Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28- March 3, 2017

2D FLUID SHAPE DESIGN BY DIRECT MANIPULATION

Yuto Hayakawa† Issei Fujishiro†
†Graduate School of Science and Technology, Keio University

ABSTRACT

In this paper, we propose a simple interface for novices to design the shape of 2D fluid intuitively through direct manipulation, where parameters of fluid simulation are specified with hand motions to be captured. Recognizing specific hand motions as hand gesture commands allows the users to control the dynamics of 2D fluid at will, without detailed knowledge of fluid simulation.

1. INTRODUCTION

In recent video production, non-photorealistic fluid has been frequently used, where it is required not only to maintain fluid likeness, but also to control the fluid behavior as expected. The reality and controllability of fluid should be balanced carefully in visual simulation of fluid.

One of the approaches to fluid simulation is particle-based, which assumes that fluid is a set of particles, gives the physical quantity to each particle as a mass point, and moves the particles according to the governing equation. There are various parameters for particle-based fluid simulation. Since several kinds of control particles are used for fluid control, parameters for handling each control particle are also required. However, without underlying knowledge and rich expertise, it is difficult to adjust these parameters appropriately.

In this work, we propose a simple interface to design the shape of 2D fluid intuitively through direct manipulation, where parameters of fluid simulation are specified with hand motions to be captured by Leap Motion. Recognized specific hand gesture commands allow the users to control the dynamics of 2D fluid without detailed knowledge of fluid simulation. Our final goal is to develop a system that allows even novices to design their own fluid shape at will thanks to such direct manipulation.

The remaining sections of this paper are organized as follows. We introduce several related works in Section 2. After giving an overview of our approach in Section 3, we describe how to recognize hand gesture commands in Section 4 and how to perform fluid simulation in accordance with them in Section 5. In Section 6, we show the implementation details and preliminary execution results. Finally, we conclude the paper with a few remarks on future work in Section 7.

2. RELATED WORK

There are a number of known methods for fluid simulation. As an Eulerian approach, grid-based methods divide the simulation space by a grid and compute the physical quantities on the grid. On the other hand, as a Lagrangian approach, particle-based methods assume that fluid is a set of particles and take mass point for each of the particles. The particle-based methods are suitable for real-time applications because they have a limited realism but are relatively fast compared to the grid-based methods. Representative particle methods include SPH (Smoothed Particle Hydrodynamics) method and MPS (Moving Particle Semi-implicit) method. The difference between the SPH and the MPS lies in the way how to calculate the fluid pressure. The MPS method, which implicitly calculates the pressure, can simulate non-compressible fluid, but its calculation is relatively slow when compared to the SPH method. The SPH method simulates weak compression of fluid, but its computation is fast and realizes real-time fluid simulation. Because we aimed at an interactive system, it is necessary to use real-time fluid simulation that reflects the user's intention immediately. Thus, we prefer the SPH method [1].

In this section, we clarify the contributions of this paper by citing a few research attempts at fluid control. Thurey et al. [3] proposed a method for fluid control using control particles. In this method, control particles have the physical quantities but behave independently from the fluid particles. The method allows a user to freely move the control particles, which clothe themselves in the fluid particles. Zhang et al. [4] proposed a method to relate hand motions with control particles representing the mesh models and bone models.

In the work by Zhang et al. [4], all of the fluid shape changes are pre-determined by the input motion data. In our system, the user is allowed to control 2D fluid on the fly. In accordance with the intermediate result, he/she may specify the next fluid control. Such interactive fluid control can differentiate our system from the related work.

There also exist researches on fluid control systems by sketches. Ueda et al. [4] simulate blood flows on the surface of skin model along the user's sketch. Pan et al. [5] control the flow by specifying a keyframe and sketching for fluid simulation. Our system can model fluid such as shape control, and set the framework of fluid simulation such as configuring the range to be controlled.

Xu et al. [6] proposed a system to automatically generate 2D marbling. Our system could be used for generating similar textures.

Fluid simulation can be performed with computer graphics software such as blender [7]. When producing animations by fluid simulation with such a software, it takes a great deal of steps for parameter settings. However, even a single gesture command can be related to multiple parameter settings in our system. As a result, animation by fluid simulation can be produced with a relatively small number of steps.

3. APPROACH

Figure 1 shows the processing loop with our system. Hand motion data is acquired by Leap Motion and interpreted as a gesture command, which is classified into either of seven types. Then the system simulates the fluid behavior according to the gesture command. By projecting the 3D hands onto the 2D plane, they are being displayed over the simulated fluid. Figure 3 shows a snapshot of our system in actual use.

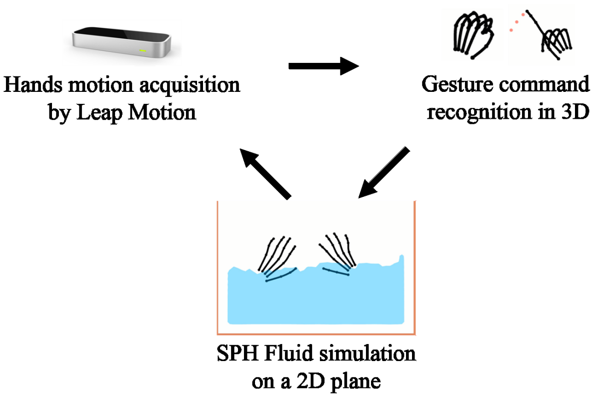


Figure 1: **Processing loop with our system**

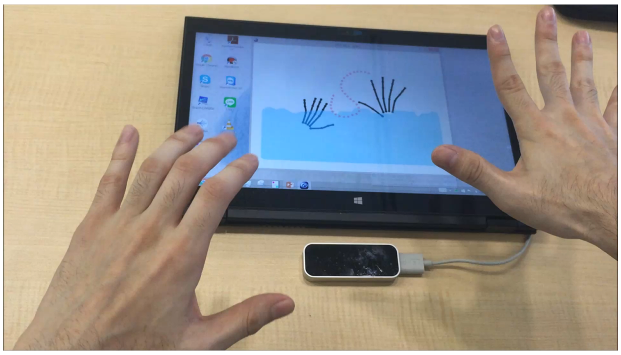


Figure 2: **Our system in actual use**

Table 1 gives the list of gesture commands that can be interpreted by our system. The hand motion data is recognized based on the combination of up-and-down movements of the ten fingers in 3D. Primary controls are given by left hand motions, while those controls can be assisted with left hand motions. In a case that a gesture command needs more than one specific motion data, the selection of the final gesture command is determined by the left hand.

Table 1: **Gesture command list. Right arrows indicate the state transition between the consecutive gesture commands**

No	Command name	Left hand	Right hand	Command semantics
1	Touching fluid			Touching fluid ON, OFF
2	Reset			Reset the execution state of the command and return to initial condition
3	Fluid shape control			Drawing control particles on the trajectory of index finger tip
				Deforming fluid to the trajectory-shaped fluid
4	Generating flow		2 → 	Generating flow on control particles. Change flow speed by moving right hand
5	Range specification	 or 		Specifying range of fluid particles or the control particles by the trajectory of index finger tip
				Specifying particles inside the range
6	Scaling fluid shape		4 → 	Specifying fluid shape inside the range
7	Viscosity control			Adjusting viscosity by interval of the index finger and thumb

The gesture command is identified from the vertical movement of each finger. The center of the palm and the position of each finger tip are found from the motion of both hands with their 3D positions. From these positions, a vector in the direction the hand is pointing from the palm and a vector vertically downward from the palm are calculated. Vertical movement of each finger is determined by the inner products of the vector from the center of the palm to each finger tip and the above two reference vectors.

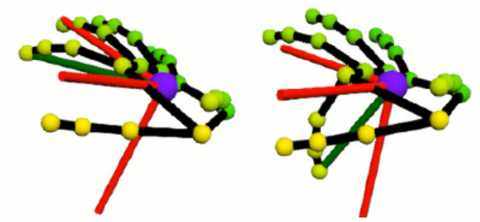


Figure 3: **Recognition of right hand index finger**

4. DIRECT FLUID CONTROL

We leverage the SPH method in order to simulate the fluid behavior. By solving the pressure of the governing equation explicitly, the resuting fluid becomes weakly compressible, and thus making the fluid simulation faster. Navier-Stokes equation is used as the governing equation of fluid:

$$\rho\frac{D\mathbf{v}}{Dt} = \mathbf{F}^{\mathbf{pressure}} + \mathbf{F}^{\mathbf{viscosity}} + \rho g + \mathbf{F}^{\mathbf{externals}}, \quad (1)$$

where **v** denotes the particle velocity. Navier-Stokes equation consists of pressure term, viscous term, gravity, and external force term. The external force term is used for fluid control of our method.

The following sections describe the control of the fluid simulation in accordance with the gesture commands.

4.1. FLUID SHAPE CONTROL

The fluid shape control relies on the method by Thurey et al. [3], which controls the fluid by the control particles. Any fluid shape can be drawn by attracting the fluid particles to the corresponding control particles. The attraction to control particles is incorporated into the external force term of Equation (1). For any particle i, the summation of attraction forces to all the control particles is represented by the following equation:

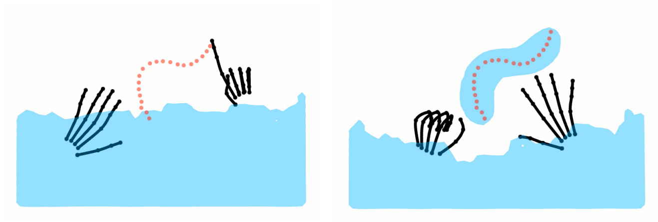
$$\mathbf{F}_i^{\mathbf{attraction}} = w\sum_{c_j\in C_i}\alpha_{c_j}\frac{\mathbf{r}_{c_j}-\mathbf{r}_i}{||\mathbf{r}_{c_j}-\mathbf{r}_i||}W(\mathbf{r}_{c_j}-\mathbf{r}_i,h), \quad (2)$$

where C_i denotes the set of control particles for fluid particle i, w a coefficient which acts as attraction when it is positive, and repulsive force otherwise, and α_{c_j} is the scale factor of attraction for control particle c_j:

$$\alpha_{c_j} = 1 - min(1, \sum_i V_i W(\mathbf{r}_{c_j}-\mathbf{r}_i,h)), \quad (3)$$

where V_i denotes the volume of fluid particles. The density of fluid particles that are attracted to control particles is adjusted to be uniform by the scale factor.

The gesture command is activated by raising only the right index finger, and then the control particles are drawn in accordance with the position of the moving right index finger tip, as shown in Figure 4. The fluid is deformed along the trajectory of the right index finger tip. After drawing of the control particles, attraction to the control particles acts on the fluid while clenching the left hand.



(a) **Drawing the control particles (red)** (b) **Deforming fluid along the control particles**

Figure 4: **Fluid shape control by control particles**

4.2. GENERATING FLOW

Flow field is generated simultaneously with interpretation of the gesture command for fluid shape control. The trajectory of the right index finger tip derives the flow field. Flow field is generated because the scale factor of Equation (3) makes the density of fluid particles around control particles uniform without sustaining attraction to the terminating control particles. Also, the velocity of flow field corresponds to the motion data, and changes the particle velocity of Equation (1).

The fluid, deformed by the gesture command of the fluid shape control in Section 4.1, induces the flow field by raising the right index and middle fingers, as shown in Figure 5. The velocity of the flow corresponds to the speed of moving the two fingers.

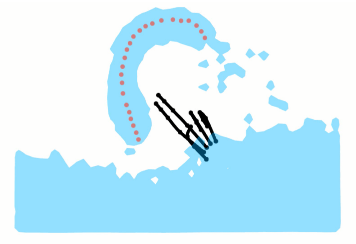
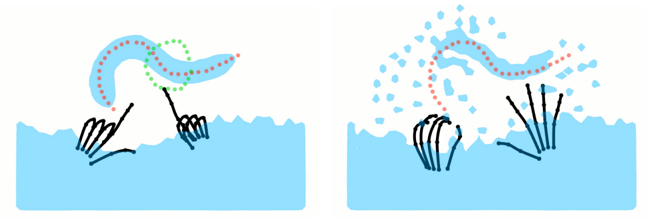


Figure 5: **Generating flow on the control particles**

4.3. RANGE SPECIFICATION

Range specification is performed either to the fluid particles or to the control particles. The fluid particles are selected by the motion raising the left index finger and thumb while the control particles are selected by raising the left index finger, middle finger and thumb. Inside or outside of the range can be determined by the Winding Number algorithm. Range specification starts with raising the right index finger, and then the particles in the range surrounded by the trajectory of the right index finger tip are specified by clenching the left hand. Figure 6 shows the result of range specification to the fluid particles. Only the specified fluid particles are subject to the gesture command of fluid shape control whereas the other particles are not affected by the attractive force.



(a) **Specifying range of fluid particles (green)** (b) **Shape control of only specified particles**

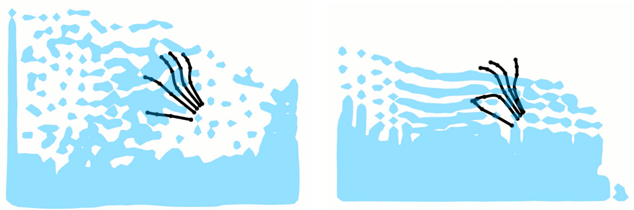
Figure 6: **Range specification for fluid particles**

4.4. SCALING FLUID SHAPE

Scaling fluid shape acts to control the particles that are specified by the gesture command of range specification. The interval of the specified control particles can be narrowed by raising the right index finger and middle finger while raising the right thumb. If the interval of control particles is below a pre-determined value, the interval spreads. The interval of control particles can be set arbitrarily because this gesture command can be stopped by raising all the right fingers.

4.5. VISCOSITY CONTROL

Viscosity of the fluid can be controlled globally. The viscosity coefficient in the governing equation decides the motion data. Viscosity control is activated by putting together the right index finger tip and the right thumb tip. After the maximum value of the viscosity coefficient was set by the motion putting together these two fingers, the viscosity coefficient can be adjusted by the interval length. Figure 7 compares the two fluids whose viscosity values are adjusted by this command.



(a) **Splash of low viscosity fluid** (b) **Splash of high viscosity fluid**

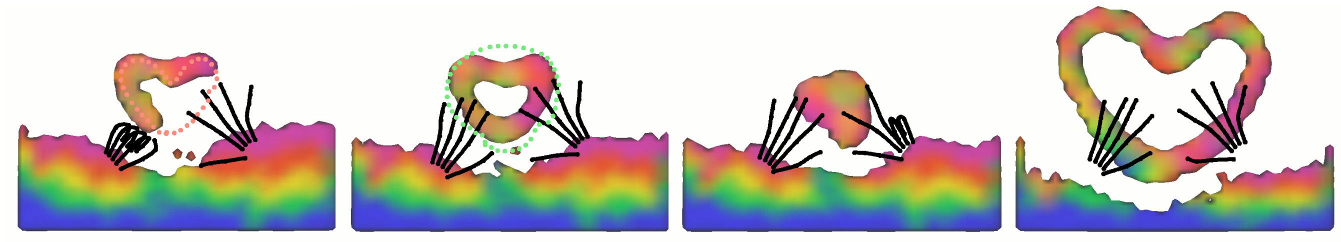
Figure 7: **Global viscosity control**

4.6. TOUCHING FLUID

Collision detection of the hands with the fluid can be controlled by a specific gesture command, which is activated by putting together all the right fingers except thumb. Touching fluid is realized by placing the fluid particles along the handprint displayed on the 2D place.

In addition to the control of fluid behavior through the gesture commands, it is possible to guide the control through giving a direct touch feedback of the fluid. Figure 8 shows the collision between the right hand and the fluid.

Figure 8: **Collision between the right hand and the fluid**



(a) **Control particles in a heart shape** (b) **Specifying control particles** (c) **Fluid ball** (d) **Deforming to heart shape**

Figure 9: **Composite operation example: Deforming the fluid ball to heart shape**

5. IMPLEMENTATION AND RESULTS

The system is running in real-time with 600 particles on a standard PC (CPU: Intel Core i7-5500U 2.40GHz, RAM: 8.00GB). Fluid mixing effect can be visualized by the use of colored particles which are initially placed in a horizontal stripe fashion.

We deformed a fluid ball into a heart-shape by a sequence of gesture commands, as shown in Figure 9. The initial fluid ball was generated in the following way: control particles, originally arranged to the shape of a heart by fluid shape control (Figure 9(a)), were specified (Figure 9(b)) and brought closer to each other by scaling fluid shape. Then, by expanding the fluid ball (Figure 9(c)), deforming the fluid to the heart shape (Figure 9(d)) can be animated.

6. ONGOING ISSUES

Rendering issue remains as a future work. Even in 2D, generating realistic appearance of fluid similar to water is the vital element for enriching video works.

When considering a 3D extension, the present direct manipulation would be expected to become more beneficial for intuitive and on-the-fly modeling of fluid shape. Current extension of our system to 3D is shown in Figure 10.

ACKNOWLEDGMENT

This work has been supported in part by JSPS KAKENHI under the Grant-in-Aid for Scientific Research (A) No. 262440015.

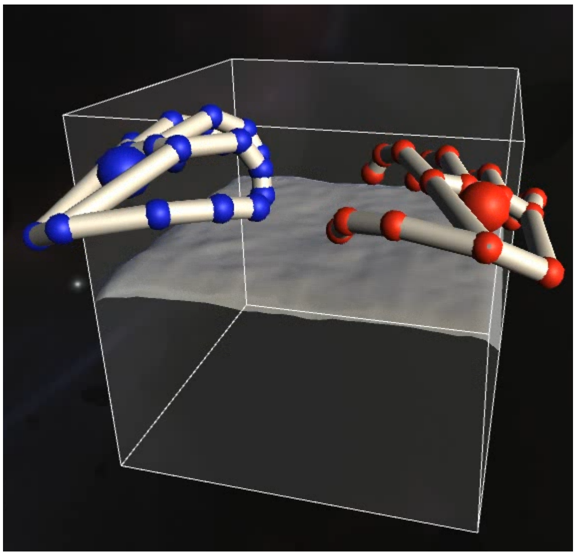


Figure 10: **A sample scene of 3D extension**

REFERENCES

[1] M. Muller, D. Charypar, and M. Gross: “Particle-based Fluid Simulation for Interactive Applications,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, pp. 154-159, 2003.

[2] K. Ueda and I. Fujishiro: “Adsorptive SPH for directable bleeding simulation,” in *Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, pp. 9-16, 2015.

[3] N. Thurey, R. Keiser, M. Pauly, and U. Rude: “Detail-preserving fluid control,” in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 7-12, 2006.

[4] S. Zhang, X. Yang, Z. Wu, and H. Liu: "Position-based fluid control," in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, pp. 61-68, 2015.

[5] Z. Pan, J. Huang, Y. Tong, C. Zheng, and H. Bao: "Interactive localized liquid motion editing," in *ACM Transactions on Graphics*, Volume 32, Issue 6, Article No. 184, 2013.

[6] X. Mao, T. Suzuki, and A. Imamiya: "AtelierM: A physically based interactive system for creating traditional marbling textures," in *Proceedings of ACM 1st International Conference on Computer Graphics and Interactive Techniques in Australia and South East Asia*, pp. 79-86, 2003.

[7] *blender*, https://blender.jp/ (accessible on January 22, 2017).