes with thousands of objects

RL Controlled Bipedal Robot in C++ *Spring 2021*

- Improved the stability/power efficiency in an under-actuated three-link bipedal robot using 3-DOF reaction wheels
- Experimented with RL controllers and compared with PID and LQR controllers

Custom Video Compression Scheme in Python *Spring 2021*

- Implemented custom image compression algorithm to send best possible quality GIF in no more than 10 kB while optimizing signal-to-noise ratio ***(1st place in competition of 50 people, 66x compression)***
- Sparsified image with SVD/DCT, downsampled color based on eigen information using LZMA compression, and finally blurred image before sending packets with APRS; recovery was done by interpolating each frame

Additional Projects *2022 - 2024*

- Used threads and AVX SIMD extensions to achieve ***66x*** time improvement on Mandelbrot calculations
- Implemented Conflict Based Search to plan paths for multiple agents using A* as a subroutine
- Implemented multi-view 3D reconstruction algorithms to match images (RANSAC, SIFT)

Technical Skills

- C++, Python, ROS, Typescript/Javascript, TensorFlow, Linux, Javascript, OpenCV, gRPC, TCP