

XUAN (TOM) ZHAO

Tel: Removed due to spam calls ◇ Email: xz1919@ic.ac.uk ◇ Website: tomzhao.me

Github: github.com/zhao-xuan ◇ LinkedIn: [linkedin.com/in/zhaoxuan0914](https://www.linkedin.com/in/zhaoxuan0914)

EDUCATION

Imperial College London

September 2019 - June 2023

Year 4 MEng student at [Department of Computing](#)

[First Class Honour](#) in all years (77.84%)

Thesis Title: High-Fidelity Image Synthesis from Pulmonary Nodule Lesion Maps using Diffusion Models

Year 1/2 Topics: Calculus, Linear Algebra, Algorithm I/II, Probability & Statistics, Computer Architecture, Database, Haskell, Java, C, Operating System, Compiler, Software Design, Network & Communication

Year 3/4 Topics: Intro to Machine Learning, Advanced Computer Architecture, Graphics, Computer Vision, ML for Imaging, Robotics, Reinforcement Learning, Deep Learning, Natural Language Processing

PUBLICATION

High-Fidelity Image Synthesis from Pulmonary Nodule Lesion Maps using Semantic Diffusion Model. Xuan Zhao, Benjamin Hou

MIDL 2023 Short Paper Track Submission

- In this paper, we explore the use of Semantic Diffusion Models (SDM) to generate high-fidelity pulmonary CT images from segmentation maps and perform quantitative evaluation of the quality of images using performance of two downstream models and FID score with comparison of SPADE-generated images.

WORK EXPERIENCE

Microsoft UK

April 2022 - Sep 2022

Software Engineer at [Azure for Operators Team](#)

Arm Ltd.

July 2021 - March 2022

Linux Kernel Engineer in [Morello Kernel Team](#)

Adaps Photonics

Sep 2020 - Jan 2021

Linux Kernel Engineer & Software Engineer

Pulse Secure

July 2020 - September 2020

Software Engineer & UI Testing Engineer

ACHIEVEMENTS

Imperial Computing [Year 3 Dean's List](#)

2022

IC Hack 22: [Best Entertainment Hack Prize](#)

2022

Year 3 [Computer Graphics Ray Tracing Prize](#)

2022

Year 3 Robotics Best Modified Robot Prize

2022

LeetCode Cup Group Contest: Top 100/3.4%

2021

LeetCode Cup Solo Contest: Top 500/4.7%

2020

[Imperial Year 1 Best Overall C Group Project](#)

2020

PROJECTS

Image Processing Engine

Third Year Group Project

Desktop Image Processing Software Built for Research Purposes

github.com/G14-Y3/IPEwG

- Developed several advanced features such as **neural style transfer**, **depth estimation**, **steganography**, **neural-network-based denoise**, and **false coloring**. Trained 6 different neural style transfer models using Pytorch with cuda, and used **jit tracing** to convert pth models into pt models.
- Used **Pytorch** to train/trace the denoise, depth detection models and used **lib-torch** to load the model in Kotlin and perform forward pass. Used **mmdnn** to convert Tensorflow/Keras model to Pytorch model.
- Developed an image processing software in Kotlin and used **JavaFX** as the GUI framework.

Tetris++

First Year C Group Project Extension

Human or AI Playing Tetris Using Accelerometer

github.com/zhao-xuan/TetrisPlusPlus

- Responsible for building the AI using **genetic algorithm** and fine-tuned it for both a conservative AI and a risky AI. Implemented another AI using **Q-learning algorithm** as well. Used **ncurses** to build the command line interface.
- Implemented **Raspberry Pi** version using **C**. People can play it using different sensors on an LED screen.

MTTKRP Program Optimization on Raspberry Pi Advanced Computer Architecture Coursework
MTTKRP: Matricized Tensor Times Khatri-Rao Product [Link to the report](#)

- Inspected the effect of **compiler optimization**, **cache usage pattern** in the program, **SIMD optimization** on Arm Cortex A72, **loop unrolling + loop fusion**, **hardware multi-threading** and **overclocking** on the execution time of the MTTKRP program.
- The experiment is conducted on a Raspberry Pi with Arm Cortex A72, with a systematically designed strategy and carefully selected steps to carry out the optimizations.

Investigation on Instruction Level Parallelism Advanced Computer Architecture Coursework
in the Fast Fourier Transform Program [Link to the report](#)

- Investigated on the effect of changing the configurations of an **out-of-order execution CPU** on a simulator, including experimenting with different sizes/configurations of the RUU (Register Update Unit), **branch predictor**, **load/store queue**, L1/2 caches, the number of ALUs, and the number of instructions fetched/issue/decode.
- The goal is to develop a systematic strategy on quickly **identifying the configuration that produces the lowest energy cost** while maintaining a reasonable instructions-per-cycle speed.

Magic Wand ICHACK 22 Best Entertainment Hack
A gesture-based cursor control application github.com/danieldeng2/idiurus

- It's a **gesture-based cursor control** application that allows you to operate your devices without using a trackpad or a mouse. You can open/close a tab in Chrome, or drag a window around using a click and drag gesture, as well as scrolling a webpage and even take a screenshot.
- Developed it in **Python** using the **mediapipe** library to detect and map interest point of a hand, **pynput** to control the mouse, and **QtforPython** (or **PySide6**) to write nice GUI window.

Pintos Second Year Group Project
A Simple Operating System implemented Using C github.com/zhao-xuan/pintos_45

- Developed a simple operating system with three other group members. Implementation includes a **Round-Robin process scheduler** and a **priority-based scheduler** based on CPU time of each process, the synchronization mechanisms (including **lock**, **semaphore**, and **monitor**).
- Enabled execution of **user-space programs** and several standard Unix system calls, such as **open()/close()**.
- Implemented **virtual memory** and page management, including **page allocation**, **copy-on-write** and **shared-pages**, **memory-mapped files**, and **page swapping** based on **Second-Chance algorithm**. All implementations choices related to memory management are original by our team.

WACC Compiler Second Year Group Project
A WACC Language Compiler Implemented Using Java blog.tomzhao.me/?p=572

- Implemented a WACC language compiler capable of generating both **Armv6/Arm11** architecture assembly and **Intel x86-64**. WACC is a simple language whose syntax is similar to C/Pascal.
- Used **ANTLR** tool to generate the language parser and used the generated parser to construct syntax tree, during which **symbol table** is constructed and **syntax and semantic checks** are performed, as well as the generation of error messages if any **syntax or semantic errors** occur.
- Supported basic control flows, including **if-else**, **for/while** loops, **switch**, **break**, and **continue**. It supports basic, pointer, and **struct** types, as well as functions and **nested functions**. It also supports **import**, native **List/Map/Set**, **system calls** (**read**, **print**, **exit**, etc.) and **-O1** level optimization.

ARM11 Emulator/Assembler First Year C Group Project
Running four basic types of ARM11 instructions on x86-64 machines blog.tomzhao.me/?p=572

- Built **ARM11 emulator** and **assembler** that support four basic types of **Armv6/ARM11** instruction set.
- Used function pointer, macro functions in both emulator and assembler. Implemented a **decoder** to decode binary executable **hash-table** and **tokenizer** to convert assembly lines to binary codes in assembler.
- Used advanced **Makefile** techniques such as **MMD** flag that drastically reduced bugs and development time. Fully utilized **git** issues/merge requests and other git features for group work.