

Summary: Screen space fluid rendering with curvature flow

Bram Musters, Maarten Terpstra

March 11, 2015

The paper describes an approach to render the surface of particle-based fluids. The paper divides the technique in different steps, in the following sections, each step is explained in detail. Different methods that try to accomplish the same goal are also mentioned.

Surface depth determination

The first step of the technique is to render the particles as spheres using point sprites, to a frame-buffer object. Then the depth of the particle that is closest to the camera is determined per fragment using a depth-test.

Surface depth smoothing

The depth values are smoothed in the z direction using curvature flow. Curvature flow reduces the mean curvature of a surface, in order to obtain a smoother surface. This results in a more natural simulation, because in a natural fluid there are no sharp discontinuities between particles. The mean curvature determines how much a point is translated along the z direction.

Thickness

Thickness is the measure that attenuates colors depending on how deep an opaque object is submerged in water. If an opaque object is deeper in the water its colors will look less lively and will be less visible. To simulate this feature, the amount of water particles between the camera and an opaque object is determined and the colors are attenuated based on this number. This number is written to a second render target.

Noise texture

Natural surfaces are not completely flat due to other factors such as wind and current. To add this surface detail and thus obtain more natural surfaces, Perlin noise is added to each particle. This way, the noise is advected throughout the simulation, adding to the realism.

Rendering

In the final step, all the previous steps are combined into a final scene by rendering a full-screen quad. The colors of the particles are determined using Fresnel equations and Phong shading. Fresnel equations determine the reflection and refraction component of fluids, these are used to compute the specular highlight using Phong shading.