Creating Endless Water Flow Animation using Particle Data

Masanori Sotozaki[1], Yoshinori Dobashi[1,2] and Tsuyoshi Yamamoto[1]
[1]*Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido, Japan*
[2]*JST CREST, Tokyo, Japan*

Keywords: Particle-based Simulation, Endless Animation, Water, Natural phenomena, River.

Abstract: In this paper, we propose an efficient method for synthesizing water animation such as waterfall or rivers using particle-based simulation. Recently, physically based simulation has become a popular technique to create realistic animation of natural phenomena in many applications, e.g. commercial films, movies and games. Particularly, there is a growing demand on synthesizing realistic animation of fluids, such as water. However, realistic fluid animation requires a high computational cost. Some applications requiring real time performances, such as games, cannot afford such a high computational cost. In this paper, we propose an efficient method for creating endless animations of water flow using particle data generated by fluid simulation. We store a set of dynamic particles in a database and use them repeatedly to produce endless animations.

1 INTRODUCTION

Computer graphics has become a popular technique and used in many applications, such as movies, games, and commercial films. In particular, physically-based simulation is often used recently because it can create realistic animations of natural phenomena by simulating actual physical processes. Many physically-based methods have been proposed for simulating fluids such as water, smoke, fire, and so on. There are two main approaches for physically-based fluid simulation: grid- and particle-based methods. Grid-based methods uses grids to store physical quantities, such as velocities, at each grid point (Stam, 1999), (Foster and Fedkiw, 2001). Particle-based methods represent the fluid with particles, calculating the velocity and position of each particle (Müller et al., 2003). Particle-based methods are popular for simulating water motion since it can handle drastic deformations of water surface such as splash. However, physically-based methods have some problems. One of the problems with physically-based simulations is its expensive computational cost.

In entertainment applications, it is often required to create endless water sequences such as waterfalls, rivers, and fountains. To meet such a requirement, advected textures techniques are sometimes used (Neyret, 2003). Noise functions are often used for dynamic water surface motion (Perlin, 1985). However, these approaches cannot generate realistic movements since these are not physically-based methods.

In this paper, the main contribution is an effective approach for creating endless water animation. We use particles created by particle-based water simulation. In a preprocess, water flow is simulated using particles and the motions of the particles are stored in a database. Our method uses the database to create endless animations of water flow. Our method has the following three features:

- Since our method uses the particle database created by the fluid simulation, we can synthesize realistic motion of water with high visual quality.
- Our method synthesizes endless water animations from the particle dataset that are created for a short period of time.
- Our method is efficient since no fluid simulation is required once the particle dataset has been created.

The rest of this paper is organized as follows. In Section 2, we briefly discuss some related work to clarify the advantages of our method. Next, in Section 3, we describe our proposed method. Some experimental results are shown in Section 4. Finally, in Section 5, we conclude this paper.

2 RELATED WORK

This section describes some related work. First, we describe particle-based fluid simulation in Subsection

Sotozaki, M., Dobashi, Y. and Yamamoto, T.
Creating Endless Water Flow Animation using Particle Data.
DOI: 10.5220/0005746602340239
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 1: GRAPP*, pages 236-241
ISBN: 978-989-758-175-5

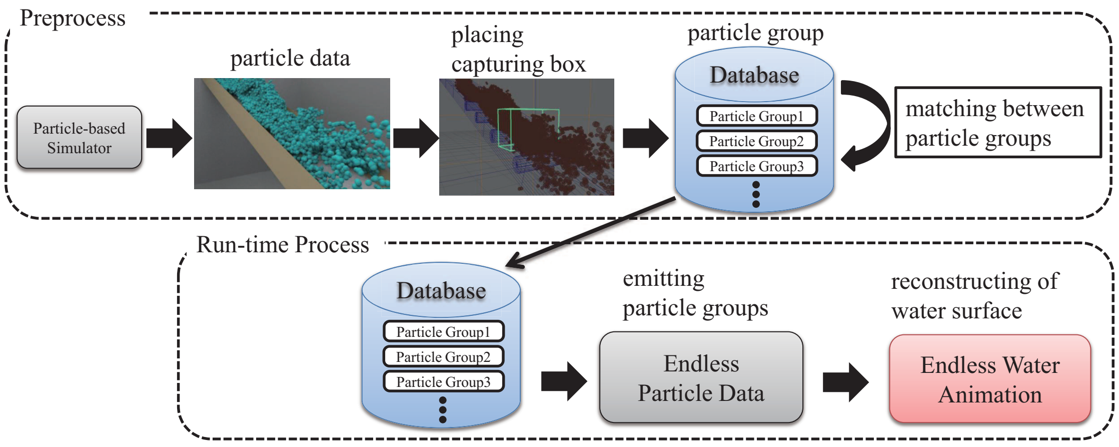

Figure 1: Overview of our method.

2.1. Next, we describe image-based methods for creating endless water animation in Subsection 2.2.

2.1 Particle-based Fluid Simulation

Using fluid simulation can generate realistic fluid animation. Particle-based simulations are popular for water simulation. Müller et al. proposed a particle-based simulation called Smoothed Particle Hydrodynamics (SPH) method (Müller et al., 2003). This method discretizes the fluid with particles, and solve Navier-Stokes equations to calculate water motions. Koshizuka et al. proposed Moving Particle Semi-implicit (MPS) mehod for handling incompressible fluids (Koshizuka and Oka, 1996). Zhu et al. developed a method called Fluid-Implicit-Particle (FLIP) which combines grid and particle based simulations (Zhu and Bridson, 2005). This method reduces errors by using particles for the advection process in fluid simulation. Recently, many methods have been proposed to improve the SPH method. Becker et al. proposed a method that enforces incompressibllity for SPH simulation (Becker and Teschner, 2007). Solenthaler et al. improved SPH to take long time steps (Solenthaler and Pajarola, 2009). Müller et al. proposed Position Based Dynamics (PBD) method (Müller et al., 2007). Macklin et al. proposed Position Based Fluids (PBF) using PBD for particle-based simulation (Macklin and Müller, 2013). This is suitable for parallelization using Graphics Processing Unit (GPU). Nugjgar et al. proposed creating endless water surface animation using Markov-Type Vector Field (Nugjgar and Chiba, 2013). This method employ surface model, our method employs particle database approach.

2.2 Image-based Methods

While fluid simulation generates realistic animation, one of the problems is its high computational cost. This is especially problematic for creating highly realistic images by increasing the number of particles. An alternative approach to the fluid simulation is to use videos of real water flow. This subsection discusses some of those methods using videos. Bhat et al. proposed the method generating seamless endless animation (Bhat et al., 2004). This method is able to synthesize and edit endless 2D animation using 2D video input. Okabe et al. proposed fluid video synthesise method from single view image with fluid video databases (Okabe et al., 2011). Since this method uses 2D images, so it cannot handle free viewpoints. Many methods that reconstruct water surfaces from images were proposed (Yu and Quan, 2013), (Li et al., 2013), (Ihrke et al., 2005), (Hilsenstein, 2005). However, these cannot express dynamic animation such as splash since 2D images do not have geometric information. Since our method uses simulated particles, we can handle arbitrary viewpoints and dynamic motions of water.

3 OUR METHOD

The proposed method is suitable for creating steady flows, such as river and waterfall. Figure 1 shows an overview of our method. As shown in Figure 1(a), we first creates the particle dataset by fluid simulation using the SPH method in a preprocess (Subsection 3.1).

3.1 Database Construction

The database of particles is constructed by running the particle-based water simulation as a preprocess.

Our method assumes steady flows such as streams and is not suitable for such a situation where water is poured into a glass. While simulation is running, we divide the particles into a set of groups, which we call *particle groups*. In the following, we explain how the particle groups are generated by using a 2D example shown in Figure 2. In order to simulate water flow, a particle emitter is placed at a position where the water flow begins. We also set a terminal boundary where particles are forcibly eliminated (see Fig. 2). The water flow is then simulated by computing the motions of the particles between the emitter and the terminal boundary. In order to create particle groups, a capturing box is placed at a user-specified position as shown in Fig. 2. At each time step of the simulation, our system checks the number of particles inside the capturing box. If the number exceeds N_{min}, the particles in the box form a particle group. N_{min} is specified by the user. The trajectories of the particles are recorded until they reach the terminal boundary. We repeat these processes until the number of particle groups reaches a user-specified number, M. In the following sections, let us denote i-th particle group by S_i, where $i = 0, 1, \cdots, M-1$.

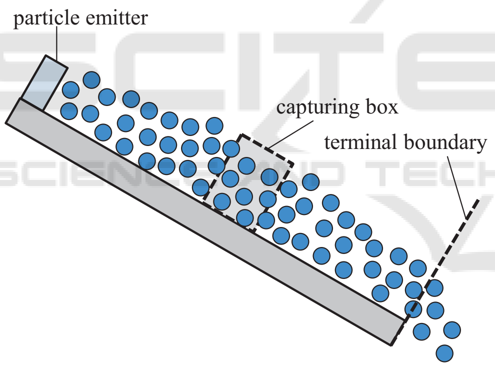

Figure 2: Creating particle groups.

3.2 Similarity between Particle Groups

Our method creates endless animation of water flow by repeatedly selecting a particle group and emitting the particles in the selected group. Our method selects the particle group sequentially based on the similarities between particle groups. This section explains the computation of the similarity between a pair of particle groups.

The similarity D_{ij} between two particle groups S_i and S_j is calculated in the following way. We use the distributions of the particles at the time step when they are in the capturing box. Although the numbers of particles in these groups are not necessarily the same, we assume that they are same. When the numbers

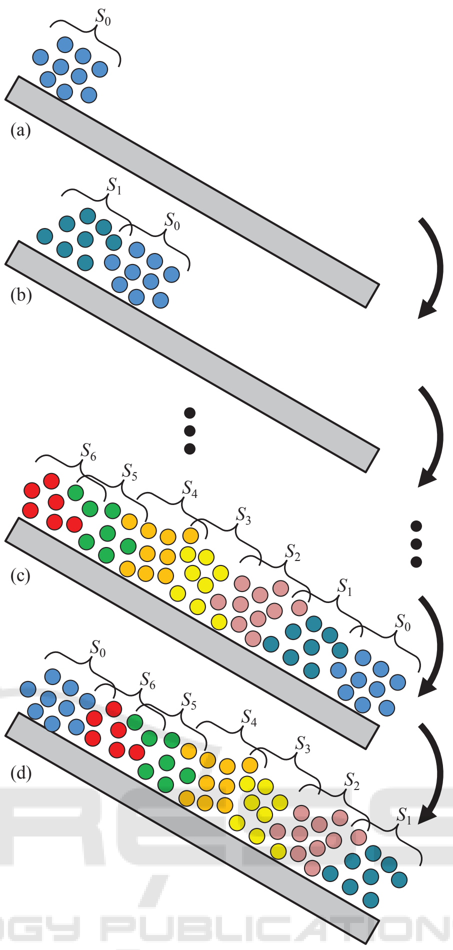

Figure 3: Creating endless water animation.

are different, we apply a clustering algorithm to one of the groups containing larger number of particles so that the number of clusters becomes the same as the number of the particle of the other group. Next, we compute a perfect matching between the particles in S_i and those in S_j such that the following cost function is minimized.

$$\sum_{i,j} ||x_i - x_j||^2, \qquad (1)$$

where x_i and x_j are positions of the corresponding particles i and j in the matching between S_i and S_j, respectively. This problem is equivalent to the minimum assignment problem. Our method solves this problem using Hungarian method (Kuhn, 1955). We use the minimum cost as the similarity D_{ij}.

3.3 Creating Endless Animation

We explain our method for creating endless animation of water flow by using Figure 3. Figure 3 shows a two-dimensional example for simplicity and uses seven particle groups, S_0, S_1, ..., S_6. We use similarity D_{ij} as the transition probability matrix and create the endless animation by randomly choosing the particle groups using D_{ij}.

First, our method emits particles in S_0 (see Figure 3(a)). We then select a particle group from the rest of the particle groups by using a random number obeying $D_{0j}/\sum_k D_{0k}$ so that a particle group similar to S_0 is likely to be selected. In Figure 3 (b), S_1 is selected and the particles in S_1 are emitted. As shown in Figure 3(b), the particles in S_0 and S_1 are overlapped in order to reduce the discontinuity between the particle groups. The next particle group is again selected in the same way by using a random number obeying $D_{1j}/\sum_k D_{1k}$. The endless water animation is created by repeating these processes.

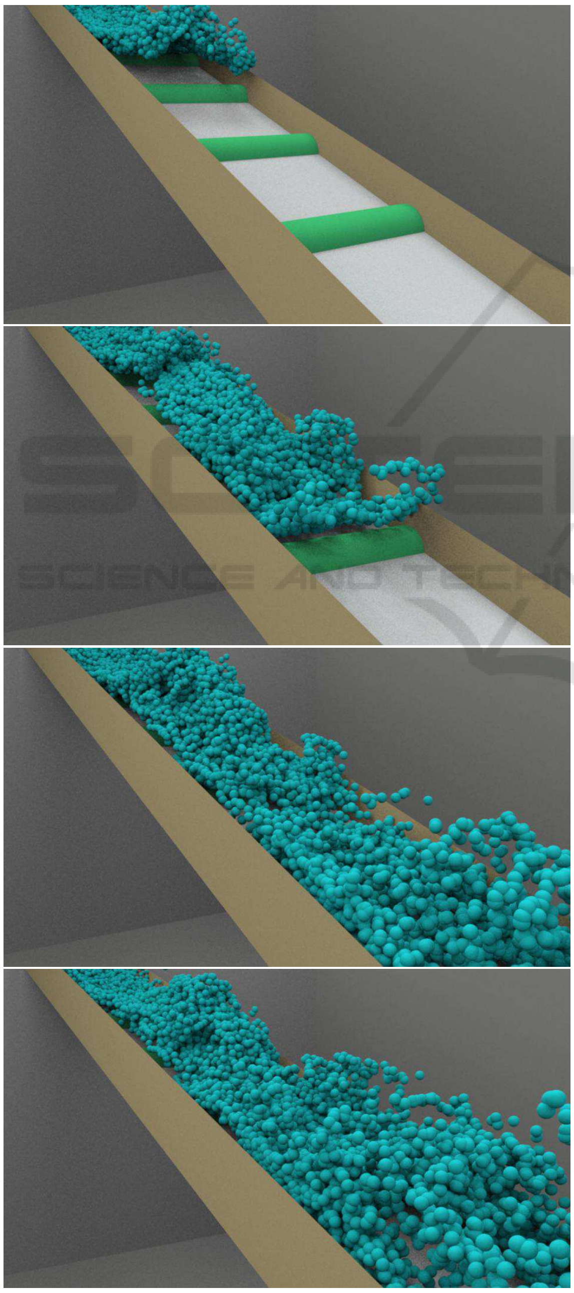
Figure 4: River results using our method.

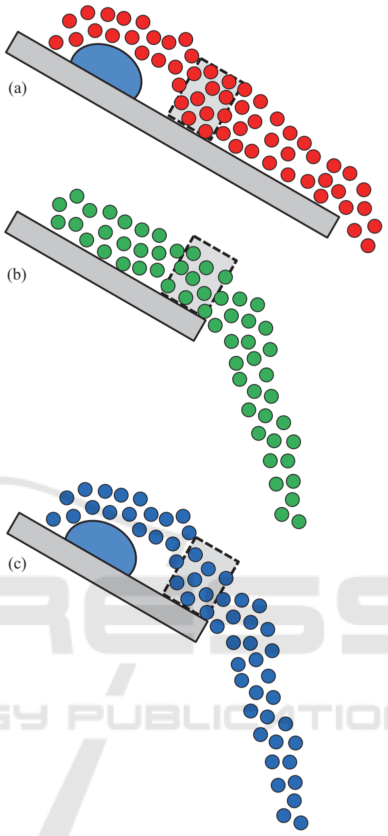

Figure 5: Blending particle groups. The top(a) and middle(b) figures show particles simulated with different conditions. The bottom figure shows particles created by blending particles in (a) and (b).

3.4 Blending Particle Groups

This section describes a method for synthesizing water flows by blending two sets of particle groups that are created by water simulations under two different conditions.

Let us explain the method by using Fig. 5, where the water flow shown in Fig. 5(c) is synthesized by blending particles shown in Figs. 5(a) and (b). Let us denote the two sets of particle groups, G_a and G_b, created by using the two results shown in Figs. 5(a) and (b), respectively. We first translate the positions of the particles so that the capturing boxes for G_a and G_b are at the same position. Then, the water flow is synthesized by switching the particles in G_a to the particles in G_b at the position of the capturing box. In order to achieve smooth transitions, a particle group in G_a is switched to a particle group in G_b that is most similar to the particle group in G_a in terms of the cost function expressed by Eq. 1. The positions of the particles are linearly blended between the corresponding particles.

3.5 Reconstruction of Water Surfaces

The water surface is reconstructed in the following way. We assign a kernel function for each particle. We use an isotropic kernel function (Müller et al., 2003). A grid covering all the particles is then generated and we accumulate the kernel functions of all the particles into the grid. The water surface is obtained by computing isosurfaces by using the marching cubes method (Lorensen and Cline, 1987). However, when using this method, the discontinuity between the particle groups would be visible. As described in the previous section, the successive particle groups overlap to avoid this problem. The kernel functions of the particles in the overlapped region are blended together so that the discontinuity is less visible. Figure 6 shows an example of water surface reconstruction.

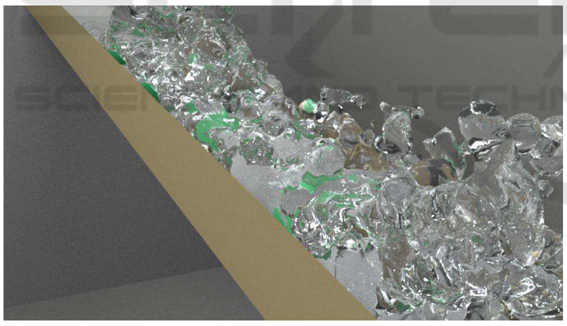
Figure 6: The example of water surface reconstruction.

4 RESULTS

This section shows results using the proposed method. The examples shown in this section are calculated on a PC with Intel Core^{TM}i5 2.7GHz(CPU) and 8GB memory. We use a commercial software (RealFlow2013,) for creating the particle groups. Table 1 shows the calculation time. The images are rendered by Autodesk Maya 2016(Maya2016,).

Figure 4 shows an example of a synthetic river generated by using proposed method. The number of particles is approximately 20,000. Twenty particle groups are created. The capturing box is placed in front of the particle emitter. This example demonstrates that our method successfully creates endless water animation

Figure 7 shows another example created by blend-

Table 1: Calculation times for the two examples. T_s is the calculation time for creating 200 frame particle data. T_{CPG} is the time for creating a single particle data. T_{CEA} is the for creating endless animation of 1,000 frames and T_{BPG} is the time for blending two particle groups consisting of 566 particles.

Snece	T_s	T_{CPG}	T_{CEA}	T_{BPG}
River	157s	0.001s	13.773s	—
Pipe	218s	0.001s	8.463s	664.573s

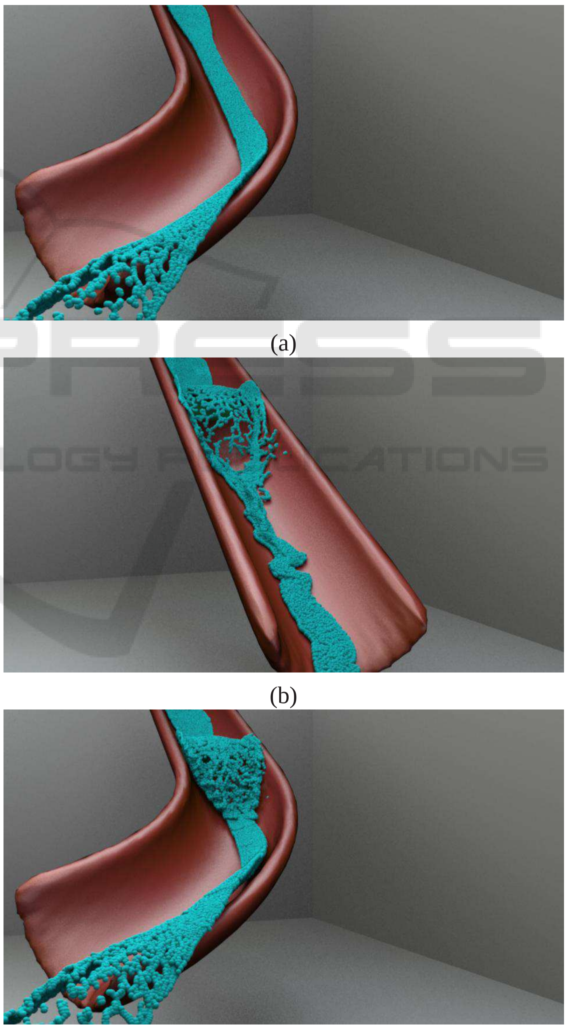
(a)
(b)
(c)

Figure 7: Water flow in the curved pipe. (a) is created by computing motions of particles in the curved pipe. (b) is created by computing particles in the straight pipe with an obstacle in the middle. (c) is created by blending particles in (a) and (b) using our method.

ing particles simulated with two different conditions. Fig. 7(a) and (b) show the two sets of the particles and Fig. 7(b) shows the particles created by blending the particles in Figs. 7(a) and (b).

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed the method for creating endless water animation using simulated particles. We also proposed the method for blending particles simulated under different conditions. The proposed method can create high quality animations with low cost by re-using the particle groups repeatedly.

There are several possible future works. First, we would like to extend our method to other natural phenomena, such as fire and smoke. Since our blending method cannot handle more than two sets of the particle groups, we improve the method to handle multiple sets of particle groups.

REFERENCES

Becker, M. and Teschner, M. (2007). Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 209–217. Eurographics Association.

Bhat, K. S., Seitz, S. M., Hodgins, J. K., and Khosla, P. K. (2004). Flow-based video synthesis and editing. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 360–363, New York, NY, USA. ACM.

Foster, N. and Fedkiw, R. (2001). Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 23–30, New York, NY, USA. ACM.

Hilsenstein, V. (2005). Surface reconstruction of water waves using thermographic stereo imaging. In *Image and Vision Computing New Zealand*, pages 102–107. Citeseer.

Ihrke, I., Goidluecke, B., and Magnor, M. (2005). Reconstructing the geometry of flowing water. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1055–1060 Vol. 2.

Koshizuka, S. and Oka, Y. (1996). Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*, 123(3):421–434.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Li, C., Pickup, D., Saunders, T., Cosker, D., Marshall, D., Hall, P. S., and Willis, P. (2013). Water surface modeling from a single viewpoint video. *Visualization and Computer Graphics, IEEE Transactions on*, 19(7):1242–1251.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA. ACM.

Macklin, M. and Müller, M. (2013). Position based fluids. *ACM Trans. Graph.*, 32(4):104:1–104:12.

Maya2016. `http://www.autodesk.com/maya/`.

Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 154–159. Eurographics Association.

Müller, M., Heidelberger, B., Hennix, M., and Ratcliff, J. (2007). Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118.

Neyret, F. (2003). Advected textures. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 147–153, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Nugjgar, P. and Chiba, N. (2013). Markov-type vector field for endless surface animation of water stream. *Vis. Comput.*, 29(9):959–968.

Okabe, M., Anjyo, K., and Onai, R. (2011). Creating fluid animation from a single image using video database. *Comput. Graph. Forum*, 30(7):1973–1982.

Perlin, K. (1985). An image synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 287–296, New York, NY, USA. ACM.

RealFlow2013. `http://www.realflow.com/`.

Solenthaler, B. and Pajarola, R. (2009). Predictive-corrective incompressible sph. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 40:1–40:6, New York, NY, USA. ACM.

Stam, J. (1999). Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 121–128, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Yu, M. and Quan, H. (2013). Fluid surface reconstruction based on specular reflection model. *Journal of Visualization and Computer Animation*, 24(5):497–510.

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