Wrath Of The Titans - Creating CG Lava With Advected Sculpts

Daniel Seddon, Daniel LeTarte, James Kirk
Method Studios

1 Introduction

One of the many challenges on Wrath Of the Titans was to have controllable CG lava emerge from one of the heroes' arms and advance forward through a vast stone chamber, increasing in volume as it went. We first simulated the lava, building a system to emulate the subtleties of the lava's movement, and then advected surface detail along with the fluid motion.

We rely a lot on Ptex for the storage of sculpted vector displacements – this seemed ideal for the lava, where creating UV coordinates was complex and time consuming. To support Ptex, we created a surfacing algorithm based around marching cubes, but instead using octahedrons, guaranteeing that all faces would be quads (recommended for Ptex).

Figure 1: *Results of our sculpt advection method.*

2 Simulation

Simulations were designed with the necessity of outputting a persistent scalar surface field and a velocity field that was detailed enough to exhibit fluid motion. To achieve this, the lava was simulated with both level-set and particle based solvers depending on the level of control required. In addition to high viscosity and surface tension properties, areas of flow that strayed too far from artist defined paths received speed and temperature penalties, which resulted in direct modification to the simulation's collision field. Adjustments to the collision field mid-simulation yielded unique emergent behavior, as lava could melt through rock or cool and harden to form barriers along its route.

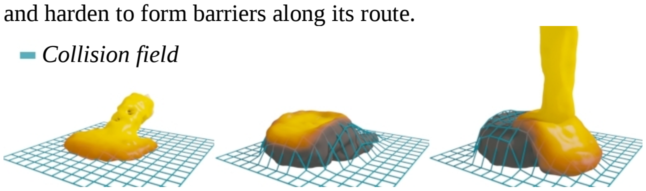

Figure 2: *Editing the collision field based on lava temperature.*

3 Surfacing – Marching Octohedrons

The final result of our fluid simulations was an SDF (signed distance field) used to build a mesh. Typically, a marching cubes algorithm is employed for this, but we implemented a similar method based around octahedrons.

Octahedrons have the advantage that all of the valances are four where faces meet. This means that cutting an octahedron always results in faces that can be built from quads, suitable for sculpting with Ptex.

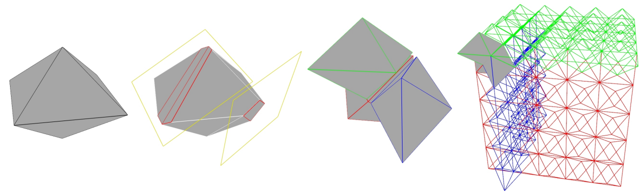

Figure 3: *Cutting an octohedron and aligning to a grid.*

4 Advecting Sculpts

Once we created a surface, our aim was to sculpt it to give the appearance of lava. We chose one or more keyframes and added surface detail in 3D Coat, a sculpting package which includes Ptex support. Using Ptex removed the need to create UV coordinates on complex procedural surfaces.

Moving the sculpt with the lava was done by advecting the mesh itself back through the simulation. As a result of the surfacing, every frame would have a unique mesh with different topology. Each of those meshes would be advected backwards through each of the previous frames. To do this efficiently, all meshes for all frames were kept in memory. The SDFs were then read in back to front, surfacing and advecting each mesh as they went. Once each mesh was advected back to the keyframe, the position of each vertex at this frame was recorded as a rest position that could be used as a spatial lookup in to the sculpt at that frame.

Each mesh would be pushed along the velocity field of the current frame then pushed down on to the isocontour of the SDF of the previous frame, to the nearest point on the fluid surface. This is important as we are advecting surface detail, not a procedural texture, so we want the advection to remain on the surface.

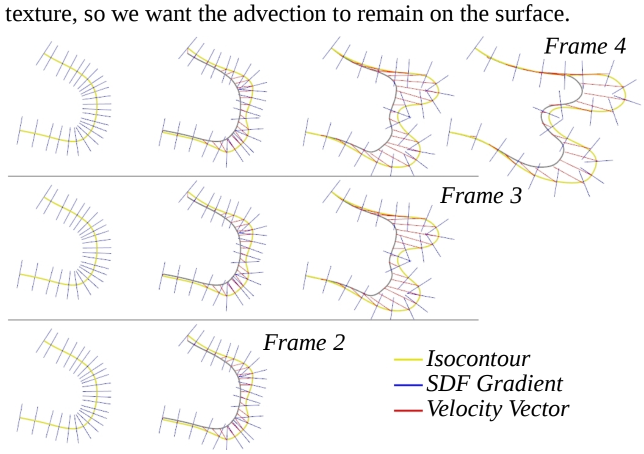

Figure 4: *Backwards advection whilst snapping to isocontour.*

Advecting back for every frame in this way means you are working out the origin of each point on the keyframe – many points can map back to a single origin, as they often do. As a result of this our approach deals with issues like branching and the creation of new surface area.

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008