The Tar Monster: Creating a Character With Fluid Simulation

Mark Wiebe
Frantic Films
mwiebe@franticfilms.com

Ben Houston
Frantic Films
bhouston@franticfilms.com

1 Introduction

Creating the Tar Monster in Scooby Doo 2 presented a unique challenge, because the desired effect of a continually flowing textured character with expressive features had never been done before. Starting from a fluid simulator as in [Enright et al. 2002], we developed the liquid skin technique which applies a fluid layer over an animated character. In addition, the facial animation was preserved by using localized morphing, whereby a specified portion of the simulation is made to match the Tar Monster geometry. The result is a character from whom fluid constantly emanates, with texture sliding down its body and fluid splashing during vigorous arm gestures. Similar previous work includes [Sumner et al. 2003], where the "TX" character is gradually liquefied. Our method, while producing a comparable result in that texture is applied to fluid flow on a character, uses quantities defined over the volume rather than particles for control and texture coordinates.

2 The Liquid Skin Effect

The liquid skin on the animated character is applied by blending the character's velocities into the fluid velocities near the character surface. To maintain the character's integrity and replenish fluid as it falls, the character's geometry is unioned into the liquid every time step. Intuitive control over how much flow is produced is provided by a half-life H of the blending in the middle of the blend region. Let γ vary linearly between zero and one, from a specified distance inside the character to a specified distance outside the character. The ratio α for blending the character velocities over a time step Δt is then

$$\alpha = \begin{cases} 0.5^{\Delta t \frac{\gamma}{(1-\gamma)H}}, & \text{if } 0 \le \gamma < 1 \\ 0, & \text{if } \gamma = 1 \end{cases}$$

3 Texturing

The texturing done differs from [Sumner et al. 2003] in that texture coordinates are tracked by advecting a texture coordinate volume in tandem with the fluid similar to [Stam 1999], rather than using particles. Because the texture coordinate volume is used to track the texture coordinates on a free surface rather than a solid fluid volume, a naive implementation of this method can result in obvious jittering artifacts where there are slight differences between the texture volume advection and the free surface fluid advection. These problems are resolved by removing any gradient in the texture volume along the normal direction of the free surface fluid, and maintaining this throughout the simulation. This is done via the method described in [Enright et al. 2002] for extrapolating a quantity away from a surface. When liquid skin fluid is replenished, this also provides good values for the freshly unioned surface. This method of texturing provides simple and efficient access to the textures during renders. A limitation is its inability to handle texture discontinuities. We used continuous texture coordinates derived from world space and baked onto the character to avoid this problem.

4 Localized Morphing

While the liquid skin technique provides the desired flowing look, the facial animation becomes obscured behind the flow. To show

Figure 1: The gray scale morphing input, and Tar Monster

that animation, we use a localized morphing technique controlled by a gray scale texture to selectively match the fluid to the character. During the simulation, the morphing is set to converge to the actual character geometry at a given rate, in a similar fashion to Section 2. After the simulation, the morphing is done again at a higher resolution using RLE sparse level sets as described in [Houston et al. 2004], and a fixed amount of blending is used. This is done because the simulation is not done at a high enough resolution to capture the detail in the face. First, the level sets are reinitialized and the gray scale blending factor is extrapolated to fill the volume where the surface could end up. Then the following formula, provided for the post processing step, is applied to the fluid level set values ϕ when $0 < \phi_{\text{character}} < m$. δ is the value of the gray scale painting on the character, and m is distance to the effect's outer reach.

$$\alpha = \left(1 - \left(\frac{\phi_{\text{character}}}{m}\right)^2\right)(3\delta^2 - 2\delta^3)$$

α is then used to blend linearly between $\phi_{\text{character}}$ and ϕ_{fluid}. One caveat of this blending approach is that if the area near the face is filled with fluid beyond the maximum distance from the surface, a blob will be left hanging in mid air. Because the morphing operates both during simulation and later in rendering, this problem doesn't occur in practice for our application. To match the textures in the localized morph region, we use a blend ratio similar to the one used by Section 2, but thinner and made to vary from completely inside the character to outside the surface using the gray scale control parameter.

References

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. In *Proceedings of SIGGRAPH 2002*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 736–744.

HOUSTON, B., WIEBE, M., AND BATTY, C. 2004. Rle sparse level sets. In *SIGGRAPH 2004 Sketches & Applications*, ACM Press.

STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 121–128.

SUMNER, N., HOON, S., GEIGER, W., MARINO, S., RASMUSSEN, N., AND FEDKIW, R. 2003. Melting a terminatrix. In *SIGGRAPH 2003 Sketches & Applications*, ACM Press.