Zhihao Ruan

3131 Walnut St Apt 553, Philadelphia, PA 19104
ruanzh@grasp.upenn.edu • +1 (734) 881-4112 • https://zhihaoruan.xyz

Education

University of Pennsylvania — Philadelphia, PA
- *Master of Science in Engineering in Robotics* — *May 2022*
  General Robotics, Automation, Sensing & Perception (GRASP) Laboratory
  - **Selected Coursework:** Introduction to Optimization, Advanced Topics in Machine Perception, Interactive Computer Graphics

University of Michigan — Ann Arbor, MI
- *Bachelor of Science in Computer Science Engineering,* ***GPA: 3.9/4.0*** — *Sept 2018 – May 2020*
  College of Engineering
  - **Selected Coursework:** Embedded Systems Design (grader), Machine Learning (grader), Computer Vision, Autonomous Robotics, GPU Programming & Architecture, Operating Systems

Shanghai Jiao Tong University — Shanghai, China
- *Bachelor of Science in Electrical and Computer Engineering,* ***GPA: 3.6/4.0*** — *Sept 2016 – Aug 2020*
  University of Michigan-Shanghai Jiao Tong University Joint Institute (UM-SJTU Joint Institute)
  - **Selected Coursework:** Electromagnetics, Signals and Systems, Analog Circuits, Semiconductor Devices

Scholarship and Honors

James B. Angell Scholar — *Mar 2020*
University of Michigan Honors — *Dec 2018, May 2019, Dec 2019*
University of Michigan "Dean's List" — *Dec 2018, Apr 2019, Dec 2019*
SJTU Undergraduate Excellent Scholarship — *June 2017, June 2018*
UM-SJTU Joint Institute "Dean's List" — *June 2017, Dec 2017, June 2018*
"Honorable Mention" of 2017 Interdisciplinary Contest in Modeling — *Apr 2017*

Research Experience

- **Interactive Sensing and Computing (ISC) Lab** — Ann Arbor, MI
  Research Assistant of Prof. Alanson Sample, Synthetic Health Sensor — *Jan 2019 – Dec 2019*
  - Built an embedded system in **C & MATLAB** with **STM32** microprocessor and Panasonic's GridEye® 8×8 IR sensor through **I²C & UART** that can collect, detect and visualize heat distribution in the room.
  - Constructed **a complete C API** for Panasonic's GridEye® 8×8 IR sensor for **STM32** microprocessor from scratch.
  - Implemented **Direct Digital Synthesis (DDS) in C** of a frequency-sweep ultrasonic sine wave from 39kHz to 41kHz with STM32 microprocessor and ultrasonic transducers.
  - Implemented two different ultrasound distance measurement algorithms in **Python** including **FMCW (Frequency-Modulated Continuous Wave) algorithm** and **phase-based ranging algorithm** with STM32 microprocessor.

Project Experience

- **Mini Minecraft in C++** — Philadelphia, PA
  CIS 560: Interactive Computer Graphics, Univ. of Pennsylvania — *Nov 2020 – Dec 2020*
  - Achieved mesh rendering with **interleaved Vertex Buffer Object (VBO)** data in Qt and OpenGL and applied **CPU parallel optimization**.
  - Created **procedural weather** effect in GLSL shader program, **L-system** river generation, **procedural terrain** generation with **Perlin Noise** and terrain smoothing with **Fractal Brownian Motion (FBM)**.
  - Implemented mesh collision checking with **ray tracing** and **3D grid marching**.
  - Created and implemented **animated texturing** for "grass," "dirt," "stone," "ice," "water," "lava," etc.
  - Implemented **redstone texturing & redstone circuit logic**; supports "redstone wire," "power lever," "redstone torch".

- **Deep Neural Network Implementation for Machine Perception** — Philadelphia, PA
  CIS 680: Advanced Topics in Machine Perception, Univ. of Pennsylvania — *Sept 2020 – Dec 2020*
  - Implemented **YOLO v1** for object detection in **PyTorch**, featuring anchor-based detection & **Non-Max Suppression (NMS)**.
  - Implemented **SOLO** for instance segmentation, using **ResNet50 FPN** as backbone and **MatrixNMS**.

- Implemented **Regional Proposal Network (RPN)** and **FasterRCNN** object detection head using ResNet50 FPN as backbone, featuring **ROI Pooling**, reaching an **mAP of 0.581**.
- Implemented multiple GAN architectures including **Variational Auto-Encoders (VAE)**, **DC-GAN**, **CycleGAN**, **BicycleGAN**. Achieved **Fréchet Inception Distance (FID)** score of 70.87 on CycleGAN and 76.82 on BicycleGAN.

- **Computer Graphics Development with OpenGL** — Philadelphia, PA
  CIS 560: Interactive Computer Graphics, Univ. of Pennsylvania — *Sept 2020 – Nov 2020*
  - Implemented **3D mesh rasterization** with provided mesh file in **Qt 5.15** and **OpenGL**, integrated with **scene graphs**, with polar spherical camera model, perspective-correct **barycentric interpolation**, texture mapping, Lambertian reflection, and custom-scale **anti-aliasing**.
  - Implemented various shaders in **GLSL** including **Blinn-Phong reflection** shader, **Worley noise** shader, Gaussian blur shader, and **Matcap** shader.
  - Built an deformable/editable mesh with **Half-edge mesh structure**, featuring a Maya-like application interface with Qt and **Catmull-Clark surface subdivision**.
- **Real-Time On-Device Flow Statistics Detection and Prediction** — Shanghai, China
  Undergraduate Major Design Experience, UM-SJTU Joint Institute — *June 2020 – Aug 2020*
  - Built a system in **Python** which detects human traffic flow, automatically analyzes & detects entrances on **Raspberry Pi 4B**, stores data on a server, visualizes analyzed data on a self-designed front-end website **in real time** ($\sim$ 15 FPS with Google Coral Edge TPU USB accelerator).
  - Achieved **20 FPS and 90% accuracy** object tracking & people counting with **self-designed Kalman filter tracker**, automatic entrance detection with density-based clustering algorithm — **DBSCAN** in Python.
- **Linux Infrastructure Implementation on x86 PCs** — Ann Arbor, MI
  EECS 482: Introduction to Operating Systems, Univ. of Michigan — *Jan 2020 – Apr 2020*
  - Implemented mutex, conditional variables & thread class using context switching API in **Linux kernel library** with multiprocessor support provided using **interrupt handling** and **CPU guard** in C++.
  - Built Unix pager with swap-backed & file-backed pages with **page fault handing** and **process switching & forking** support in C++.
  - Emulated Unix-style file system using inode and directory entry structures with client request handling using **Berkeley sockets** and safe concurrency using **C++ Boost library**.
- **MXNet Optimization with GPU** — Ann Arbor, MI
  EECS 498: GPU Programming & Architecture, Univ. of Michigan — *Nov 2019 – Dec 2019*
  - Implemented GPU parallelization of forward kernel of MXNet incubator 1.3.x with **tiled matrix multiplication** and **GPU shared memory**; boosted inference speed **from 30s to $<$ 3s** on fashion-mnist dataset and Nvidia TITAN Xp graphics card.
- **DOAPP: Dynamic Object Avoidance and Path Planning** — Ann Arbor, MI
  Undergraduate Major Design Experience, Univ. of Michigan — *Oct 2019 – Dec 2019*
  - Implemented a **GPU-accelerated motion planning algorithm** originally proposed by Chonhyon Park, et al. in **ROS C++ & CUDA** with Nvidia GPU parallel programming & optimization that could perform **real-time obstacle avoidance**.
  - Built a controller and trajectory follower in **ROS C++** for Dynamixel motors on robot arm from scratch and achieved **30 Hz signal transmission**.
- **Robotics Algorithm Development in LCM** — Ann Arbor, MI
  EECS 467: Autonomous Robotics, Univ. of Michigan — *Sept 2019 – Oct 2019*
  - Implemented a SLAM algorithm in C++ with **occupancy grid mapping, particle filter localization, A* path planning.** Entire system run on a mobile robot with Raspberry Pi 3 and LiDAR.
  - Implemented robot manipulation algorithms in Python including **Forward Kinematics, Inverse Kinematics (based on Geometry).** Entire system run on a mobile robot with robot arm that could localize & pick up a block using AprilTag automatically.
- **Cost-Function Prediction Market Simulation with Bayesian Traders** — Ann Arbor, MI
  ML Research Paper Reading Group of Prof. Sindhu Kutty, University of Michigan — *May 2019 – Aug 2019*
  - Simulated **cost-function based prediction market** mechanism **in Python**, with its performance evaluated and compared with traditional machine learning algorithms.
  - Reconstructed exponential-family prediction markets in different probability distributions **mathematically**.
- **Interactive Game: *Step on White Tiles*** — Ann Arbor, MI
  EECS 373: Introduction to Embedded Systems Design, Univ. of Michigan — *Mar 2019 – Apr 2019*
  - Visualized black & white tiles flow by driving a projector with **FPGA** by programming **VGA protocols in Verilog**.
  - Decoded signals from Nintendo controller in Verilog.
  - Built **a complete menu selection user interface in C** on an LCD display with SmartFusion® microprocessor and Nintendo controller.
  - Achieved stepping detection on projected tiles through **SPI protocol in C** with Pixy® camera.
  - Enabled sound effects using SmartFusion® microprocessor, Adafruit® Audio Sound Board and Dell® stereos in Verilog.

Skills

Programming Languages: C/C++, Python, MATLAB, Verilog.
Development Tools: STM32CubeMX, OpenCV, PyTorch, Scikit-Learn, LCM (Lightweight Communications and Marshalling), ROS (Robotics Operating System), CUDA