Construction of Non-blobby Surface from Particles

Takahiro Harada, Seiichi Koshizuka and Yoichiro Kawaguchi

The University of Tokyo

Abstract

This paper presents a method that can construct a surface with a thin region or sharp edges from particles. The surface is constructed in two stages. An implicit surface is generated by assigning a density distribution to each particle in the first step and then the surface constructed from the density distribution is changed to a surface with a thin region or sharp edges. Furthermore, the method can generate various kinds of surfaces because a surface is controlled by several parameters. Therefore, the present technique increases the range of expression of particle-based simulations. In this paper, some calculation results are presented and finally an application from the results of particle-based fluid simulation is discussed.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations

1. Introduction

Particle methods are not accompanied by numerical dissipation through advection calculations, and therefore a fluid does not lose its mass. In addition, it is easy to simulate a free surface flow because particles themselves represent a fluid and we do not have to track a surface in addition to the fluid simulation. Because a fluid is a continuum and particles are its discretized representation, a fluid surface has to be constructed to render simulated particles as a fluid. However, it is difficult to construct a good surface from simulated particles. Generally, a surface is generated by constructing an implicit surface with blobs assigned to each particle. This method can construct a surface but it is difficult to represent a thin region or sharp edges of a fluid which can be seen in the scene as illustrated in Figure 1.

This paper presents a method that can construct a surface with a thin region or sharp edges from particles. The method first generates an implicit surface by assigning a density distribution to the particles and then constructs a surface by processing the surface of the implicit surface. Existing methods that construct a surface using blobs can only control the configuration of the surface by changing the function of the density distribution. On the other hand, the proposed method can control the configuration by adjusting a few parameters of the surface in addition to changing the density distribution function. Therefore, the proposed method can increase the degrees of freedom of the configuration of the surface constructed from particles so that it can generate various surfaces from the same particle distribution. After describing the method, we show two- and three-dimensional examples with a few particles. Finally, by applying the present method to the results from Smoothed Particle Hydrodynamics (SPH) [Mon92], which is one of the particle methods, we demonstrate the capability of the method.

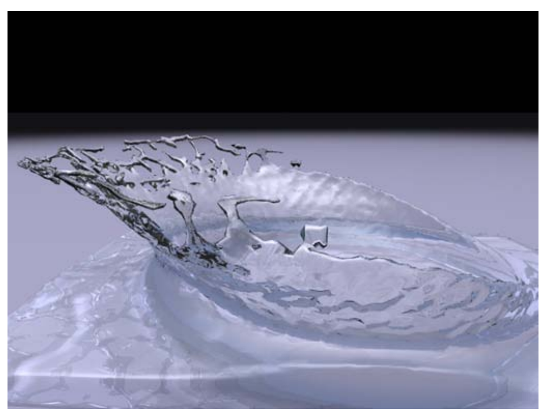

Figure 1: *A ball of fluid is thrown into a pool.*

2. Related Work

The method developed by Blinn [Bli82] is often used to construct a surface from particles and has been applied to a particle-based simulations to construct surfaces from the results. Müller *et al.* studied a real-time fluid simulation by SPH and used the method for surface construction [MCG03]. Premoze *et al.* introduced Moving Particle Semi-Implicit (MPS) method [KO96], a particle method to solve incompressible flow, to the computer graphics community [PTB*03]. However, they used the level set function for surface construction because generation of a smooth surface from particles is difficult. Zhu and Bridson developed a method that can construct a smoother surface [ZB05]. Although this method reduces the bumps on the surface, it is still difficult to generate sharp edges.

3. Method

The proposed method generates a fluid surface in two stages. A fluid surface is constructed by blobs in the first step and in the second, the surface obtained is processed.

First, blobs with a weight function w are assigned to each particle and the density $\phi(\mathbf{x})$ at an arbitrary position $\mathbf{x}$ is calculated by summing the weight function of all of the particles as follows.

$$\phi(\mathbf{x}) = \sum_j w(\mathbf{x} - \mathbf{r}_j) \tag{1}$$

where $\mathbf{r}_j$ is the position of particle j and the weight function w is defined as follows.

$$w(\mathbf{x}) = \frac{r_e}{|\mathbf{x}|} - 1 \tag{2}$$

where r_e is the effective radius.

The surface of a fluid is defined with a threshold value r_0. In three dimensions, the computational region is divided into grids and the density ϕ is calculated at the grid points. Then, a surface polygon is generated at the position where $\phi(\mathbf{x}) - r_0 = 0$. Let the number of vertices belonging to the surface be n and let their positions be $V^0 = \{\mathbf{v}_0^0, \mathbf{v}_1^0, \mathbf{v}_2^0, \cdots, \mathbf{v}_n^0\}$.

In the second step, vertex positions $T^0 = \{\mathbf{t}_0^0, \mathbf{t}_1^0, \mathbf{t}_2^0, \cdots, \mathbf{t}_n^0\}$ are computed for each vertex $\mathbf{v}_i^t$ by calculating the weighted sum of the positions of neighboring particles and the sum of the weight function w'.

$$\mathbf{t}_i^t = \frac{\sum_j w'(\mathbf{v}_i^t - \mathbf{r}_j)\mathbf{r}_j}{\sum_j w'(\mathbf{v}_i^t - \mathbf{r}_j)} \tag{3}$$

In this study, the weight function w' is set to the same function to w described above.

Vertices T^t are calculated for each vertex and then the vertex position $\mathbf{v}_i^t$ is displaced to $\mathbf{v}_i^{t+1}$ by using $\mathbf{t}_i^t$ and a parameter c ranging from 0 to 1.

$$\mathbf{v}_i^{t+1} = (1-c)\mathbf{v}_i^t + c\mathbf{t}_i^t \tag{4}$$

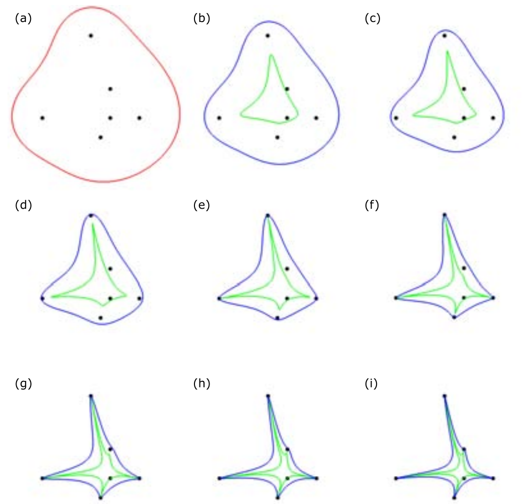

Figure 2: *An example in two dimensions. The black points indicates particle positions. Figure (a) shows the surface of the implicit surface constructed using blobs. The blue and green curves in other figures shows the surfaces* $\mathbf{v}_i^t$ *and* $\mathbf{t}_i^t$ *at* $t = 1,2,3,4,5,6,7,8$*, respectively. The parameter* c *was* 0.3 *in all of the results.*

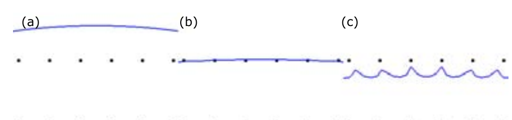

Figure 4: *An effect on an uniform particle distribution by changing parameter* t*. In these results,* $c = 0.3$ *are used for the surfaces at* $t = 3,5,7$*.*

The surface configuration can be adjusted by changing the parameter c. The normal vectors of vertices are required to shade a surface. If the surface is constructed using blobs, the normal vectors are the gradient of the density. However, the normal vectors of V^t cannot be calculated in the same way. Therefore, they are calculated by adding the normal vectors of polygons sharing the vertex and the normalized vector is set to the normal vector of the vertex.

By repeating the second step with Equations (3) and (4) and replacing $\mathbf{v}_i^t$ with $\mathbf{v}_i^{t+1}$, we can obtain a vertex $\mathbf{v}_i^{t+2}$. In this way, the configuration of the surface is adjusted by changing the values of c and t.

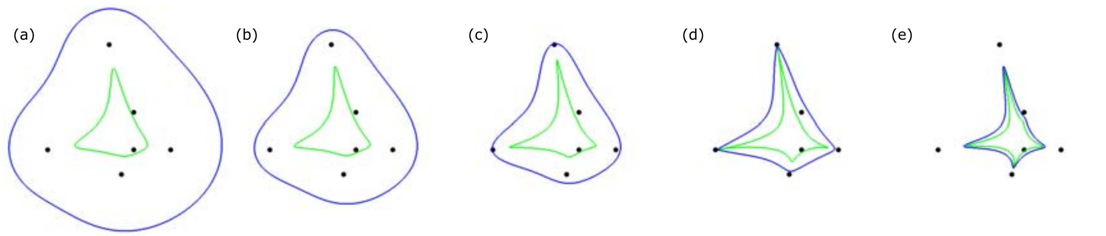

Figure 3: *Variation of the surface by changing parameter c. The blue and green curves shows* $\mathbf{v}_i^t$ *and* $\mathbf{t}_i^t$*, respectively. In these results,* $c = 0, 0.2, 0.4, 0.6, 0.8$ *are used for the surfaces at* $t = 2$*.*

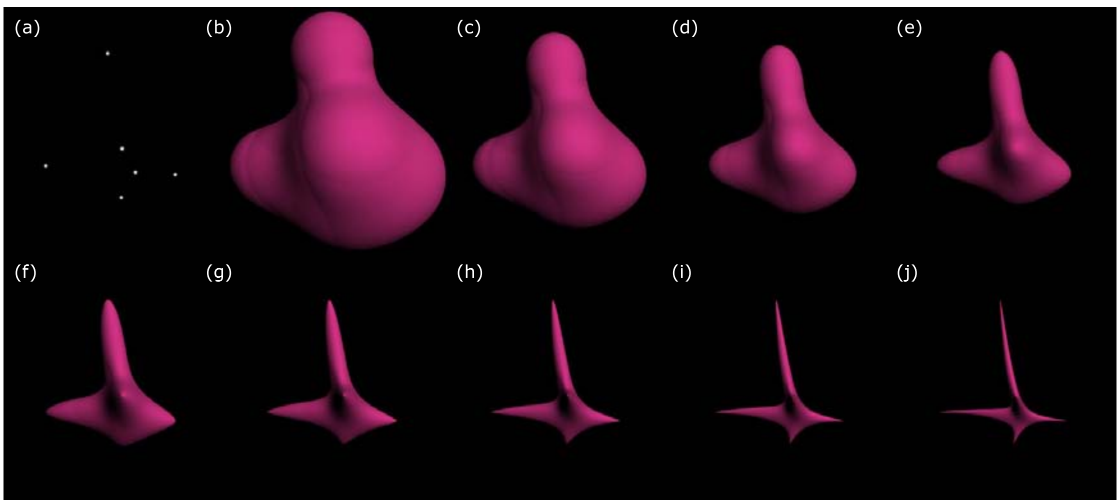

Figure 5: *An example in three dimensions. Six particles are used and their positions are shown in figure (a). The other figures shows the surface at* $t = 0, 1, 2, 3, 4, 5, 6, 7, 8$*, respectively. The parameter c was* 0.3 *in all of the results.*

4. Results and Discussion

Calculation results with 6 particles in two dimensions are shown in Figure 2. The left in the upper row shows an implicit surface calculated in the first step. The green and blue curves in the other images are T^t and V^t, respectively. Surfaces at $t = 1$ are shown in (b) and surfaces at $t = 2$ are shown in (c). We can see that as t increases, the surface V^t approaches the particle positions and a sharper surface can be constructed. Figure 3 shows the variations of the surface by changing the parameter c in Equation 4. The surfaces are constructed with $t = 2$ -the surface after two iterations. The figure on the left shows a surface with $c = 0$. This surface is the same surface obtained in the first step. As c increases, V^t gets closer to T^t. These figures shows that it is possible to construct several variations of surfaces by changing the parameters t and c. When only blobs are used, we can only adjust the surface by changing the weight function. However, with the proposed method, we can construct several surfaces by adjusting these two parameters. This method also affect flat parts of a surface. Variation of a flat surface is illustrated in Figure 4. The figure shows that we can obtain a flat surface when an appropriate set of parameters are choosen but there is also a possibility that the present method makes a surface spiky.

Figure 5 shows an example in three dimensions. After the first step, surface polygons are constructed by using Marching Cubes [LC87]. In this example, 6 particles are used and their distribution is shown in (a). The other figures shows how the surface changes as t changes. We can see that the present method can generate various surfaces that are difficult to produce using only blobs.

Then we show an application of the particle-based simulation. A fluid motion was simulated by SPH [HKK07] and surfaces were constructed from simulated particles. To search for neighboring particles of vertices efficiently, a three-dimensional grid covering the computational domain was introduced. Before constructing a surface from particles, the index of a particle is stored in the voxel to which the particle belongs. The side length of the voxel is set equal to the effective radius of the weight functions. Figure 1 is a frame from a simulation in which a ball of fluid was thrown into a water tank. This is one of the scenes where it was difficult to construct a plausible surface by blobs because thin regions cannot be expressed. Another scene rendered with a matte shader is shown in Figure 6. A circular inflow and a ball were used. The left column shows surfaces made with blobs and the right column shows surfaces to which the present method was applied. A fluid was simulated with 14,272 particles and parameters were set to $c = 0.3, t = 2$. As we can see from these figures, the blobs cannot express sharp edges and thin water. However, the proposed method can express not only sharp edges but also thin water. The surface consists of 39,564 vertices and 79,100 polygons and the computation time for this example is about 15 seconds on a PC with a Core2 X6800 CPU. When $t = 1$, the computational time is about half of the time. The computational time of the present method is proportion to the parameter t.

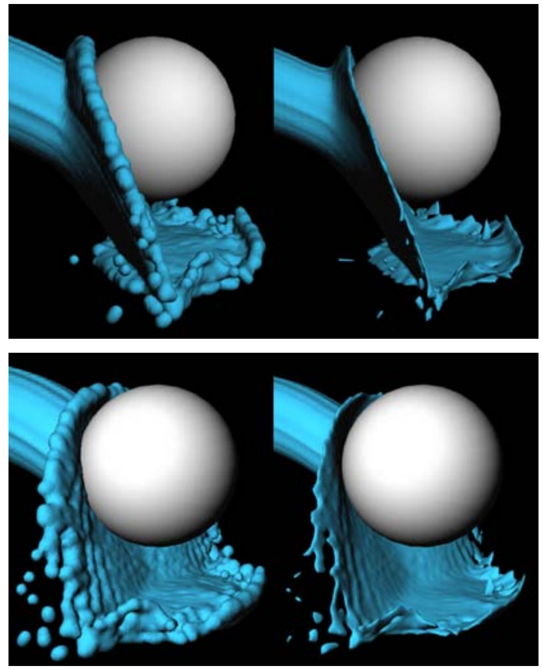

Figure 6: *An application to three-dimensional fluid simulation. The left column shows results using blobs and the right column shows results using the proposed method.*

5. Conclusion

We have presented a method that can construct a surface with thin regions or sharp edges from particles. The method can generate a surface which was difficult to produce with blobs. In addition, by adjusting both the weight functions and parameters, the method can be used to construct various surfaces. We showed how the parameters c and t affect the surface configurations in two dimensions. We also applied the proposed method to particles whose positions were calculated by a fluid simulation and showed that it can reduce the bumps and construct a sharp surface.

References

[Bli82] BLINN J.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3 (1982), 235–256.

[HKK07] HARADA T., KOSHIZUKA S., KAWAGUCHI Y.: Smoothed particle hydrodynamics on gpus. In *Proc. of Computer Graphics International* (2007), pp. 63–70.

[KO96] KOSHIZUKA S., OKA Y.: Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl.Sci.Eng. 123* (1996), 421–434.

[LC87] LORENSEN W., CLINE H.: Marthing cubes: A high resolution 3d surface construction algorithm. In *Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (1987), pp. 163–169.

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proc. of SIGGRAPH Symposium on Computer Animation* (2003), pp. 154–159.

[Mon92] MONAGHAN J.: Smoothed particle hydrodynamics. *Annu.Rev.Astrophys. 30* (1992), 543–574.

[PTB*03] PREMOZE S., TASDIZEN T., BIGLER J., LEFOHN A., WHITAKER R.: Particle-based simulation of fluids. *Computer Graphics Forum 22*, 3 (2003), 401–410.

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Transactions on Graphics 24*, 3 (2005), 965–972.