

Piotr Rybicki

GPU Software Architect · High-Performance Computing (HPC) & Real-Time Systems

Warsaw, Poland · remote

Tech lead and software architect with a GPU performance background: two internships on NVIDIA's CUDA Performance team, then four years at Robotec.AI building and leading LiDAR simulation libraries that ray-trace hundreds of virtual sensors on one GPU, adopted by third-party simulators. I work best owning a system end to end: architecture, execution model, public API, performance, clients. Comfortable anywhere from bare metal to application code, and open to both individual-contributor and lead roles.

EMPLOYMENT HISTORY

Tenstorrent – Staff Software Engineer Sep 2025 – Apr 2026 · remote

- Implemented and validated low-level features of a next-generation AI accelerator in the pre-silicon phase, e.g. L1/L2 cache intrinsics
- Integrated AI-assisted development (Claude Code) into low-level C++ workflows

Robotec.AI – Tech Lead / Senior Software Engineer Jul 2021 – Jul 2025

Software Developer, Jul 2021 – Dec 2022 · Tech Lead / Senior Software Engineer, Jan 2023 – Jul 2025

- Technical leadership of a 3–6 person team: mentoring, code review, client communication
- Product owner of LiDAR simulation libraries based on ray tracing (CUDA, OptiX):
 - RobotecGPULidar (open source)
 - RobotecGPUSensors (proprietary)
- Owned the public API adopted by third-party integrators (AWSIM, O3DE)
- Designed the software architecture, CPU/GPU execution model and sensor models; owned performance
- Real-time simulation of 237 large LiDARs (VLS-128) or 913 medium LiDARs (VLP-32C) on a single RTX 3090

NVIDIA – CUDA Performance Team Feb 2019 – Jul 2019 · Internship

- Returned to the team after the 2018 internship
- Developed an internal tool for collecting and visualizing CUDA memcpy performance
- Achieved up to 4× memcpy speedup in certain cases

NVIDIA – CUDA Performance Team Jul 2018 – Sep 2018 · Internship

- Worked on CUDA driver internals:
 - Captured baseline performance for various GPU/CPU combinations
 - Built an experimental reimplement of CUDA context switching
 - Optimized the memory ordering of key atomic operations

NoMagic – Robotics Software Engineer Jul 2017 – Sep 2017 · Internship

- Developed and maintained Gazebo simulations
- Integrated neural networks with the perception stacks of simulated and real robots
- Coauthored “Improved GQ-CNN: Deep Learning Model for Planning Robust Grasps” (arXiv:1802.05992)

ICM UW + 7bulls – CUDA GPU Programmer Jul 2016 – Sep 2016 · Internship

ICM UW: Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw

- Implemented a parallel GPU algorithm for protein chain evolution
 - ~100× speedup over the CPU version
- Wrote the CUDA backend for Metropolis Monte Carlo folding: two-stage parallel energy reduction, SoA float4 layouts, MC steps batched per kernel launch
- Delivered into BioShell, an externally maintained, multi-contributor scientific C++ library

CONTRACT WORK / PROJECTS

Out-of-Core Streaming Point Cloud Visualizer Apr 2026 – Jun 2026

- Renders an entire 15×15 km San Francisco point cloud (180 GiB) at 120 FPS within 4 GB of VRAM on an RTX 3090
- Key optimizations (demo video ↗):
 - GPU software rasterization of points using OpenGL compute shaders
 - Screen-space density prioritization
 - Asynchronous, chunked data loading

CONTRACT WORK / PROJECTS

Nencki Institute of Experimental Biology Jun 2020 – Oct 2020 · part-time

- Gathered requirements from neuroscientists at the Laboratory of Emotions Neurobiology and built Python/Qt tooling for annotating and analyzing two-photon microscopy data

Multi-GPU Computational Fluid Dynamics Dec 2018 – Apr 2019 · B2B customer · part-time

- Implemented a multi-GPU CUDA D3Q19 lattice-Boltzmann solver
 - 5654 MLUPS at 92% of peak DRAM throughput, 21× a 32-thread Ryzen 9950X3D
 - Scales to 134M cells (512³) on a single RTX 3090
- Designed a network protocol and implemented a real-time data visualization client

EDUCATION

University of Warsaw, MIM UW – Computer Science, MSc Oct 2019 – Jul 2021

Thesis: On-demand image fetching from one of over a dozen Intel RealSense depth cameras

THESIS WORK CARRIED OUT AT

NoMagic – Robotics Software Engineer Nov 2020 – Jun 2021

- Developed and maintained a robotic pick & place system, in production at warehouses in France and Switzerland
- Reduced per-camera CPU usage from 111% to 13% (~8×) in the Intel RealSense depth stack
 - Enabled running 16 cameras on one PC at ~150 ms capture latency
 - Open-sourced as a realsense-ros fork

University of Warsaw, MIM UW – Computer Science, BSc Oct 2014 – Jul 2018

Thesis: Mars rover analogue software for the University Rover Challenge 2018 (ROS/Python)

MAJOR UNIVERSITY PROJECTS

University of Warsaw Rover Team – Software Team Lead Nov 2016 – May 2017 · Nov 2017 – Jun 2018

- Led a 3–4 person team responsible for teleoperation, machine vision and rover autonomy for the University Rover Challenge

Advanced Operating Systems, University of Warsaw Mar 2020 – Jun 2020

- Wrote an x86-64 ELF post-linker in C and patched the Linux 5.5 VFS with an O_BUFFERED_WRITE flag that defers writes until fsync
- Implemented a 1.5k-LOC Linux PCI driver for an emulated GPU with per-context page tables over DMA, mmap-able buffer fds and IRQ handling

Latte Compiler Nov 2019 – Jan 2020

- Built a Rust compiler for Latte (a Java-like language), emitting LLVM IR directly in SSA form with phi nodes instead of stack slots
- Implemented a LALRPOP parser, a type and return-flow checker, and constant folding; tested on 90 programs

SAT Solver May 2020

- Built a DPLL SAT solver in Rust with unit propagation, pure-literal elimination and constant folding, plus a recursive-descent formula parser with both naive and Tseitin CNF conversion

Embedded systems Oct 2016 – Nov 2016 · Nov 2019 – May 2020

- Built a Raspberry Pi differential-drive robot control stack in C++: quadrature encoders, H-bridge PWM, I2C ADC and PID speed control over UDP
- Programmed a bare-metal STM32F411 in C without HAL: EXTI/NVIC button interrupts, DMA-driven USART and a ring-buffered LED command protocol

SKILLS

C++17/20	C	Python	Rust	C#	CUDA	OptiX	Vulkan	OpenGL	Nsight Compute / Systems	NVTX
LLVM	Linux kernel	PCI / DMA	QEMU	STM32	I2C / GPIO / PWM	ROS 2	ROS 1	Unity	Qt	NumPy
Scikit	Multithreading	Lock-free	Profiling	CMake	Git	Docker	GoogleTest	Unit testing		

LANGUAGES Polish (native) · English (C1)