Benjamin Mastripolito

benpm@cs.utah.edu ★ *benpm.github.io*

Education

MS Computing (Graphics & Visualization Track)
University of Utah

Aug 2022 - July 2024
3.9 GPA

BS Computer Science
New Mexico Institute of Mining and Technology (NMT)

Aug 2016 - May 2020
3.79 GPA

Professional Experience

Graduate Research Assistant @ *University of Utah*
Fall 2023 - ...

Research with Dr. Cem Yuksel (*cem@cemyuksel.com*) on building a more intuitive 3D modeling methodology
- Co-wrote a 3D modeling application in C++ and OpenGL
- Implemented a compact half-edge mesh representation, reducing 36 to 16 bytes per triangle
- Implemented mesh editing tools such as extrusion, scale, translate, rotate, manipulation of geometry using harmonic weight calculation to preserve mesh details

Post-Baccalaureate Student @ *Los Alamos National Laboratory*
June 2020 - July 2022

- Research on parallel interpolation algorithms for physical equation-of-state data under mentorship of Dr. Daniel Sheppard (*danielsheppard@lanl.gov*)
- Published "SIMD-Optimized Search Over Sorted Data" in ASME Computing and Engineering, in which I developed a SIMD-optimized logarithm-based search algorithm that improved over binary search by 100% in vectorized contexts

Undergraduate Research Assistant @ *New Mexico Institute of Mining and Technology*
Jan 2019 - June 2020

- Worked with Dr. Denis Cohen (*denis.cohen@gmail.com*), and a fellow student to rewrite, parallelize, and optimize a complex 3D landslide simulation from scratch, while redesigning it to operate on unstructured meshes rather than a regular grid.

Parallel Computing Research Internship @ *Los Alamos National Laboratory*
June 2019 - Aug 2019

- Worked as a student in the Parallel Computing Summer Research Internship at LANL researching performance analysis techniques on parallel algorithms, under mentorship of Dr. Rao Garimella (*rao@lanl.gov*)

Projects

Elastic Hair Simulation and Rendering
2023

- Implemented an interactive hair simulation using discrete elastic rods, and rigid-body collision using position-based dynamics. Wrote an OpenGL compute shader able to generate 64 rendered hairs per simulated hair in < 10ms every frame.
- Implemented deep opacity shadow maps, alongside a physically-based hair shading model for realistic hair shading.

GPU-Accelerated Surface Meshing
2023-2024

Implemented a dual-mesh isosurface extraction algorithm as well as a surface geometry smoothing scheme using a trilinear interpolation as a multi-stage OpenGL compute shader pipeline. The pipeline is able to generate 0.7 million vertices and triangles in less than 10 milliseconds on modern hardware.

WebGL Compact Cellular Automata
2020

- Developed a method for storage of arbitrary, multi-state cellular automata rules which exponentially improves on a naive implementation that enumerates all permutations of the sequence of neighbor quantities.
- The method is able to store and retrieve rules into textures, with a practical maximum 14 states into less than 2^{25} bits - a single 4k byte texture. A naive implementation would require nearly 2^{44} bits. Outside the constraints of 4k texture, the method generalizes to any number of states, and any size of neighborhood, not just the standard 8-cell 2D moore neighborhood.

Rust Parallel Particle Simulation
2021

Co-developed a highly optimized parallel particle simulator in Rust, able to simultaneously simulate and render over 1,000,000 colliding particles at interactive frame rates.

CUDA Raytracer
2020

Created a parallel raytracing program using NVIDIA CUDA C++ and OpenGL. Supports BSDF materials.

Technical Skills

Confident with C++, OpenGL, and CMake. Somewhat experienced with CUDA, DirectX 12, OpenMP, Rust, WebGPU, WebGL, and Vulkan. Working knowledge of linear algebra, numerical methods, and parallel algorithm design.