Surface reconstruction for particle-based methods

1 Introduction

In Computer Graphics, particle-based methods are used to simulate fluids. Particles are interpolation points and not real fluid elements. Thus, there is no straightforward way to extract a smooth polygonal description of the actual liquid surface. Different methods have been proposed to solve this problem. At the moment, we use a parametrization of the isosurface proposed in [ZB05]. For the interpolation, we apply the marching cubes method [LC87]. However, the resulting mesh suffers from temporal aliasing as triangles pop in or out.

The goal of this master-thesis project is to find an alternative method for extracting a smooth polygonal description which improves the visual quality. It would be very nice, if you are able to discuss different approaches in the end. I would suggest the following roadmap:

- First, make yourself familiar with the current method [ZB05]. Since the mesh quality also depends on the number of underlying particles, it is recommended to use our Particle Render software. However, if you want to implement [ZB05], you are free to use your own software.
- Get familiar with the other approaches mentioned in the following and implement (some of) them. Thus, this means that you should extend our Particle Renderer software by other surface reconstruction methods.
- At this point, you should have the basics to discuss advantages and disadvantages of the surface reconstruction approaches for particle methods. We will then see where the problems are and think about possible solutions.
- Possible solution: Exploit [KBSS01] for particle-based fluid methods. By extracting the sharp feature regions and add additional feature points, in order to get rid of aliasing effects.
- Write your thesis. Note, that the quality of your written report is very important. The more results you can discuss, the easier ;)

In the next section, I will describe the basics of promising existing methods.

2 Marching cubes

The marching cubes method is a popular method to construct an isosurface from a data volume. Isosurfaces represent points of a constant value within a volume of space. The input for the marching cubes method is a scalar field and the output is a triangulated surface mesh. For particle-based methods there are various strategies to compute the scalar field, in order to extract the fluids surface. They are described in the following.

2.1 Zhu and Bridson

In [ZB05] each scalar value represents the signed distance of a point $\mathbf{x}$ in space to the fluid surface. For a single particle $\mathbf{x}_0$ with radius r_p, the signed distance field $\Phi(\mathbf{x})$ can be written as:

$$\Phi(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_0\| - r_p. \tag{1}$$

In general (for more than one particle), an approximation of this formula is used. Thereby, $\mathbf{x}_0$ is replaced by the weighted average of neighboring particle positions $\bar{\mathbf{x}}$. Note that in our framework, the particle radius r_p is the same for all particles.

$$\Phi(\mathbf{x}) = \|\mathbf{x} - \bar{\mathbf{x}}\| - r_p \tag{2}$$

$$\bar{\mathbf{x}} = \sum_i \alpha_i \mathbf{x}_i \tag{3}$$

$$\alpha_i = \frac{W_{mc}(\mathbf{x} - \mathbf{x}_i, r_{mc})}{\sum_j W_{mc}(\mathbf{x} - \mathbf{x}_j, r_{mc})} \tag{4}$$

where W_{mc} is a kernel function proposed by Zhu and Bridson

$$W_{mc}(\mathbf{x}, r) = \max\left(0, \left(1 - \left(\frac{\|\mathbf{x}\|}{r}\right)^2\right)^3\right). \tag{5}$$

with $r = r_{mc}$ is the radius considered around $\mathbf{x}$. Typically it is chosen as $r_{mc} = 2r_p$.

Disadvantages: Artifacts in concave regions, since $\bar{\mathbf{x}}$ may erroneously end up outside the surface in concavities. They can be removed by sampling $\Phi(x)$ on a higher resolution grid and then doing a simple smoothing pass. (?) grid smoothing (?) You will find the computation in `MCIllustrator::calcNodalValuesCSPH(...)`

2.2 Solenthaler

In [SSP07] a modification of [ZB05] is presented to remove artifacts in concave regions and between isolated particles and splashes.

The Jacobi matrix of $f(\mathbf{r})$ with $\mathbf{r} = (r_x, r_y, r_z)^T$ is computated as follows:

$$\nabla f(\mathbf{r}) = \begin{pmatrix} \frac{\partial f_1}{\partial \mathbf{r}_x} & \frac{\partial f_1}{\partial \mathbf{r}_y} & \frac{\partial f_1}{\partial \mathbf{r}_z} \\ \frac{\partial f_2}{\partial \mathbf{r}_x} & \frac{\partial f_2}{\partial \mathbf{r}_y} & \frac{\partial f_2}{\partial \mathbf{r}_z} \\ \frac{\partial f_3}{\partial \mathbf{r}_x} & \frac{\partial f_3}{\partial \mathbf{r}_y} & \frac{\partial f_3}{\partial \mathbf{r}_z} \end{pmatrix} \tag{6}$$

where

$$f(\mathbf{r}) = \sum_j \mathbf{r} W(|\mathbf{r} - \mathbf{r}_j|) \tag{7}$$

and

$$\begin{aligned} f_1(\mathbf{r}) &= \sum_j r_x W(|\mathbf{r} - \mathbf{r}_j|) & (8) \\ f_2(\mathbf{r}) &= \sum_j r_y W(|\mathbf{r} - \mathbf{r}_j|) & (9) \\ f_3(\mathbf{r}) &= \sum_j r_z W(|\mathbf{r} - \mathbf{r}_j|) & (10) \end{aligned}$$

W is definded as

$$W(\mathbf{x}) = \begin{cases} 0 & |\mathbf{x}| \geq r_{mc} \\ \left(1 - \left(\frac{|\mathbf{x}|}{r_{mc}}\right)^2\right)^3 & |\mathbf{x}| < r_{mc} \end{cases} \tag{11}$$

where $\mathbf{r} - \mathbf{r}_j = \mathbf{x}$.

Now, the partial derivative of f_1 with respect to r_x is computed as

$$\begin{aligned} \frac{\partial f_1(\mathbf{r})}{\partial r_x} &= \frac{\partial r_x}{\partial r_x} W(\mathbf{x}) + r_x \frac{\partial W(\mathbf{x})}{\partial r_x} & (12) \\ &= W(\mathbf{x}) + r_x \frac{\partial W(\mathbf{x})}{\partial r_x} & (13) \end{aligned}$$

The partial derivative of $W(\mathbf{x})$ with respect to r_x is computed as

$$\frac{\partial W(\mathbf{x})}{\partial r_x} = \begin{cases} 0 & |\mathbf{x}| \geq r_{mc} \\ 3\left(1 - \left(\frac{|\mathbf{x}|}{r_{mc}}\right)^2\right)^2 \cdot -2\left(\frac{|\mathbf{x}|}{r_{mc}}\right) \cdot \left(-\frac{x}{|\mathbf{x}| r_{mc}}\right) & |\mathbf{x}| < r_{mc} \end{cases} \tag{14}$$

Note that $x = r_x - r_{j_x}$. So in total we get

$$\frac{\partial f_1(\mathbf{r})}{\partial r_x} = \begin{cases} 0 & |\mathbf{x}| \geq r_{mc} \\ W(\mathbf{x}) + r_x 3\left(1 - \left(\frac{|\mathbf{x}|}{r_{mc}}\right)^2\right)^2 \cdot \frac{2x}{r_{mc}^2} & |\mathbf{x}| < r_{mc} \end{cases} \tag{15}$$

2.3 Adams

[APKG07] use approximate particle-to-surface distances d_i to define a smooth fluid surface. The d_i are carried along with the particles and updated using an efficient redistancing algorithm. The level set function is defined as:

$$\Phi(\mathbf{x}) = d(\mathbf{x}) - \|\mathbf{x} - \bar{\mathbf{x}}\| \quad (16)$$

$$d(\mathbf{x}) = \sum_i \alpha_i d_i \quad (17)$$

$$\alpha_i = \frac{W_{mc}(\mathbf{x} - \mathbf{x}_i, r_p)}{\sum_j W_{mc}(\mathbf{x} - \mathbf{x}_j, r_p)} \quad (18)$$

Note that they use the same smoothing kernel as [ZB05]. However, they choose $r_{mc} = r_p$. The authors report that their reconstruction method improves [ZB05].

To implement this method, we have to compute the d_i. But how?

The redistancing can be implemented efficiently by projecting particles near the surface onto the surface and propagating the distance information to the other particles in the interior of the volume. This is achieved using a simple binary search along the ray segment from $\mathbf{x}_i$ to $\mathbf{x}_i + r_p \nabla\Phi(\mathbf{x}_i)$. Note that $\Phi(\mathbf{x}_i)$ is a vector that points outside of the volume. Explain how to compute $\Phi(\mathbf{x}_i)$.

Thus, they evaluate $\Phi(\mathbf{x}_i + s\nabla\Phi(\mathbf{x}_i))$ for $s = 0, s = r_p/2$ and $s = r_p$. The binary search recurses into the interval where the sign changes. They found that one recursion step and linear interpolation between the last two obtained points is adequate to compute a projection point $\mathbf{y}_i$ for each particle p_i. The projection is performed for all particles p_i with a distance-to-surface d_i smaller than their support radius r_p.

Howto initialize the d_i ? How does the gradient $\nabla\Phi$ look like?

Compare this method with [ZB05] according to quality and performance.

2.4 Foster and Fedkiw

In [FF01], a smoothing function is used to reduce unnatural folds or corners in the surface. The smoothing equation writes

$$\Phi_\eta = -S\left(\Phi_{\eta-1}\right)\left(|\nabla\Phi| - 1\right) \quad (19)$$

where $\Phi_0 = \Phi$ and $S(\Phi_\eta)$ can be computed as

$$S(\Phi) = \frac{\Phi}{\sqrt{\Phi^2 + \Delta\tau^2}}. \quad (20)$$

$\Delta\tau^2$ corresponds to the grid spacing used in the grid based method of [FF01]. However, we can try to use the spacing size of the marching cubes grid.

3 Extending the marching cubes method

3.1 Extract single particles

The surface mesh can consume a lot of memory. Thus, the computation, file operations and the rendering can take too much time. Furthermore, the visual results are poor in regions were the particle number is low, i.e. drops. We should try to differentiate between the main fluid volume and single drops. These particles can be rendered as single points, spheres for example. They should not contribute to the surface mesh. The material properties should be chosen in such a way that the single particles are nearly not visible, but look like spray, if in a group.

3.2 Sharp feature extraction and smoothing

Read [KBSS01].

4 Marching Tiles

Maybe it could be rewarding to look into [Wil08].

References

[APKG07] Bart Adams, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. Adaptively sampled particle fluids. *ACM Trans. Graph.*, 26(3):48, 2007.

[FF01] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM.

[KBSS01] Leif P. Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 57–66, New York, NY, USA, 2001. ACM.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[SSP07] Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. A unified particle model for fluid–solid interactions: Research articles. *Comput. Animat. Virtual Worlds*, 18(1):69–82, 2007.

[Wil08] Brent Warren Williams. Fluid surface reconstruction from particles. Master's thesis, University of British Columbia, 2008.

[ZB05] Y. Zhu and R. Bridson. Animating sand as a fluid. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 965–972, New York, NY, USA, 2005. ACM Press.