David Wallace
Reel Contents, 2008

1425 Allston Way • Berkeley, CA 94702
(510) 395-2432 • david.wallace@gmail.com
http://birdhat.org/reel

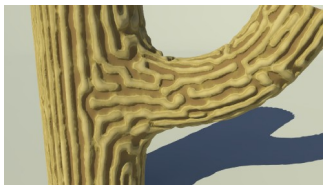
(All work was done by me unless otherwise noted.)

WALL-E Clouds

I wrote a volumetric shading/lighting engine in SLIM and used it to create these procedural clouds. The general density is controlled by an image map while the fine detail is automatically generated by a shading network I wrote, and the clouds feature procedural swirling motion. (The clouds I made became part of the set; I did not work specifically on this shot.)

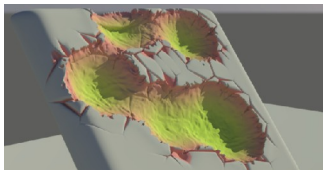
Procedural Geometry in XSI

All of these images were created within a month of the first time I used XSI. These scenes run interactively in the viewport, and the user can adjust their parameters as they run.



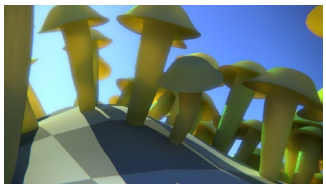
I implemented reaction-diffusion, an algorithm which generates organic patterns resembling coral or zebra stripes. My implementation works on any poly object, regardless of topology. Here I've used it to generate a cactus pattern which automatically follows the shape of the object, with no UV-mapping needed. This would be an easy way to apply bark to trees.

I created a recursive geometry generator and used it to make trees and spiral tentacles.



These craters are dynamically created on any poly object in response to particle collisions, with no UV mapping needed. The craters consist of shader-controlled procedural displacement which is linked to the cumulative particle collision data.

Procedurally-generated plants with sub-surface scattering which respond to wind forces as they grow.



Automatically-generated mushrooms which can be set to grow out of any surface. They push on each other as they grow to avoid intersecting, and they respond to gravity and the slope of the ground.

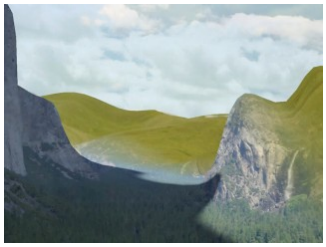
Lighting & Compositing in Shake

A shot from *Savannah, Georgia*, a short I wrote and directed. Because we had no render farm, I set up a system to compute lighting in Shake from a normal pass. This allowed me to get faster feedback as I was adjusting the lighting in the context of the final comp.
I art-directed, modeled, lit, shaded, and composited this scene. (Animation & rigging not by me.)

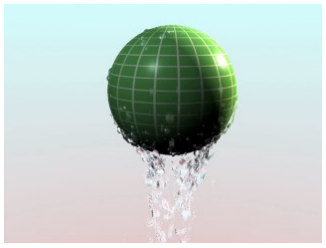
A deep zoom exercise in Shake built from a series of screenshots of Google Maps.

Maya: FX and Shading

Blowing grass using Maya Fur.



Morphing landscapes using Maya soft-bodies, MEL, and camera projection of photographs.

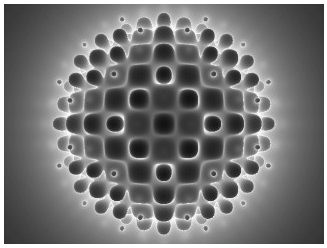


A dripping ball. The water droplets adhere to the surface until they reach the bottom of the sphere.

A banana made from scratch in 60 minutes (modeling, shading, lighting). This was a 3D icon in an interactive museum exhibit, and I made several other objects in 60 minutes each for this project.

A wet-concrete shader rendered in Mental Ray.

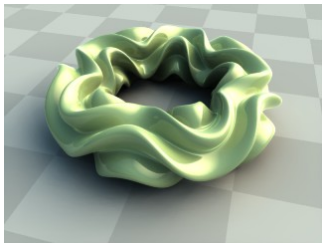
Implicit Surface Raytracer



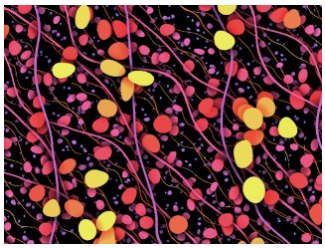
I implemented a raytracer of implicit surfaces using the Sphere Tracing algorithm (Hart, 1996) with shadows, Perlin noise, and specular highlights. Written in Python in four days while in school. The black-and-white images are histograms of render time per pixel, which conveniently double as volumetric density renders.

POV-Ray: Procedural Geometry and Animation

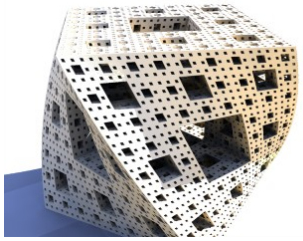
POV-Ray is a renderer with no GUI – everything is created procedurally through its built-in programming language.



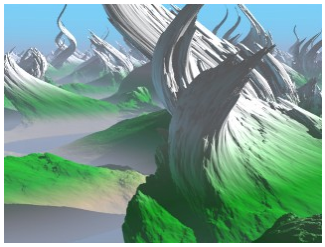
I implemented functional displacement of parametric surfaces and used it to make these displaced tori.



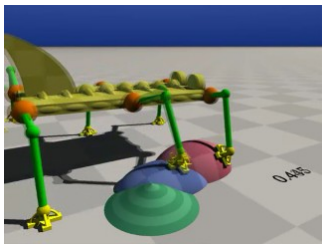
This is a single animated procedural shader.



I wrote an implicit function for this fractal, allowing it to be warped easily and rendered to an arbitrarily high level of detail with a very low memory footprint.



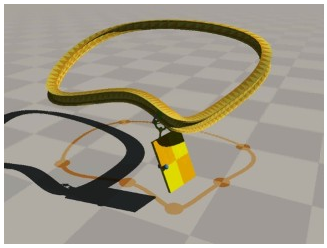
These landscapes are implicit surfaces created from summed noise functions. They extend infinitely to the horizon and use little memory.



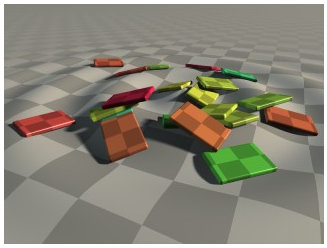
A walking robot. The user simply sets a spline path for the body to follow.

- The feet automatically place themselves on the ground, adapting to obstacles.
- The legs use inverse kinematics code I wrote.
- The body automatically wobbles with the motion of the legs.

Physical Simulation



I wrote code to simulate a door swinging from a hinge as it moves along a track. Inspired by Monsters Inc.



Results of an algorithm I developed to pile up objects in a physically plausible way without resorting to a full rigid-body simulation.

Subsurface Scattering

This uses the built-in “projection pattern” in a novel way to visually approximate subsurface scattering, which POV-Ray does not support natively.