Texturing Fluids

Vivek Kwatra[1] David Adalsteinsson[1] Nipun Kwatra[2] Mark Carlson[3] Ming C. Lin[1]

[1]UNC-Chapel Hill, [2]Georgia Tech, [3]DNA Productions

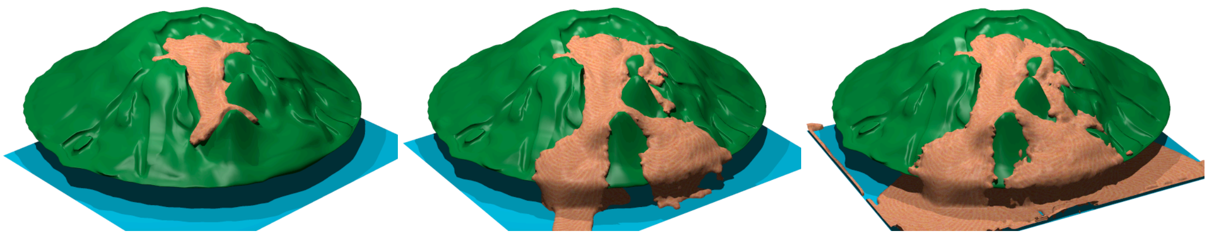

Figure 1: **Lava Scene:** Lava flowing along a mountain.

1 Motivation

Realistic modeling, simulation, and rendering of fluid media have applications in various domains, including special effects, computer animation, electronic games, engineering visualization, medical simulation, etc. Often the computational expense involved in simulating complex fluid phenomena limits the spatio-temporal resolution at which these simulations can be performed. This limitation makes it extremely difficult to synthesize complex fine-resolution microstructures on the free surface of the fluid, e.g., patterns in lava flow, small-scale waves in a river stream, leaves and flowers in the creek, etc. Even with highly robust and sophisticated simulation systems, capable of modeling such details, it is non-trivial to control the shape and appearance of these patterns within the simulation. We explore an alternative approach which makes use of samples of fluid shape and surface appearance to aid and complement fluid simulations.

2 Overview

We present a novel technique for synthesizing textures over dynamically changing fluid surfaces. We couple controllable texture synthesis with fluid simulation to perform spatio-temporally coherent fluid texturing. The main elements of our system include (i) a fluid simulator for generating the dynamic surface with velocity information, (ii) a technique for performing texture synthesis on the fluid surface coherent with neighboring surfaces, and (iii) a method for transporting texture information from one surface to the other. Fig. 2 shows a flow chart of how these three components interact with each other for fluid texturing.

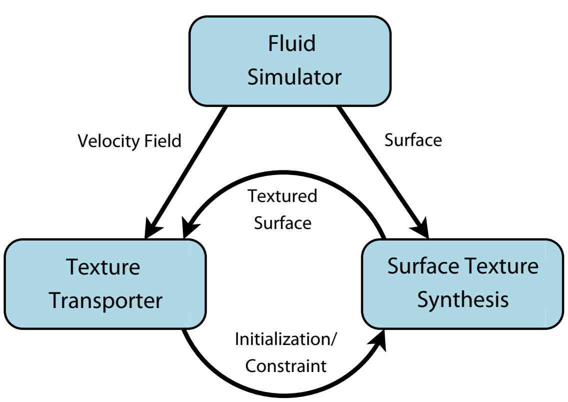

Figure 2: An overview of our fluid texture synthesis system.

We can plug-in any fluid simulator into our system that can generate the 3D velocity field of the fluid and its free surface at each iteration. We start by obtaining the free surface of the fluid for the first time step and then texture this surface using our novel "surface texture optimization" algorithm, which extends and generalizes the idea of texture optimization [Kwatra et al. 2005] to handle synthesis over 3D dynamic surfaces. We then transport the texture to the fluid free surface for the second time step using our texture transport technique. The transported quantities include texture colors (and any other associated properties like alpha mask or displacement), as well as local orientation vectors that are needed to synthesize the texture on the 3D surface.

Our technique for transporting surface orientation vectors is particularly novel as it allows for vector advection along the velocity field, which is more complex than scalar advection that is commonly used for advecting texture coordinates. For example, in a rotational velocity field, the advected vector should also rotate as it moves from one point to another. However, scalar advection of its components would (incorrectly) only translates the vector. To this end, we derive a vector advection equation that tracks changes in the orientation of the vector in addition to its location.

For texture synthesis, we place *points* on the fluid surface sampling the surface regularly, and store texture information at these points. During texture optimization, we compare unstructured point neighborhoods on the surface with grid-aligned pixel neighborhoods in the source texture by re-sampling the point neighborhood onto a regular grid. An energy function is defined that measures the squared error between each synthesized surface neighborhood and its closest match in the input texture. Optimization proceeds by minimizing this energy function w.r.t to the surface texture.

We demonstrate our algorithm using a variety of image textures on several scenes, including a broken dam, a river scene, and lava flow, as shown in Fig 1, and in the supplementary video. More information is provided in the supplementary document.

References

Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics, SIGGRAPH 2005 24*, 3 (August), 795–802.