

Kyle Lukaszek

Profile

Master's student in Computer Science with expertise in GPU-accelerated systems, scientific visualization, and machine learning optimization. Published researcher combining low-level programming with quantitative methods. Experienced in CUDA development, cross-platform drivers, and high-performance tensor operations. Strong foundation in both theoretical research and practical systems implementation.

Skills

Languages: C/C++, C#, Python, Haskell, TypeScript, C, CUDA, WGSL, GLSL, HLSL, Slang, OpenCL
APIs: WebGPU, Metal, Vulkan, OpenGL, WASM, SQL, PyTorch, JAX
Tools: GDB, Valgrind, Perf, RenderDoc, NSight, Git, Nix, Unity, Visual Studio, Xcode
Spoken: English (Native), French (Immersion, DELF B2)

Experience

- Jan 2024 – **Research Assistant, Human-Computer Interaction, University of Guelph**
Jan 2025
- Developed a cross-platform C driver and CLI tool for the JETI Spectral 1501, replacing a Windows-only FTDI driver with a low-latency scriptable interface.
 - Optimized serial read/write routines and built tensor-style polling strategies for consistent spectro-radiometric data acquisition.
 - Reduced latency and setup time by over 50%, enabling seamless integration into HCI and perceptual testing pipelines.
 - Integrated iOS SensorKit with custom ambient light models for iOS-based perceptual validation.
 - Co-authored a peer-reviewed paper accepted to Graphics Interface 2024.
- May 2023 – **Research Assistant, Machine Learning, University of Guelph**
Sept 2023
- Implemented and optimized low-level CUDA kernels for NLP inference and data preprocessing tasks on large-scale Twitter corpora.
 - Engineered multiple high-throughput ML/IR pipelines in PyTorch, SciKit, and C/CUDA, reducing data processing time by over 3×.
 - Performed performance profiling with NSight and integrated batch-level optimizations.
 - Results later supported an [IEEE T-CSS publication](#) on online hate detection.
- Jan 2025 – **Teaching Assistant, CIS*2750, University of Guelph**
Apr 2025
- Supported instruction in C systems development, SQL, and FFI integration for TUI development.
 - Mentored a class of 300+ undergraduate students in intermediate-advanced programming concepts.
 - Graded and provided feedback to students based on their submitted coursework.

Education

- May 2025 – **Master of Science in Computer Science, University of Guelph**
Present
- Thesis research in efficient rendering of tensors for scientific visualization.
 - Supervised by Dr. Denis Nikitenko and Dr. David Flatla
- Sept 2020 – **Honours Bachelor of Computing, University of Guelph**
April 2025
- Area of Application in Mathematics.
 - Focused coursework in algorithms, compiler theory, machine learning, computer graphics, graph theory, differential equations, and POSIX systems programming.
 - Average: 80%

Publications

- GI 2024 **Modelling The Effects of Bright Environments on Colour Perception**, *Graphics Interface 2024*, Halifax, NS
- Presented at *Graphics Interface 2024*, Halifax.
 - Co-authored with Dr. Denis Nikitenko and Dr. David Flatla's lab.
 - Explored the intersection of optics, experimental research, and perceptual HCI.
 - [Read the paper here](#) ([for free](#))

Selected Projects

SDL3 Native Haskell bindings for the SDL3 Library

- Haskell Bindings**
- Enables functional programming access to the entirety of SDL3's API.
 - High-performance type-safe abstractions over SDL3's C interface.
 - 30+ cross-platform examples, including several GPGPU and rendering demos.
 - Supports DirectX12, Vulkan, and Metal for cross-platform application development.

SDL3 Feasibility study of a WebGPU backend for SDL3

- WebGPU Experiments**
- Prototyped a [6K-line C](#) backend for WebGPU, enabling both in-browser (WASM) and native examples for specific uses.
 - Achieved 10 GitHub stars as a proof-of-concept during active development.
 - Currently on hold—upstream SDL wants to rework its GPU API for long-term stability before any official WebGPU integration is added to the mix.

Google Tint WebAssembly port of Google's shader cross-compiler

- Tint WASM**
- Enables reflection and cross-compilation of SPIR-V and WGSL in-browser.
 - C++ bindings for seamless native/WASM integration.

These Real-time 3D visualization of CIELUV and sRGB colour spaces

- Colours Do (Not) Exist**
- Rendered 16M particles using WebGPU; sub-16ms framerate.
 - Implemented efficient compute kernels for performing 3D colour space gamut transformations across different spaces.
 - Part of my [undergraduate thesis](#) on perceptual colourspace rendering using modern compute pipelines.

References

Available upon request.