

Tony Yiding Tian

Philadelphia, PA | (267)249-1202 | tonygt@seas.upenn.edu | tonyxtian.com

EDUCATION

University of Pennsylvania - School of Engineering and Applied Sciences

Philadelphia, PA

B.S.E. in Computer Engineering & Accelerated M.S.E. in Computer Graphics

May 2027

- GPA: 3.78 — Relevant Courses: GPU Programming, Machine Learning, Operating Systems Design & Implementation, Data Structure & Algorithm, Advanced Rendering, Computer Animation, Embedded Systems

PROJECTS

MatForge - Advanced Material Rendering System | *Vulkan, Slang, C++*

Nov 2025 – Dec 2025

- Architected production-quality GPU path tracer implementing four SIGGRAPH papers (2023-2024) in unified Vulkan ray tracing pipeline, contributing 2,400+ lines of C++ and Slang code for sampling and geometry systems
- Implemented Quad-Optimized Low-Discrepancy Sequences (QOLDS) from SIGGRAPH 2024, engineering base-3 Sobol' generator with Owen scrambling across 47 dimensions, achieving +2.57 dB PSNR and 44.7% MSE reduction vs PCG sampling at 512 SPP with < 1% performance overhead
- Developed RMIP (Rectangular MinMax Image Pyramid) intersection shader, enabling tessellation-free displacement mapping through hierarchical texture-space ray traversal with custom Vulkan intersection shaders
- Integrated techniques into complete Monte Carlo path tracing pipeline supporting glTF 2.0 scenes, HDR environment maps, and KHR_materials_displacement extension for physically-based rendering
- GitHub Repo: github.com/MatForge/MatForge

CUDA Path Tracer - 3D PBR Renderer | *CUDA, GLSL, C++*

Sep 2025 - Oct 2025

- Monte Carlo path tracer capable of rendering complex 3D scenes with custom 3D models and environment maps
- Implemented shading BSDF kernel supporting global illumination, multiple importance sampling, anti-aliasing, sub-surface scattering, capable of rendering various PBR material types with albedo and texture maps
- Integrated third-party libraries of tinyGLTF to support glTF 2.0 mesh loading and Nvidia OptiX for denoising
- Utilized various techniques to boost performance: material sorting (+5%), Russian Roulette (+6% - 24%), stream compaction (+24% - 67%), and Bounding Volume Hierachy (3× - 160× framerate in complex scenes)
- Project Repo and Demo: github.com/tonygt/CUDA-Path-Tracer. A previous standalone performance focused stream compaction project with detailed analysis in Nsight: github.com/tonygt/Project2-Stream-Compaction

PennOS - UNIX-like Operating System | *C, Shell, Kernel*

Mar 2025 – May 2025

- Architected and implemented a complete user-level operating system in C with team of 4, featuring 8000+ lines of systems code with full process lifecycle management
- Designed Process Control Block (PCB) data structure managing 50+ concurrent processes with metadata including PID allocation, priority levels, parent-child relationships, signal handling, and user/kernel stack management
- Implemented preemptive multi-level priority scheduler supporting 3 priority levels with Round Robin time-slicing (10ms quantum), preventing starvation through priority aging and achieving 95% CPU utilization
- Built POSIX-compliant interactive shell supporting 15+ built-in commands (ps, kill, jobs, fg/bg), I/O redirection, pipeline chaining, and batch script execution with robust error handling
- Developed comprehensive system call interface including p_spawn(), p_wait(), p_sleep(), signal delivery (SIGTERM, SIGSTOP, SIGCONT), and file I/O wrappers with proper resource cleanup and zombie process reaping
- Implemented job control and signal handling system supporting background process management and graceful process termination across parent-child hierarchies

EXPERIENCE

Linux Kernel Policies Research Assistant - PURM Scholar

May 2025 – Aug 2025

Learning Directed Operating System (LDOS), Prof. Sebastian Angel

Philadelphia, PA

- Developed eBPF-based kernel monitoring infrastructure collecting real-time TCP networking metrics retrieved from 5 crucial kernel tcp functions of tcp_v4_rcv, v4_connect, state_process, congestion_control, and cubic
- Engineered high-performance data analysis pipeline processing 10,000+ TCP state transitions per second, identifying critical performance bottlenecks in kernel networking policies and congestion control algorithms
- Architected and contributed 2000+ lines of C and Python code to open-source KernMLOps repo, implementing kernel probing infrastructure used by 15+ researchers. GitHub Repo: github.com/tonygt/KernMLOps

TECHNICAL SKILLS

Programming: CUDA, C/C++, WGS, GLSL, Parallel algorithms, Memory management, Rendering pipeline

Graphics/Rendering: Nsight Profiling, Path Tracing, Deferred Rendering, Rasterization, Animation systems, PBR

Tools/APIs: Nvidia Nsight, WebGPU, Vulkan, Visual Studio, Qt, OpenGL, WebGPU, Git, CMake, Clang, GDB