

New Challenges for Cellular Automata Simulation on the GPU

John Tran

Don Jordan

David Luebke

University of Virginia

Cellular automata (CA) simulations consist of a collection of independently acting *cells* and *states* associated with those cells; cells change state in lockstep, with each cell's next state determined by its current state and the states of cells in some local neighborhood. The discrete nature of CA simulations allows them to be computed efficiently and a vast array of computations have been implemented as CA simulations, including simulations of physical processes more typically modeled with partial differential equations. The best-known CA simulation is Conway's "Game of Life," a very simple rule set that exhibits a wide range of complex behavior. An early NVIDIA demo showcased dependent texture mapping by implementing the game of life on a GPU. We have been studying the suitability and implementation of more elaborate CA simulations—in particular CA simulations using more sophisticated local neighborhood computations—on graphics hardware.

Harris et al. [2002] used a floating-point variation of CA's called a *coupled map lattice* (CML) to create physically based simulations of phenomena such as boiling, convection, cloud formation, and reaction-diffusion processes. They implemented the CML on early programmable graphics hardware and achieved speedups of about 25x over a roughly equivalent CPU implementation. However, their simulations required sampling only the four direct nearest neighbors. For many CA simulations, this is insufficient; for example, cells may query a non-fixed radius of neighbors. The resulting large and varying neighborhoods eliminate some of the simplicity exploited in Harris et al.'s GPU implementation. So while this work demonstrated that GPU's could be a useful tool for speeding up simple CA simulations, more complex CA models with less straightforward texture sampling require different strategies in their GPU implementations.

The study of *excitable media* is one domain where CA models have been found useful for approximating real life behavior. Excitable media have the property that signals can propagate through the medium without damping. One example of excitable media, which has motivated our work, is cardiac tissue simulation. CA models of excitable media are useful in studying cardiac electrophysiological behavior. Without the use of CA models, cardiac tissue simulations require solving elaborate systems of partial differential equations. Such problems would typically be sent off to supercomputer facilities, often computing for hours or days to simulate even a small piece of tissue. Therefore much interest has been shown in CA models that produce the same macroscopic behavior as the PDE models at far less computational expense. Gerhardt et al. [1990] designed a CA model that adheres to the curvature and dispersion properties found experimentally in excitable media.

The driving goal of our research is to study atrial fibrillation in real time. Atrial fibrillation (AF) is a heart disorder found in about 2 million Americans where the atrium of the heart beats excessively in uncontrolled rhythms asynchronous with the regular beat of the heart. This can cause a wide variety of symptoms, including shortness of breath, dizziness, and chest pain, as well as lead to other heart disorders. AF is often caused by dead cardiac cells which create a region of inactivity on the tissue. This can result in reentrant waves which re-excite recovering regions of tissue before expected. To meet our goal of real-time AF simulation and visualization, we hope to develop a CA model that can generate the same behavior as cardiac tissue electrical activity, and to accelerate that model on the GPU. With such a model, a user could interact with the sim-

ulation in real time, introducing new regions of conduction block. For example, such a model would enable *virtual ablation therapy* prior to operating on a patient to see if the proposed solution is even feasible. We chose the Gerhardt model as a starting point.

We have implemented the Gerhardt model in a fragment shader for the GPU and have achieved speedups of roughly 10x over an equivalent CPU implementation. The challenge here is the size of the local neighborhood: the Gerhardt model pools the excitation states of cells in a circular neighborhood within an arbitrary radius to determine the new excitation state of the cell. We chose a 3-cell radius, requiring a 7x7 neighborhood. We attribute the 10x speedup to the high memory bandwidth of the GPU, as well as to the design of the texture cache (optimized for 2D lookups). Although this result is promising, we believe that the original Gerhardt model does not simulate real life behavior of cardiac tissue accurately enough for use in virtual ablation, and have been investigating alternative CA models.

One conceptually simple improvement is to jitter the location of the cell center within each cell, potentially affecting its "visibility" to other cells. This change allows the model to generate circular focal waves—which the original model could not do—but proves difficult to efficiently implement on the GPU since the contributing neighborhood now varies for each cell. To support this on the GPU, we precompute a binary visibility map that indicates which neighbors are visible for each cell. The fragment shader implementing the CA simulation uses this map to weight contributions from each neighbor. A radius of 3 cell widths would seem to require 48 binary visibility values, but since some immediate neighbors always fall within the radius we can pack the visibility map into a single 32-bit texture of the same dimensions as the CA domain. However, lacking bitwise manipulation on current GPUs, unpacking the visibility requires expensive `fmod` operations and degrades performance to only 20% faster than the CPU implementation.

We are now working on a new CA model to replace the Gerhardt model. Since we are interested in simulating cardiac tissue, we should not require a neighborhood radius of greater than one (real cardiac cells only interact with direct neighbors via physical connections called gap junctions). Our new model also reflects the shape of real cardiac cells, which have varying lengths and are laid out in brick-like patterns. As a result, each cell connects to a variable number of neighbors. The resulting connectivity graph is a planar graph. Our current challenge is to find an efficient mapping of this model to the GPU with its regular-grid texture domains. Our preliminary results are encouraging and we hope to achieve significant speedup (10x or better) over the CPU implementation, while still maintaining the properties of cardiac wave propagation.

References

- GERHARDT, M., SCHUSTER, H., AND TYSON, J. J. 1990. A cellular automation model of excitable media including curvature and dispersion. *Science*.
- HARRIS, M. J., COOMBE, G., SCHEUERMANN, T., AND LASTRA, A. 2002. Physically-based visual simulation on graphics hardware. In *Proceedings of SIGGRAPH/Eurographics Workshop on Graphics Hardware*, 109–118.