World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:19, No:03, 2025

Speeding Up Lenia: A Comparative Study Between Existing Implementations and CUDA C++ with OpenGL Interop

Authors : L. Diogo, A. Legrand, J. Nguyen-Cao, J. Rogeau, S. Bornhofen

Abstract : Lenia is a system of cellular automata with continuous states, space and time, which surprises not only with the emergence of interesting life-like structures but also with its beauty. This paper reports ongoing research on a GPU implementation of Lenia using CUDA C++ and OpenGL Interoperability. We demonstrate how CUDA as a low-level GPU programming paradigm allows optimizing performance and memory usage of the Lenia algorithm. A comparative analysis through experimental runs with existing implementations shows that the CUDA implementation outperforms the others by one order of magnitude or more. Cellular automata hold significant interest due to their ability to model complex phenomena in systems with simple rules and structures. They allow exploring emergent behavior such as self-organization and adaptation, and find applications in various fields, including computer science, physics, biology, and sociology. Unlike classic cellular automata which rely on discrete cells and values, Lenia generalizes the concept of cellular automata to continuous space, time and states, thus providing additional fluidity and richness in emerging phenomena. In the current literature, there are many implementations of Lenia utilizing various programming languages and visualization libraries. However, each implementation also presents certain drawbacks, which serve as motivation for further research and development. In particular, speed is a critical factor when studying Lenia, for several reasons. Rapid simulation allows researchers to observe the emergence of patterns and behaviors in more configurations, on bigger grids and over longer periods without annoying waiting times. Thereby, they enable the exploration and discovery of new species within the Lenia ecosystem more efficiently. Moreover, faster simulations are beneficial when we include additional time-consuming algorithms such as computer vision or machine learning to evolve and optimize specific Lenia configurations. We developed a Lenia implementation for GPU using the C++ and CUDA programming languages, and CUDA/OpenGL Interoperability for immediate rendering. The goal of our experiment is to benchmark this implementation compared to the existing ones in terms of speed, memory usage, configurability and scalability. In our comparison we focus on the most important Lenia implementations, selected for their prominence, accessibility and widespread use in the scientific community. The implementations include MATLAB, JavaScript, ShaderToy GLSL, Jupyter, Rust and R. The list is not exhaustive but provides a broad view of the principal current approaches and their respective strengths and weaknesses. Our comparison primarily considers computational performance and memory efficiency, as these factors are critical for large-scale simulations, but we also investigate the ease of use and configurability. The experimental runs conducted so far demonstrate that the CUDA C++ implementation outperforms the other implementations by one order of magnitude or more. The benefits of using the GPU become apparent especially with larger grids and convolution kernels. However, our research is still ongoing. We are currently exploring the impact of several software design choices and optimization techniques, such as convolution with Fast Fourier Transforms (FFT), various GPU memory management scenarios, and the trade-off between speed and accuracy using single versus double precision floating point arithmetic. The results will give valuable insights into the practice of parallel programming of the Lenia algorithm, and all conclusions will be thoroughly presented in the conference paper. The final version of our CUDA C++ implementation will be published on github and made freely accessible to the Alife community for further development.

Keywords : artificial life, cellular automaton, GPU optimization, Lenia, comparative analysis.
Conference Title : ICPDSSE 2025 : International Conference on Parallel, Distributed Systems and Software Engineering
Conference Location : Paris, France
Conference Dates : March 29-30, 2025