Simulation, Modelling and Visualisation: Toolkits for Building Simulated Worlds.

D. P. Playne, A. P. Gerdelan and K. A. Hawick

daniel.playne@gmail.com, gerdelan@gmail.com and k.a.hawick@massey.ac.nz
Institute of Information and Mathematical Sciences, Massey University, Auckland

Massey University

Abstract

Simulation and Modelling continue to play important roles underpining the computational sciences [1]. In Computer Science we have a number of on-going simulation projects making use of field equation models, microscopic entity models and particle models to support research with Cahn-Hillard Cook [2] and Ginzburg-Landau equations, artificial agent models (predator-prey systems, battlefield simulations, robot tanks, robot soccer), planetary systems, molecular dynamics systems and collision-dynamics based systems. All three simulation categories have a strong need for good visualisation capabilities. While as individuals we have a range of experience and practical familiarity with the tools and technologies available, it has been difficult for individual researchers to stay abreast with the full complement of available approaches. We present here a comparative technology and design review of visualisation and rendering methods and systems that we have used as a collected resource for further simulation and modelling design.

Field Equation

Field equations approximate the behaviour of the microscopic atoms over a discrete macroscopic cell. The system is split into discrete cells and the state of each cell is defined by a set of properties. Each cell interacts with the cells in the area surrounding it and changes its state according to the properties of the surrounding cells and field equations of the simulation. Visualising these simulations involves displaying some property or combination of the properties of the cells in a graphical way. These simulations can be created for any number of dimensions, however our visualisation techniques are currently limited to three dimensions. Each dimension involves a different arrangement of cells, one-dimensional simulations are a line of cells, two dimensions a plane and in three dimensions the cells are arranged as a cube.

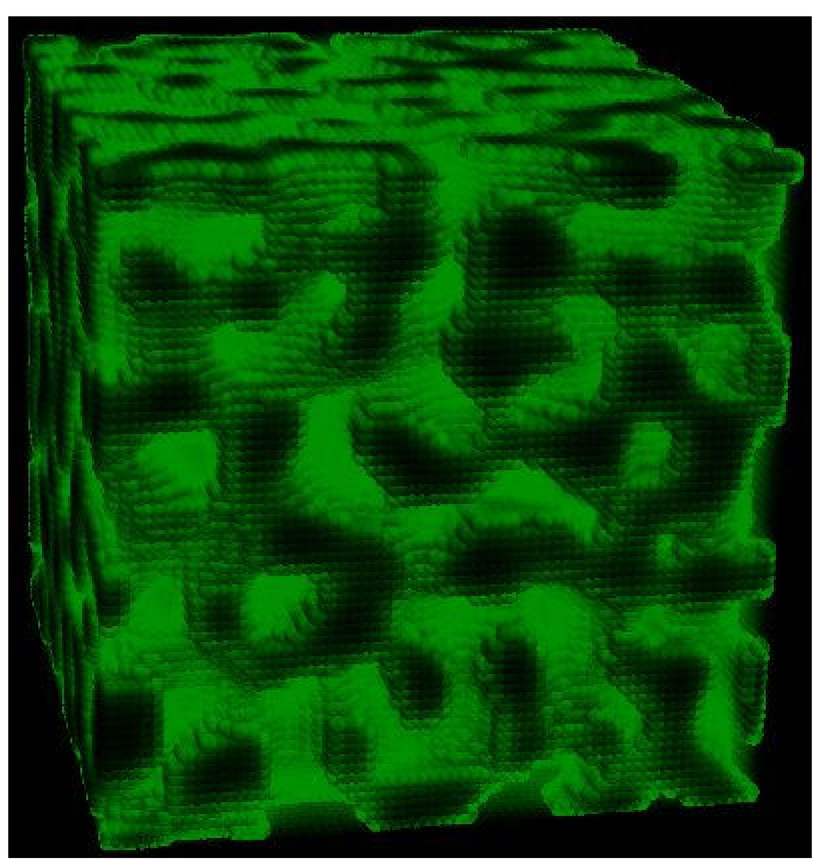

Figure 1: 3D Field Equation Visualisation.

Figure 1 shows a snapshot of the visualisation of a Cahn-Hilliard-Cook field equation simulation in three dimensions. Started from a random uniform configuration, the field cells gradually coalesce and merge driven by surface tension effects, into the emergent camouflage pattern. This sort of algorithm can be used to simulate real world textures and patterns that arise from physical processes and systems. The background image for this poster was generated from a two dimensional Cahn-Hilliard-Cook simulation.

Event-Driven

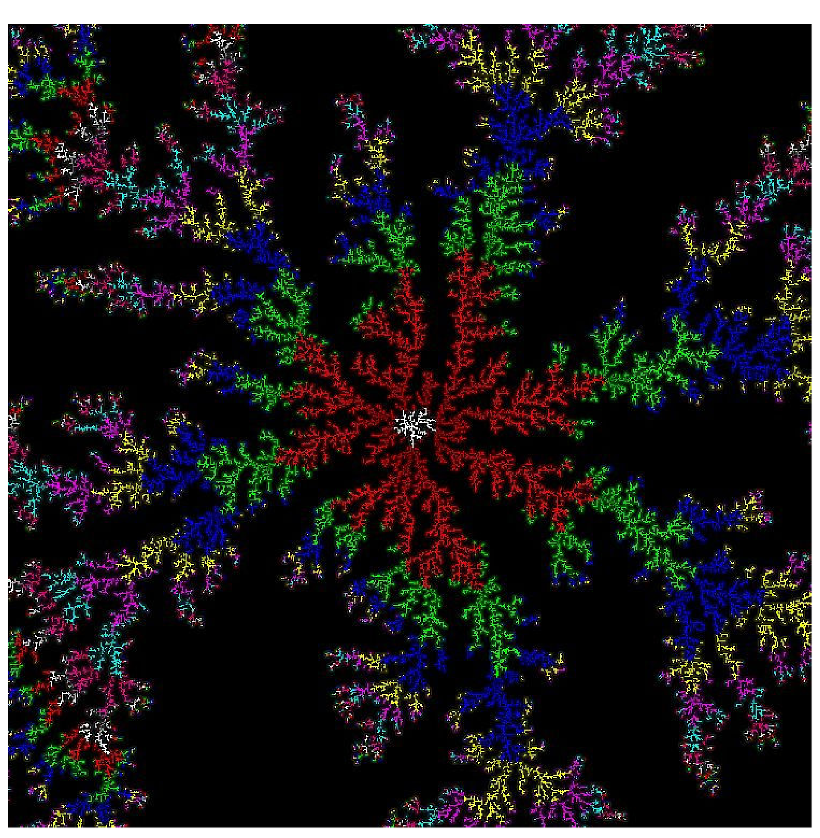

Figure 2: A diffusion-limted factal aggregate grown in 2D.

Figure 2 shows a fractal cluster grown using a diffusion-limited aggregation algorithm [3]. This is a form of event driven simulation, where a seed cell planted at the centre of the image is gradually grown by releasing "walker" particles around about it. The walkers diffuse randomly and attach themselves to the growing aggregate. This cluster was grown on a square mesh by walking and attachment events but it is also possible to employ this algorithm on continuous x-y coordinates. This aggregate has an approximate fractal dimension of d=1.6, roughly intermediate between 1 and 2 spatial dimensions. It is also possible to grow aggregates embedded in a 3-D space.

Particle Dynamics

Particle dynamics simulations are concerned with the motion/behaviour of particles or objects in space. The particle's motion is normally constrained by Newton's Laws of Motion and by some potential equation describing an attractive/repulsive force between them or some external force acting upon all particles. In each simulation, particles have a different set of properties that define their state, but in almost all particles have their position in space as part of their state. This creates the opportunity to graphically display the particles simply as an object in space according to its position in the simulator. As time progresses, the particles motion can be rendered as a movie. This can be done as real-time a display at lower quality, or as a high-quality rendered movie. These particle simulations can also be run in any number of dimensions but once again the visualisation of them is limited to the three-dimensions we have the capability to display.

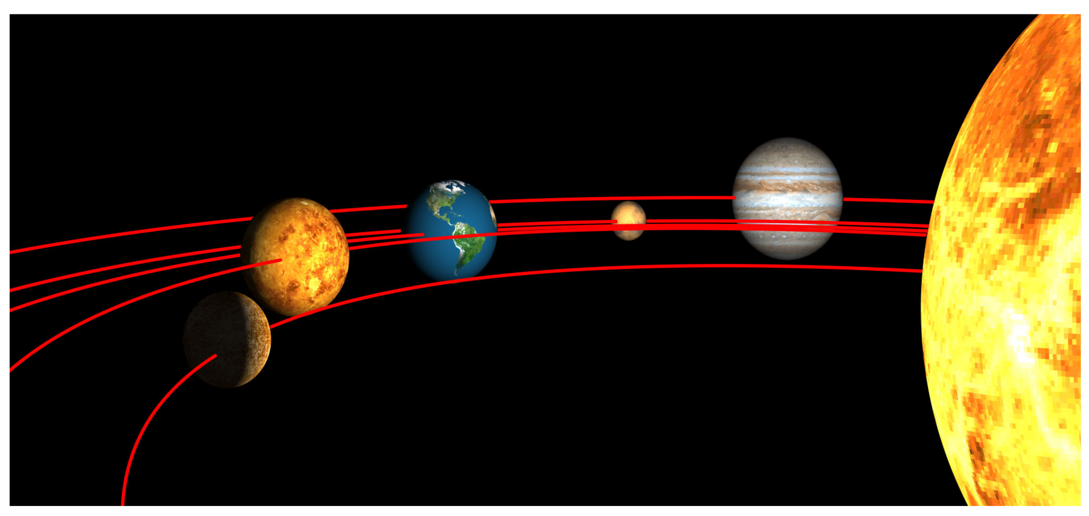

Figure 3: A ray-traced image of a Planetary Simulation.

An example of particle simulation in three dimensions is the computer simulation of planetary motion. Gravity attracts the planets together as they orbit each other, these planets can be visualised as spheres, possibly with textures applied to them to make them look more realistic. Figure 3 shows a rendered image of the visualisation of such a simulation. This specific planetary simulator is configured to simulate planets in the same configuration as the planets in our solar system. The planets have been rendered using a high-quality ray-tracing engine and have been textured using images taken of the planets in our solar system.

Combined Visualisation

Complex simulations of real-world environments require a combination of event-driven, particle, and field visualisation models in order to provide a realistic visual representation. Whilst have used particle visualisation to approximate more subtle real-world phenomena such as smoke and dust, as seen in Figure 4, terrain and landscapes are generally field models, and the majority of visualised components (projectiles, vehicles, people) are treated as entities steered by an event-driven visualisation framework. In real-world three-dimensional simulations, elements of the three visualisation types are combined, in a realism-calculation balance to produce convincing representations of the world whilst retaining an efficient rendering cycle.

Figure 4: Real-world simulations often combine different visualisation models.

Visual immersion is an important aspect of real-world simulation, and can be aided by creating accurate models of real-world entities to-scale. This is best done using three dimensional modeling software to build a mesh of the vehicle from real blue-prints. We can then use photos of the actual real-world object, taken from different angles, to accurately *texture* the model. Our attempts at this process are illustrated in Figure 5

Figure 5: A Visualisation of a Soviet T-28 Tank Created from Blue-Prints and Photographs.

Most realistic terrain in 3D simulations is now generated by fractal modelers, which provide pseudo-realistic terrain but allow very little design control to scenario designers. The terrain illustrated in Figure 6, is however based on a novel method, in that it is visually created in 3D by hand, in real time, from within the simulation itself, with a variety of WYSIWYG ("What you see is what you get") tools that we have built into the simulator. This work is based on the ETM2 ("Editable Terrain Manager") for the OGRE graphics library [4], and gives us the added advantage that we can allow simulated elements in our event driven models to deform the landscape during simulation according to the physics model.

Figure 6: User-Assisted Dynamic Landscape Generation.

Combined visualisations can be quickly constructed by assembling an effective *toolkit* of different libraries and modules. There are now a huge variety of freely available open-source tools, libraries and high-level wrappers that offer very sophisticated visualisation technology. It no longer the task of simulation builders to develop visualisation techniques for their simulations from the ground up, but rather to assemble a collection of these libraries and other resources that best suit the nature of their particular simulation project.

In summary, the simulation and visualisation techniques that we have presented here can readily be integrated together and used in both student projects and production-level programmes[5]. The Massey, Albany Computer Science courses attempt to provide a thorough background in the concepts and ideas behind these techniques. For example; **159 Teaching Papers: 102, 201, 234** provide suitable background for programming visual simulation programmes, and **Paper 235** provides an in-depth look at visualisation tools and ideas. Other papers give some insights into some of the Artificial Intelligence and computational agent algorithms. It is also possible to pursue these ideas in projects at both under-graduate (**Paper 333**) and post-graduate level (**Papers: 793, 794, 795**.)

References

[1] J. L. Casti, *Would-Be Worlds: How Simulation is Changing the Frontiers of Science.* Wiley, 1996. ISBN-0471123080.

[2] K. Hawick and D. Playne, "Modelling and Visualising the Cahn-Hilliard-Cook Equation," tech. rep., Computer Science, Massey University, 2008. CSTN-049.

[3] T.A.Witten and L.M.Sander, "Diffusion Limited Aggregation, a Kinetic critical Phenomenon," *Phys.Rev.Lett.*, vol. 47, pp. 1400–1403, Nov 1981.

[4] G. Junker, *Pro OGRE 3D Programming.* APress, 2006. ISBN 1590597109.

[5] D. Playne, A. Gerdelan, and K. Hawick, "Simulation, modelling and visualisation: Toolkits for building simulated worlds.," tech. rep., Massey University, Albany, Auckland, June 2008. CSTN-052.