

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

A Texture Synthesis Method for Liquid Animations

Adam W. Bargteil Funshing Sin Jonathan E. Michaels Tolga G. Goktekin James F. O'Brien

University of California, Berkeley

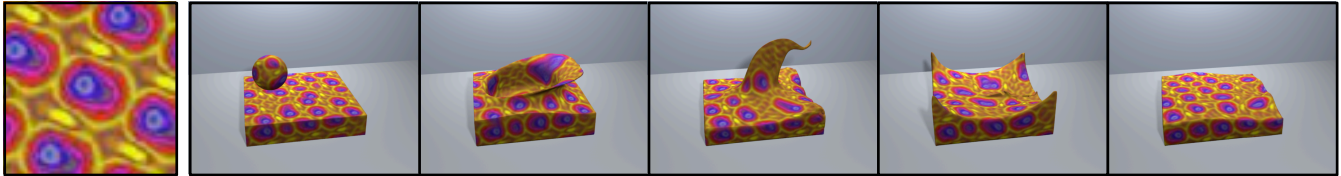


Figure 1: This simulation of splashing fluid was textured using our texture synthesis method for liquid animations. Even though the surface distorts significantly, the salient characteristics of the synthesized texture remain constant.

Overview

Liquid simulation techniques have become a standard tool in production environments, producing extremely realistic liquid motion in a variety of films, commercials, and video games. Surface texturing is an essential computer graphics tool, which gives artists additional control over their results by allowing them to stylize surfaces or add detail to a low-resolution simulations. For example, an artist could use texturing techniques to add the appearance of foam to a wave, bubbles to beer, or fat globules to soup. Unfortunately, texturing liquid surfaces is difficult because the surfaces have no inherent parameterization.

Creating a temporally consistent parameterization is extremely difficult for two primary reasons. First, liquid simulations are characterized by their complex and frequent topological changes. These topological changes result in significant discontinuities in any parameter tracked on the surface. Second, liquid surfaces tend to stretch and compress dramatically over the course of a simulation. Similarly, an advected parameterization will also stretch and compress. For these reasons, advected texture coordinates are often unsuitable for texturing liquid surfaces.

In this sketch we present a method for generating textures on animated liquid surfaces. Rather than advecting texture coordinates on the surface, we synthesize a new texture at every frame. We initialize the texture with color values advected from the surface at the previous frame. We then run an optimization procedure which attempts to match the surface texture to an input sample texture and, for temporal coherence, the advected colors.

Methods

Our method is built from three relatively new computer graphics technologies: the ability to track surface properties in liquid simulations, techniques to parameterize overlapping patches of surface, and an optimization-based technique for texture synthesis. By combining these technologies we have developed a new algorithm that generates coherent, undistorted textures on liquid surfaces based on sample textures.

Surface Tracking A necessary feature of any liquid simulation system is the ability to track the liquid's free surface. While several techniques exist, the semi-Lagrangian contouring method presented by Bargteil et al. [2006] also provides a mapping between liquid surfaces at adjacent timesteps. This mapping allows accurate and efficient tracking of arbitrary surface properties on the actual liquid surface. We make use of this feature to advect colors on the surface through time.

Surface Parameterization To apply any texture to the surface, we must construct some parameterization of the surface. For our optimization-based synthesis method, we create local parameterizations of a set of overlapping patches on the surface. For each patch, the parameterization allows us to map colors on the surface to two-dimensional texture space and vice versa. Our parameterization technique is based on that presented by Sorkine et al. [2002].

Texture Synthesis Due to discontinuities, surface stretching/compression, or blurring of the surface signal, the distorted pattern of colors generated by the semi-Lagrangian mapping typically will not closely match the original texture pattern. To force the surface colors to more closely match the input sample texture, we employ an optimization-based texture synthesis method based on the one presented by Kwatra et al. [2005]. This optimization approach is particularly appealing in our context. In many parts of the surface that have experienced minimal distortion, the advected colors may quite closely match the sample texture. Consequently, the optimization makes only minor changes. Additionally, we achieve temporal coherence by initializing the optimization with the advected colors and penalizing poor matches to the advected colors during optimization.

Conclusion

By synthesizing a new texture for every frame, our method is able to overcome the discontinuities and distortions of an advected parameterization. We avoid discontinuities in the parameterization due to topological changes by building a new parameterization of the surface for each frame. Discontinuities in advected colors are removed during the optimization procedure. Similarly, we avoid stretched and compressed parameterizations; because we optimize the surface texture for every frame, it maintains a consistent level of detail throughout the simulation. We ensure temporal coherence by initializing the optimization with the advected colors and including a coherence term in the energy function used during optimization. As a result, our method is able to produce textures with excellent temporal coherence, while still matching the input sample texture.

References

- BARGTEIL, A. W., GOKTEKIN, T. G., O'BRIEN, J. F., AND STRAIN, J. A. 2006. A semi-Lagrangian contouring method for fluid simulation. *ACM Trans. Graph.* 25, 1.
- KWATRA, V., ESSA, I., BOBICK, A., AND KWATRA, N. 2005. Texture optimization for example-based synthesis. *ACM Trans. Graph.* 24, 3, 795–802.
- SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-distortion piecewise mesh parameterization. In *the Proceedings of IEEE Visualization '02*, 355–362.