Fluid Sketching—Immersive Sketching Based on Fluid Flow

Sevinc Eroglu* Sascha Gebhardt† Patric Schmitz‡ Dominik Rausch§ Torsten Wolfgang Kuhlen¶

Visual Computing Institute
RWTH Aachen University

Figure 1: Artist performing a sketching session with our Fluid Sketching application.

Abstract

Fluid artwork refers to works of art based on the aesthetics of fluid motion, such as smoke photography, ink injection into water, and paper marbling. Inspired by such types of art, we created Fluid Sketching as a novel medium for creating 3D fluid artwork in immersive virtual environments. It allows artists to draw 3D fluid-like sketches and manipulate them via six degrees of freedom input devices. Different brush stroke settings are available, varying the characteristics of the fluid. Because of fluids' nature, the diffusion of the drawn fluid sketch is animated, and artists have control over altering the fluid properties and stopping the diffusion process whenever they are satisfied with the current result. Furthermore, they can shape the drawn sketch by directly interacting with it, either with their hand or by blowing into the fluid. We rely on particle advection via curl-noise as a fast procedural method for animating the fluid flow.

Index Terms: Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality; Human-centered computing—Human computer interaction (HCI)—Interaction devices—Sound-based input / output; Human-centered computing—Human computer interaction (HCI)—HCI design and evaluation methods—User studies

*e-mail:sevinc.eroglu@rwth-aachen.de
†e-mail:sascha.daniel.gebhardt@rwth-aachen.de
‡e-mail:patric.schmitz@cs.rwth-aachen.de
§e-mail:dominik.rausch@rwth-aachen.de
¶e-mail:kuhlen@vr.rwth-aachen.de

1 Introduction

Recently, the interest in Virtual Reality (VR) has been increasing in a wide range of areas from academic research to engineering, design, business, entertainment, and arts. Considering the area of art, still only few VR sketching approaches exist. Yet, VR allows for drawing and freely expressing ideas directly in a 3D context. Artists can draw and perceive their creations in life-size, unconstrained by the two-dimensional world of paper or computer monitors.

We present an immersive sketching system that enables the creation of 3D structures that look and behave like fluids. This work is motivated by the complexity and beauty of the natural phenomena in fluid flow, like the interaction between fluids of different viscosities or trailing clouds of swirling smoke. Our goal is to provide artists with a tool that enables them to capture the beauty of these phenomena, but with more control and possibilities than in the real world.

Paper marbling is a traditional form of fluid artwork. The basic process comprises placing water in a tray and sprinkling or dropping oil-based inks on the surface. By using tools like metal wires, styluses, or combs, artists can shape the oily surface and transfer the result to a sheet of paper [29]. However, the ongoing diffusion process demands high levels of expertise and training, since it is impossible to undo any step of the creation process.

Another example of fluid art is photography of twirling smoke or ink injection into water. Creating such art is a challenging because it requires the choice and preparation of a complex setup of equipment, camera and light settings, and the right environmental controls [2]. Even when the preparation is complete, artists do not have exact control of the smoke or ink due to their nature.

As illustrated by these examples, the reason for its beauty is also the main limiting factor for creating fluid artwork: its diffuse and volatile behavior. With our solution, we aim to overcome this by combining a natural interaction concept with a high degree of control of the fluid behavior. While interaction is the main focus of our work, we also need to address the topic of fluid dynamics—the study of motion of fluids such as liquids, plasmas, gasses, and plastic solids [19]—to provide a fast and aesthetic fluid simulation and rendering system that suits our interaction needs.

Our resulting Fluid Sketching system enables artists to draw 3D fluid-like sketches in immersive virtual environments (IVEs) and manipulate them via direct hand interaction and a blowing metaphor. Users can configure various attributes of the fluid and the diffusion process itself. Considering the computational cost of fluid dynamics simulations, we aim to overcome the expensive computation for simulating fluids in real-time. To this end, we use a procedural method because of its low computational cost and the high degree of animator control.

In summary, the main contributions of our work are as follows: first, we introduce a novel medium for the creation of 3D fluid artwork. Second, we present a novel blowing-based interaction metaphor for the manipulation of such artwork. Third, we demonstrate the usefulness of our approach by means of a qualitative user study with artists and VR experts.

We present our work by first giving an overview of related work from the fields of immersive sketching and real-time fluid dynamics. Next, we outline the technical aspects of our Fluid Sketching system, before moving on to our interaction concept. Afterwards, we present the qualitative user study before concluding our paper.

2 Related Work

Regarding immersive sketching interfaces, Galyean and Hughes [11] introduce a voxel-based sculpting tool that mimics working with clay or wax. They provide various tools to add to or to cut away from the material. Sachs et al. [37] present a CAD system for designing 3D shapes using six degrees of freedom (6DOF) input for both hands. Models solely consist of lines, and users construct shapes in four steps: drawing, editing, fitting surfaces to groups of linked curves and deforming these surfaces to obtain the desired detail. With Holosketch, Deering [8] presents the first sketching system in a head-tracked stereo VR environment, allowing for the creation and manipulation of 3D geometries.

CavePainting [21] is the first sketching system designed for CAVE environments. It provides various brush types that can be selected by dipping a tracked paint brush into virtual cups on a table interface. Some brushes exhibit complex behavior based on physical properties, like paint shooting out in the direction of the brush, falling with gravity, or splatting against the CAVE walls. FreeDrawer [42] permits the creation and manipulation of spline-based freeform surfaces on the responsive workbench. Based on the experience with this system, the authors argue for the lack of force-feedback to be a drawback for immersive sketching applications. To address this issue, Drawing on Air [20] is presented, which is a haptic-aided input technique for drawing 3D curves in a fishtank VR setup. Its key feature is adjusting parameters of the line (orientation, thickness, and color) as it is being drawn. Rausch et al. present a sketching interface for architectural modification in IVEs [34]. Users can create line drawings to add annotations and objects to existing architectural models. The authors further extend the system with a sketch-recognition framework [35] providing a set of 28 commands and corresponding symbols. Lift-Off [17] is an immersive 3D modeling interface to create complex models with a controlled, handcrafted style. Artists start by creating 2D sketches on paper, which they can then import into the application. By automatically detecting curves in the drawings, the system allows artists to transform it into 3D shapes via extrusion and surface creation through sweeping. Tilt Brush[1] is a commercial VR sketching application that offers 24 types of brush strokes, including effects such as smoke, fire, lightning, and falling snow.

Like most of the approaches for immersive sketching presented above, our sketching system enables users to take advantage of 6DOF input for creating 3D artworks and to explore them in an IVE. However, our work differs in that we enable users to create fluid artworks that are not based on geometric meshes but on particle populations. Furthermore, we extend the existing body of research with a novel blowing metaphor.

Regarding the accurate simulation of fluid dynamics, today's de-facto standard is solving the Navier-Stokes equations, which describe the behavior of fluids as a function of pressure, velocity, and time [39]. There are two common methods to solve these equations: grid-based (Eulerian) and particle-based (Lagrangian) methods.

Stam [40] presents a grid-based real-time fluid simulation running on the CPU. Based on Stam's algorithm, Crane et al. [7] implement real-time fluid simulation on the GPU, using the MacCormack scheme to improve advection to second order accuracy. Further performance improvements are achieved by using multigrid solvers [41].

Smoothed particle hydrodynamics (SPH) [13,27] is well known for particle-based fluid simulation. Desbrun and Gascuel [9] use SPH to animate highly deformable bodies and Müller et al. [30] implement interactive fluid simulation based on it. Various extensions and improvements on the SPH concept in the computer graphics area exist [16].

Fluid-implicit particle (FLIP) is a hybrid approach that uses a grid for solving pressure and calculating velocity changes, which are then followed by particles. The technique is used and adapted by Zhu and Bridson [43] for animating sand as a fluid and by Greenwood and House [14] to simulate air bubbles.

Recently, machine learning is used for animating fluids. Jeong et al. [24] present a regression-forests-based approach for fluid simulation by using SPH. Their GPU implementation is capable of handling > 2 million particles in real-time. Chu et al. [6] use convolutional neural networks to generate a descriptor to encode the similarity between fluid regions with varying resolution.

Fluid dynamics are used in simulations of the previously discussed marbling art. The first digital marbling system uses 2D Navier-Stokes equations [4]. Acar and Boulanger present a generic tool that observes a multiscale fluid model for digital marbling [3]. By using small-scale mesoscopic details, they simulate complex fluid behaviors at the macroscopic level. While these approaches can model complex flows, they are not suitable for real-time applications. By solving the Navier-Stokes equations on the GPU, real-time marbling simulations have been developed [18, 26].

Procedural methods, while not physically correct, are a plausible and less computationally demanding alternative to computational fluid dynamics approaches. To mimic the appearance of fluid dynamics with small-scale turbulent details, wavelet turbulence [22] and curl-noise [5] are used. In Smoke Brush [1], smoke-like turbulence for 2D digital paintings is simulated with curl-noise. It allows artists to paint animated smoke effects on a digital canvas via tablet controls, also allowing to add them to existing images.

Drawing with the Flow [38] is a 2D sketch-based interface that enables users to draw directly on a fluid-flow dataset, where the ink follows the flow-field. Forbes et al. [10] present an interactive fluid simulation, which can be influenced by parameter adjustments and can be integrated into media arts projects.

Considering the body of existing research and performing tests with previous approaches, we rely on a curl-noise based procedural method to realize our Fluid Sketching system. This way, we can handle large enough particle populations in real-time to create detailed immersive fluid artworks. While the resulting fluid simulation is not physically correct, it is still plausible and appealing. It also offers a higher degree of artistic control than strictly physical models or data driven approaches.

3 Fluid Simulation

Our fluid simulation system is based on the procedural curl-noise approach of Bridson et al. [5]. It creates plausible fluid advection fields from the curl of a Perlin noise field [31], while consuming only

[1] https://www.tiltbrush.com/

little computing power. The resulting velocity field is divergence-free, which is an important attribute of incompressible fluids.

To create realistic fluid behavior, the velocity field changes over time, such as, e.g., in FlowNoise [33], by using a time-varying noise function $\vec{\psi}(\vec{x},t) = \vec{N}(\vec{x}/L,t)$. If the variable $\vec{x}$ of the noise function is scaled by $1/L$, the partial derivatives of this function vary over a length scale L and determine the diameter of vortices. We use four-dimensional Simplex noise [32] instead of the original Perlin noise [31], with the time variable in the fourth dimension. It allows for higher-dimensional noise fields with less computational cost, has easily computable analytic derivatives, and allows for a more efficient implementation in hardware compared to Perlin noise.

Bridson et al. suggest summing up several octaves of the noise function at different scales to produce turbulent structures. Lewis Fry Richardson describes their self-similar appearance as follows: "Big whirls have little whirls that feed on their velocity, and little whirls have lesser whirls and so on to viscosity" [36]. To describe this kind of self-similar structure, fractional Brownian motion (fBm) was first introduced by Andrei Nikolaevich Kolmogorov [23] and later studied by Mandelbrot and Van Ness [28]. It is the summation of several evaluations of a noise function at different frequencies (scales) that are varying in amplitude. Each of these components is called an octave. For each octave, the frequency increases by a lacunarity factor and the amplitude decreases by a gain factor, which is also called persistence.

We use the noise function of Bridson et al. [5] for the fBm. To compute the curl of the potential field $\nabla \times \vec{\psi}$, we require the derivative of the potential field $\nabla \vec{\psi}$, which is generated by summing the noise derivatives at time t at different scales:

$$\nabla \vec{\psi}(\vec{x},t) = \sum_{i=0}^{O-1} p^i \nabla \vec{N}(\vec{x} \cdot l^i / L, t) \quad (1)$$

Here, L is the vorticity scale, p is the persistence, l is lacunarity, and O is the number of octaves. L and p are user-configurable factors that enable interactive control of the fluid behavior. The scale of vortices can be varied from large to small by the scale factor L, and more to less turbulent behavior can be achieved by changing the persistence factor p. With a higher value of O more physically plausible turbulence behavior can be simulated at the cost of decreasing simulation speed. We use $O = 3$ and $l = 2$, which we found to produce plausible results while maintaining interactive frame rates. In the implementation, we add a constant offset to each dimension of the noise function to make the potential field uncorrelated.

3.1 Particle System

Particles are used for the visual fluid representation and the positions are advected in the velocity field using first-order Euler integration. Computation of the velocity field and particle advection are implemented on the GPU. Several additional properties are associated with each particle: the velocity from the previous iteration, particle color, a damping coefficient, a freeze state, and particle age. All of these are used to realize certain behaviors, which we illustrate in the following.

To reduce the number of particles to simulate and render, particle positions are constrained to the interaction volume of the target virtual environment. Particles leaving the interaction volume are temporarily marked inactive and will not be considered for computation and rendering. This yields an increase in application performance, while only disregarding particles the user would not be able to interact with anyways.

Driven by the velocity field, all particles move independently of each other. To increase the plausibility of their behavior, we model interaction between particles on top of this. Since modeling all pair-wise interactions would be prohibitive, we implement an approximation based on a low-resolution grid. Each grid cell stores the average velocity of all contained particles and each particle is affected by its current and neighboring cells' average velocities.

3.2 External Influences

To enable direct interaction with the fluid, we can apply external velocities to the particle system. The overall external velocity is the sum of a spherical velocity term $\vec{v}_{sphere}$ and a conical velocity term $\vec{v}_{cone}$.

Particles inside a spherical region are affected by the movement velocity of that region, weighted with a linear fall-off based on the distance to the sphere center and a configurable strength factor $f_{strength}$:

$$\vec{v}_{sphere} = \frac{\Delta \vec{p}_{sphere}}{\Delta t} \left(1 - \frac{\|\vec{p}_{sphere} - \vec{p}\|}{r_{sphere}}\right) f_{strength} \quad (2)$$

Here, $\vec{p}_{sphere}$ denotes the sphere's position, $\Delta \vec{p}_{sphere}$ is the change in position since the last frame, Δt is the time between the current and the last frame, $\vec{p}$ is the particle position, and r_{sphere} is the sphere radius. By using a linear fall-off weight, continuity between neighboring particles in the interaction range is achieved. Particles that are closer to the sphere center are influenced more strongly than distant ones, which we found to produce a more plausible behavior.

The influence on particles that lie within a conical region is given by $\vec{v}_{cone}$ in Equation 3. It depends on the direction $\vec{d}_{particle-apex}$ of the ray connecting the particle position $\vec{p}$ and the cone apex $\vec{p}_{apex}$, two linear fall-off weights and a strength factor $f_{strength}$. The angle θ is computed between $\vec{d}_{particle-apex}$ and the cone axis.

$$\vec{v}_{cone} = \vec{d}_{particle-apex} \left(1 - \frac{\|\vec{p}_{apex} - \vec{p}\|}{h_{cone}}\right) \left(1 - \frac{\theta}{\alpha}\right) f_{strength} \quad (3)$$

The first linear fall-off weight in Equation 3 is based on the distance between the particle and the cone apex. It leads to particles near the apex of the cone being influenced stronger than particles which are closer to the base, where h_{cone} denotes the height of the cone. The other fall-off weight is calculated depending on the angle θ. It is computed as $(1 - \frac{\theta}{\alpha})$, where α represents the half opening angle of the cone. Therefore, particles that are closer to the center axis of the cone are affected more strongly than the ones which lie at the sides.

The combined velocity $\vec{v}_{sum}$ of each particle is determined as the weighted sum of all influences described above. First, the velocity from the previous advection iteration is used to achieve a smoothing effect and thus temporal consistency of the particle movement. Next, the average velocities of the grid cell containing the particle and its neighboring cells are added to simulate particle–particle interaction. The overall external velocity is added, which is the sum of all external spherical and conical influences. Last, the velocity of the noise field is added for the fluid-like turbulence.

Finally, we account for liquids with different viscosities. Since viscosity has a resisting effect on the movement of particles, we simulate it as dampening, which has been used before to mimic viscous materials or air resistance [25]. The final particle velocity $\vec{v}$ is calculated by scaling $\vec{v}_{sum}$ with a normalized damping factor k_d:

$$\vec{v} = (1 - k_d)\vec{v}_{sum} \quad (4)$$

The new particle position is then computed as $x(t + \Delta t) = x(t) + \vec{v}\Delta t$.

4 Fluid Sketching

This section covers the user interface of the Fluid Sketching system. We first outline the technical setup of our test implementation. Next, we detail our interaction concept, before briefly explaining our rendering technique.

4.1 Technical Setup

We implemented our prototype for the five-sided aixCAVE (four walls and a floor). Its footprint of 5.25 m × 5.25 m and the height of 3.30 m allows for drawing impressive life-sized fluid sketches without the need of navigation methods apart from physical walking. Head tracking is provided via an ART opto-electronic tracking system, while stereopsis is realized via 120 Hz active stereo. The system has 25 nodes, each equipped with two Intel Xeon Westmere CPUs (6 Cores at 2.7GHz), 24 GB RAM, and two NVIDIA Quadro 6000 GPUs.

Users are equipped with three different interaction devices. An ART Flystick2 provides three types of input: its 6DOF transformation, buttons, and an analog joystick. The trigger button at the front is used for creating brush strokes. The outer left of the four buttons on the top opens the system control interface, which is realized as Extended Pie Menus [12]. In the menus, user-configurable parameters can be set, which are described later on. The other buttons on the Flystick2 give users quick access to the most frequently used features: the inner left button activates freeze mode, the inner right button toggles the eraser and the outer right button toggles pause mode.

Users can manipulate existing sketches with two different interaction metaphors: direct hand interaction and blowing into the sketch. For performing direct hand interaction, the non-dominant hand is equipped with an ART hand tracking target. The blowing metaphor is realized via a Sennheiser EW G2 transmitter in combination with a Sennheiser ME3 wireless head-mounted microphone.

4.2 Brush

The primary drawing tool of Fluid Sketching is the 3D brush, which emits particles from the tip of the Flystick2. To allow for a high degree of flexibility and creativity, various attributes of the brush are user-configurable via respective menus: brush size and color, viscosity, density, and initial speed of the emitted particles.

Emitted particles are uniformly distributed in the volume of a sphere, whose diameter corresponds to the brush size. The limited frame rate of the tracking system (60 Hz) results in gaps between measured emitter positions for fast movements. To mitigate this, the center of the emitting sphere is shifted randomly between the previous and current position of the brush for each particle. To obtain smooth, continuous brush strokes, cubic Hermite spline interpolation is applied.

Users can define the viscosity for individual brush strokes to vary between low-viscous fluids like water and high-viscous ones like honey. This assigns the normalized damping factor k_d in Equation 4.

The density parameter of the brush defines the number of emitted particles relative to the brush size. The number of emitted particles n is calculated as $n = {f_{density}}^3 \|\vec{p}_1 - \vec{p}_0\|$, where $f_{density}$ is the density and $\|\vec{p}_1 - \vec{p}_0\|$ is the length of the current frame's brush stroke segment.

Particles can be emitted with a user-configurable initial speed. If an initial speed is set, the direction of the initial particle velocity corresponds to the pointing direction of the Flystick2. This creates the effect of an aerosol spray can.

All of the parameters presented above are per-particle attributes, which are permanently assigned upon particle emission. Thus, users are given a high amount of control when drawing brush strokes, e.g., to combine fluids with different viscosities and densities.

4.3 Fluid Configuration

Users can change three parameters of the particle simulation: turbulence, vorticity, and diffusion speed. The properties have a global effect on the fluid behavior, so users immediately perceive the effects while changing them. The *turbulence* corresponds to the persistence p of the fluid advection (see Equation 1). To grant users a more intuitive understanding, we chose to present the less technical parameter name. The *vorticity* parameter corresponds to L and allows users to obtain vortices with varying sizes as illustrated in Fig. 2. The *diffusion speed* parameter controls the speed of the particle simulation. It gives users a high amount of control on the diffusion process, since small speeds allow for very precise working in the flow field, while high speeds produce time-lapse like effects.

To give more control of the overall amount of diffusion, we add an age-based settling effect for particles, which can be enabled or disabled according to users' needs. It gradually diminishes the diffusion for each individual particle over a user-specified diffusion time. If the effect is enabled, the diffusion process will decelerate and finally settle after the diffusion time has passed. Settled particles can be reactivated by directly interacting with them, either by hand or blowing interaction.

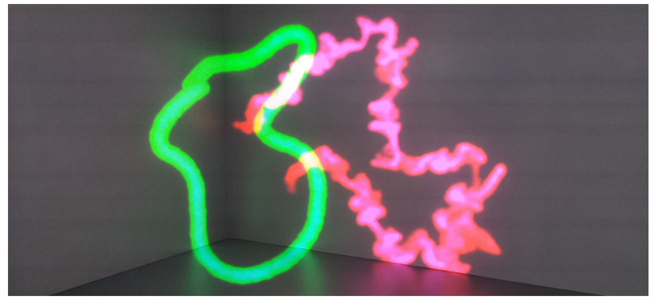

Figure 2: Strokes with different vorticities (green: low, pink: high).

4.4 Source–Target Modulation

The brush and fluid parameters presented in the previous sections can be changed between brush strokes, giving rise to a variety of stylistic choices for the drawing tool. However, it might be desired by users to change parameters *while* performing a stroke, just as, e.g, a painter is able to modulate the tilt angle and applied pressure of a brush. To account for this, we enable users to dynamically change predefined parameters while performing a brush stroke: *source–target modulation*. The brush size, color, and density can be manipulated over the course of a single stroke.

Each of these parameters—the *targets*—can be controlled by different *sources*: the X/Y direction of the Flystick2's analog joystick and the brush stroke velocity. They can be bound to the target parameters with a weighting factor ranging from zero to one. Brush configuration as described in Section 4.2 sets the base values for the brush size, color and density attributes. When picked as targets for the modulation, brush size and density are added as offsets.

A single target can be modulated by multiple sources. When a modulation is active, the offset value is added with its respective weight. This leads to the computation of the final attribute value A as the sum of A_{base} and the per-modulation offset values A_m weighted by the source attribute value s_m and an individual weighting coefficient w_m:

$$A = A_{base} + \sum_{m \in M} s_m w_m A_m \qquad (5)$$

Since subtracting color values from a base color can be non-intuitive depending on the color space, the color in the modulation settings is a target value instead of an offset. Thus, the brush color varies between the base color and one or more target colors. The interpolation is computed as a convex combination between the base color and the weighted target colors. A convex combination is a weighted linear combination of values $v = \sum_i w_i v_i$ with $w_i > 0$ and $\sum_i w_i = 1$.

The weights α_m for the convex interpolation are the product of the source modulation magnitude s and the modulation weight w_m, normalized by the number of active modulations N:

$$\alpha_m = \frac{|s| \cdot w_m}{N} \qquad (6)$$

For the convex interpolation weights to sum up to one, the base color weight α_b is calculated as one minus the sum of the target colors' interpolation weights. The sum of the target weights is in the interval $[0,1]$ due to normalization and the fact that all w_m and $|s|$ also lie in that interval. This leads to the interpolation formula of the resulting particle color C as a convex combination of the base color C_b and the per-modulation target colors C_m:

$$C = \alpha_b C_b + \sum_{m \in M} \alpha_m C_m \qquad (7)$$

By using source–target modulation, artists have a tool to increase the expressiveness of their brush strokes. To give an example: the joystick X direction is bound to the brush size and the velocity of the brush stroke is mapped to a target color. Then both parameters can be varied simultaneously during a brush stroke by moving the joystick while drawing at different velocities. An exemplary outcome of a source–target modulation is depicted in Fig. 3.

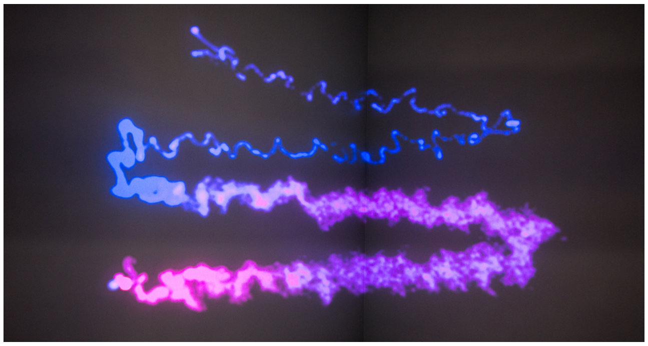

Figure 3: Variation of color and size via source–target modulation.

4.5 Direct Particle Interaction

So far, we discussed interaction techniques related to the emission and behavior of particles. However, an important aspect is to manipulate the existing artwork and directly interact with the fluid. To this end, we add three different ways of direct fluid interaction, which we discuss in the following: hand manipulation, blowing manipulation, and erasing.

4.5.1 Fluid Manipulation by Hand

Artists can manipulate the drawn fluid sketch by moving their non-dominant hand through the particle population, as illustrated in Fig. 4. This creates a spherical external influence (see Section 3.2), centered at the user's non-dominant hand.

Hand manipulation can be enabled and disabled via a menu. When enabled, users can push and drag the drawn fluid sketch in a natural and intuitive manner. The size of the hand interaction region is user-adjustable via a menu. Optional visual feedback of the interaction region is provided by a white solid sphere that matches the interaction region's position and extents.

A user-configurable strength factor is available in the menu. It determines the value of $f_{strength}$ in Equation 2. This allows the user to control the sensitivity, such that the hand interaction feels natural for the given viscosity of the fluid.

4.5.2 Fluid Manipulation by Blowing

Surveying techniques for real-world interaction with liquid colors, we found that blowing is regularly used for watercolor paintings to smear ink on the paper surface. Furthermore, turbulence in smoke, e.g., for smoke photography, can easily be created by blowing. Inspired by such techniques, we add a blowing interaction metaphor to Fluid Sketching, which is, to the best of our knowledge, a completely novel type of interaction for artworks in IVEs. An example of a blowing interaction is illustrated in Fig. 5.

We realize this interaction metaphor by applying a conical influence (see Equation 3.2), originating from the user's mouth position. The force strength is determined by the amplitude of recorded sound from the wireless microphone the user is wearing. For the realization, we use an adapted version of BlowClick [44]. The audio signature of blow events is recognized and compressed to a single strength value, which is exponentially smoothed over time. It is additionally multiplied with a user-configurable strength factor that allows for

Figure 4: User manipulates fluid sketch with her non-dominant hand.

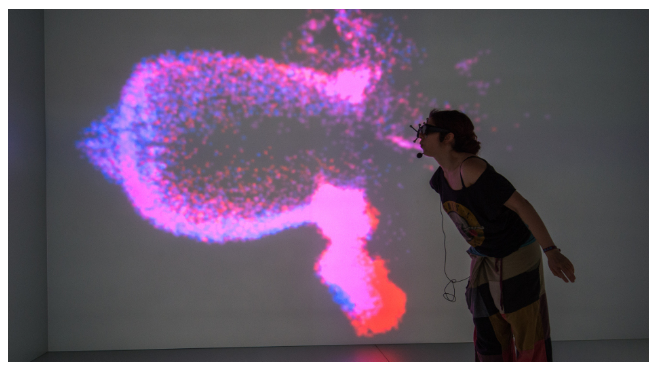

Figure 5: User blows into the sketch using the microphone.

more user control. The result is fed to the fluid simulation as the effective strength $f_{strength}$ of the conical force. The opening angle of the conical force is 60°, which showed to be a reasonable width in prior informal testing. The height of the conical interaction region h_{cone}—i.e., the blowing interaction range—is user-configurable via a menu to grant users control over the distance at which particles are affected.

4.5.3 Erasing Particles

When drawing on paper, an eraser is a vital tool to correct for mistakes. The same also accounts for our sketching system, which is why we add an eraser tool. The erase mode can be activated via the menu or a dedicated button on the Flystick2. The eraser is visualized as sphere centered at the non-dominant hand. Since the non-dominant hand is also used for shaping, the eraser sphere is colored in red for disambiguation. An example is illustrated in Fig. 6.

The system offers the user to change the eraser size via the menu. By moving the hand through space, the part of the sketch that lies

Figure 6: User erases parts of the fluid sketch.

within the sphere volume of the eraser is deleted. This is realized by marking the respective particles inactive.

4.6 Overall Scene

A challenge in creating real-world fluid artworks is that artists only have limited control over the dynamics of the diffusion process. We address this issue by providing a pause mode. When enabled, the diffusion process of the fluid gets temporarily disabled. Consequently, any particle movement stops and direct interaction with the sketch is deactivated. However, users can still draw or erase brush strokes, so even complex structures can be created without being distorted by diffusion over time. By deactivating the pause mode, the diffusion process continues for all particles. Pause mode is indicated to the users by changing the background color from light gray to dark gray.

While the pause mode grants users more control for working on details, it lacks the possibility of manipulating the existing sketch. Therefore, we additionally add the freeze mode. It also stops the diffusion process, but in contrast to the pause mode, sets the velocity of all existing particles to zero and marks them as frozen, which excludes them from the fluid simulation. Freezing only affects the fluid sketch that has already been drawn, but not newly emitted particles. To allow for local manipulations, frozen parts of the sketch can be woken up by interacting with them via hand or blowing. Additionally, particles that are affected by neighboring particles are woken up, too, which allows for the propagation of local modifications.

To save the drawn sketch for further processing, we add an export option in the OBJ file format. Thus, fluid sketches can be further edited in a 3D modeling application of the users' choice.

As illustrated above, Fluid Sketching offers a multitude of configuration capabilities. To allow users to easily restore already-used settings, we add the possibility to save and restore configurations as presets. Thus, artists are able to create their own toolset, which can grow over time and can be used to easily create more refined sketches in future drawing sessions.

4.7 Rendering

The particle visualization in the Fluid Sketching system is realized as billboard rendering with additive blending. We chose this rendering style, because it offers a reasonable trade-off between efficiency and beauty. It enables us to handle large enough particle populations for creating plausible fluid sketches while providing an appealing view.

5 User Study

To get first feedback on the overall concept of Fluid Sketching, we conducted a qualitative user study. It serves as a guideline for further refinement of the interface concept, improvement of the usability, and the implementation of additional features. To this end, we recruited two representative user groups. Professional digital artists, being the primary target audience of the system, formed one group, in order to get feedback from a practitioner's perspective. VR experts from the fields of VR research and development formed the other user group, to get feedback on interaction and implementation aspects. In the following, we first outline the procedure of the study, before successively presenting and discussing the results.

5.1 Procedure

We used the technical setup as described in Section 4.1. Participants were guided through the study by a supervisor, while an observer transcribed arising comments. The study consisted of five phases. These comprise filling out a pre-study questionnaire, getting an introduction, performing pre-defined tasks, free exploration, and filling out a post-study questionnaire.

The pre-study questionnaire captured demographic data and the participants' experience with VR technology. Furthermore, they were asked if any vision impairments were present. Additionally, a consent form was signed by the participants and they were informed about their right to quit the study at any time without justification.

Afterwards, the participants were equipped with the interaction devices, before entering the aixCAVE with the supervisor. Here, in a short instruction phase, the overall functionality and the interface of the system were explained. Additionally, the participants were instructed to think-aloud and comment freely on the Fluid Sketching system so that the observer could transcribe comments from outside the aixCAVE via microphones installed in the system.

In the next phase, the participants had to perform four sets of guided tasks. These tasks were related to the brush parameters (see Section 4.2), fluid parameters (see Section 4.3), source–target modulation (see Section 4.4), and direct particle interaction (see Section 4.5). Each task started with a systematic introduction to all parameters.

Afterwards, the free exploration phase started, during which the participants could try out the functionality of the system freely. In this phase, no further instructions were given, however the participants could ask questions and further comment on the experience.

Finally, the participants had to fill out a post-study questionnaire, comprising questions about feedback on the tasks and the opportunity to mention ideas for additional features. Moreover, the short version of the AttrakDiff 2 questionnaire [15] was handed out to obtain information on the user experience.

5.2 Results and Discussion

Overall, 10 subjects participated in the study, 4 of which were artists and 6 were VR experts. Among the participants, 2 were female and 8 were male, and all had normal or corrected-to-normal vision. Five participants were aged 25–34 years, four 35–44, and one 45–54. All artists stated that they had experience with VR systems before. Based on the transcripts, statements given by the participants were extracted. Similar statements were grouped to identify common feedback given by multiple participants.

5.2.1 Feedback on Existing Features

Regarding the configuration of the fluid parameters, three of the participants appreciated to directly see the effect of changing the fluid parameters. However, five participants stated that it is difficult to understand the parameters. One of them stated that "There are so many parameters, I can barely keep them in my mind". While we expect that users would be able to get known to all parameters when using the application for a longer time, at least on first contact, the high number of different parameters might actually be a considerable limitation. We plan to address this in different ways. First, a number of presets for various fluid types such as honey or smoke might remove pressure from the participants to directly consider all parameters from the beginning. In addition, tooltips in the parameter menus, giving a textual explanation of and a visual example for each parameter could assist users in understanding and learning the parameters.

Two participants stated that it is difficult to see how far particles are since there are no lighting or depth cues. However, one participant perceived no disturbing artifacts due to the rendering style and further mentioned that she likes the over-saturation and that it encourages to move around in the tracked space. In future design iterations, we will account for this by providing different rendering styles for particles.

Concerning direct particle interaction, the following statements arose. When the viscosity of the brush was high, three participants stated that manipulating the sketch with the hand feels like sculpting. A VR expert pointed out that it's possible to "smear" the sketch and that this allows for a lot of control for manipulation. The fact that artists compared the interaction technique to sculpting shows that it provides a direct and intuitive way of interaction. Furthermore, it was rated by a VR expert as giving a high degree of control. Overall we can conclude that it is a well-suited interaction type for Fluid Sketching.

Five participants stated that they do not need or like a sphere representation of the interaction range for hand interaction and that they had better depth perception without it. However, one artist found it comfortable to use hand interaction with the sphere visualization, stating that "It gives the illusion of having a precise tool". This indicates that making the sphere visibility configurable was the right choice, as it allows to account for different user preferences.

Regarding interaction with the blowing metaphor, nine participants stated that they liked it and two of them expressed that it "feels natural to use". Five participants mentioned that the interaction via blowing is fun. Based on these given statements, we can conclude that the feature is a valuable addition to the artists toolset enabling a natural and intuitive type of interaction. However, three VR experts noted that the blowing interaction was hard to control and one of them stated that "It often leads to destructive outcomes". We plan to address this issue by adding the blowing strength as a modulation target for the source–target modulation. Thus, finer control over the blowing strength while performing the interaction would be granted. Furthermore, six participants reported that speaking also triggers the blowing interaction. We plan to address this limitation by investigating a recently proposed addition to BlowClick [45], which distinguishes between blowing and speaking via a neural network.

For the eraser feature, three participants stated that the eraser sphere is occlusive, just as during hand interaction. However, one participant explicitly mentioned that the eraser sphere was not disturbing compared to the hand interaction mode. Two participants suggested to draw only the sphere's outline or to make it transparent in order to mitigate the occlusion, which we also deem to be a good way of improving the visualization of the eraser and sphere interaction ranges.

Regarding source–target modulation, a total of eight participants explicitly mentioned that they liked the feature. Based on these statements, we observed that the second-most appreciated feature is the source–target modulation, complementing the intuitive blowing metaphor by a means of performing precisely controlled expressive drawing gestures. However, five of the participants reported that it is complicated and they could not remember their own modulations. To address this issue, we plan to design a custom interface to quickly and comprehensibly define the modulations. Moreover, two participants stated that the source–target modulation is hard to control, although conversely, one participant commented "It gives me more control and I can really mix the effects, great!". We assume that this difference might be due to the degree of training since some of the users know the interaction devices that were used for the study.

Regarding the scene related features, the following statements emerged for freezing and pausing. Two participants positively mentioned unfreezing by directly interacting with the sketch via hand or blowing. Furthermore, two VR experts stated that they found pausing is a valuable feature that offers more control. This indicates that these features were valued and deemed useful for having precise control over the amount of diffusion.

Concerning the overall system performance, the following statements arose. One participant stated that she was confused when the system got slower and lost interest in using it. Another participant pointed out to notice the "burden on the graphics card". So far, we did not perform in-depth profiling of our application, but we plan to do so in future iterations. However, it should also be noted that the study was carried out on several-year old hardware, and we expect it to perform better on modern hardware.

Considering all given comments, we can conclude that the interaction and scene control features that were developed for the Fluid Sketching system serve their intended purpose well. The goal of providing an artist's interface that allows for natural and direct, but also precise and expressive manipulation of fluid sketches was successfully achieved.

5.2.2 Ideas for Additional Features

A goal of our user study apart from evaluating the current state was to gather ideas for improvements and additional features. Concerning overall scene related features the following suggestions were made. One artist requested a feature to import models to paint on. Two participants stated that an undo feature could be helpful for precise drawings. Two participants would like to have a preview mode to inspect emitted particles without altering the piece of art. Two participants reported that it would be nice to have different rendering options. One participant asked for a heads-up display with status information about the scene. One participant requested a feature to save individual parts of the sketch.

Regarding the brush, the following statements emerged. One participant reported that occlusion problems with the Flystick were disturbing for small structures. Moreover, he asked for more control on where the particles are emitted and suggested to include a precise indicator for the particle emission location. One participant suggested to offer different particle shapes besides the default billboard.

Several suggestions were made for additional color selection and modification features. One participant asked for a feature to change the colors of the drawn fluid sketch. She suggested two different ways of achieving this. First, to add a paint brush that changes the color of existing particles. Second, to offer hue and saturation controls to globally shift hue and change saturation to see the sketch in a different color range. One participant stated that there should be an easier way to quickly change emission colors, and suggested to add a button that switches to a new color.

Regarding the hand interaction, three participants mentioned that they would like to use both hands for shaping the sketch.

We will carefully consider these suggestions for inclusion in future design iterations. While most of the mentioned features can be realized with reasonable effort, adding an undo feature imposes several difficulties due to the dynamic nature of our fluid simulation system.

5.2.3 User Experience Questionnaire

The AttrakDiff questionnaire was used on a 7-point scale of semantic differentials with scores ranging from -3 to 3. Fluid Sketching is presented as self-oriented in the AttrakDiff portfolio with a mean of 0.43 for *pragmatic quality* and 1.50 for *hedonic quality*. The mean rating of *attractiveness* is positive with 1.95.

Based on these results, in terms of hedonic quality, Fluid Sketching is clearly above average which means it awakens curiosity, motivates, and stimulates the users. Nevertheless, the value of pragmatic quality is just above the average. Overall, the result indicates that there is room for improvement in terms of usability.

6 Conclusion and Future Work

We presented a novel medium for creating 3D fluid artwork in IVEs. It enables artists to draw fluid-like sketches using a 3D brushing tool and offers natural interaction methods for shaping the drawn sketches. Procedural curl-noise was chosen to animate the fluid, which gives a high degree of animator control while having low requirements on memory and computation time. The fluid advection is performed by a GPU-based particle system, into which spherical and conical external velocities were integrated to realize direct hand and blowing interaction. To enable artists to perform expressive brush strokes, the source–target modulation was included.

By performing a qualitative user study with VR experts and artists, we were able to demonstrate the usefulness of our approach. Overall very positive feedback was obtained, indicating that the goal of providing a novel sketching tool for the creation of 3D fluid artwork was successfully achieved.

For further improvements on the Fluid Sketching system, many useful and valuable suggestions were gathered and will be addressed in future work. Commonly requested features will be implemented and the usability of the system will be improved upon, alongside porting the application to consumer-level hardware platforms. This way, the system will become an appealing tool for digital artists and other user groups alike.

REFERENCES

[1] S. Abraham and D. Fussell. Smoke brush. In *Proc. of the Workshop on Non-Photorealistic Animation and Rendering*, pages 5–11. ACM, 2014.

[2] N. Y. F. Academy. How to photograph smoke. https://www.nyfa.edu/student-resources/how-to-photograph-smoke, 2016. Accessed: 2017-11-18.

[3] R. Acar and P. Boulanger. Digital marbling: a multiscale fluid model. *IEEE TVCG*, 12(4):600–614, 2006.

[4] B. T. Akgun. The digital art of marbled paper. *Leonardo*, 37(1):49–52, 2004.

[5] R. Bridson, J. Houriham, and M. Nordenstam. Curl-noise for procedural fluid flow. *ACM TOG*, 26(3):46, 2007.

[6] M. Chu and N. Thuerey. Data-driven synthesis of smoke flows with CNN-based feature descriptors. *ACM TOG*, 36(4):1–14, July 2017.

[7] K. Crane, I. Llamas, and S. Tariq. Real-time simulation and rendering of 3D fluids. In H. Nguyen, editor, *GPU Gems 3*, pages 633–675. Addison-Wesley, 2008.

[8] M. F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM TOCHI*, 2(3):220–238, 1995.

[9] M. Desbrun and M.-P. Gascuel. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96*, pages 61–76. Springer, 1996.

[10] A. G. Forbes, T. Höllerer, and G. Legrady. Generative fluid profiles for interactive media arts projects. In *Proc. of the Symp. on Computational Aesthetics*, pages 37–43. ACM, 2013.

[11] T. A. Galyean and J. F. Hughes. Sculpting: An interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, July 1991.

[12] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen. Extended pie menus for immersive virtual environments. *IEEE TVCG*, 19(4):644–651, 2013.

[13] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.

[14] S. T. Greenwood and D. H. House. Better with bubbles: enhancing the visual realism of simulated fluid. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer animation*, pages 287–296. Eurographics Association, 2004.

[15] M. Hassenzahl, S. Diefenbach, and A. Göritz. Needs, affect, and interactive products–facets of user experience. *Interacting with computers*, 22(5):353–362, 2010.

[16] M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner. SPH Fluids in Computer Graphics. In S. Lefebvre and M. Spagnuolo, editors, *Eurographics 2014 - State of the Art Reports*. The Eurographics Association, 2014.

[17] B. Jackson and D. F. Keefe. Lift-off: Using reference imagery and freehand sketching to create 3d models in vr. *IEEE TVCG*, 22(4):1442–1451, 2016.

[18] X. Jin, S. Chen, and X. Mao. Computer-generated marbling textures: a gpu-based design system. *IEEE CGA*, 27(2):78–84, 2007.

[19] R. W. Johnson. *Handbook of fluid dynamics*. Crc Press, 2016.

[20] D. Keefe, R. Zeleznik, and D. Laidlaw. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE TVCG*, 13(5):1067–1081, 2007.

[21] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola Jr. Cavepainting: a fully immersive 3D artistic medium and interactive experience. In *Proc. of the 2001 Symp. on Interactive 3D Graphics*, pages 85–93. ACM, 2001.

[22] T. Kim, N. Thürey, D. James, and M. Gross. Wavelet turbulence for fluid simulation. *ACM TOG*, 27(3):50:1–50:6, Aug. 2008.

[23] A. N. Kolmogorov. Wienersche spiralen und einige andere interessante kurven im hilbertschen raum. *Acad. Sci. URSS*, 26(2):115–118, 1940.

[24] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross. Data-driven fluid simulations using regression forests. *ACM TOG*, 34(6):199:1–199:9, 2015.

[25] L. Latta. Building a million particle system. In *Game Developers Conference*, 2004.

[26] S. Lu, X. Jin, H. Zhao, and Y. Zhao. Real-time image marbleization. *Multimedia Tools and Applications*, 64(3):795–808, 2013.

[27] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.

[28] B. B. Mandelbrot and J. W. Van Ness. Fractional brownian motions, fractional noises and applications. *SIAM review*, 10(4):422–437, 1968.

[29] D. V. Maurer-Mathison. *The ultimate marbling handbook: a guide to basic and advanced techniques for marbling paper and fabric*. Watson-Guptill, 1999.

[30] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics symp. on Computer animation*, pages 154–159. Eurographics Association, 2003.

[31] K. Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

[32] K. Perlin. Noise hardware. *Real-Time Shading SIGGRAPH Course Notes*, 2001.

[33] K. Perlin and F. Neyret. Flow noise. In *28th Int. Conf. on Computer Graphics and Interactive Techniques*, page 187. SIGGRAPH, 2001.

[34] D. Rausch and I. Assenmacher. A sketch-based interface for architectural modification in virtual environments. In *5. Workshop der GI-Fachgruppe VR/AR*, 2008.

[35] D. Rausch, I. Assenmacher, and T. Kuhlen. 3D sketch recognition for interaction in virtual environments. In K. Erleben, J. Bender, and M. Teschner, editors, *VRIPHYS*. The Eurographics Association, 2010.

[36] L. F. Richardson. Weather prediction by numerical process cambridge university press. *Cambridge Richardson Weather prediction by numerical process 1922*, 1922.

[37] E. Sachs, A. Roberts, and D. Stoops. 3-draw: A tool for designing 3d shapes. *IEEE Comput. Graph. Appl.*, 11(6):18–26, Nov. 1991.

[38] D. Schroeder, D. Coffey, and D. Keefe. Drawing with the flow: A sketch-based interface for illustrative visualization of 2D vector fields. In *Proc. of the Seventh Sketch-Based Interfaces and Modeling Symp.*, pages 49–56. Eurographics Association, 2010.

[39] J. Stam. Stable fluids. In *Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.

[40] J. Stam. Real-time fluid dynamics for games. In *Proc. of the Game Developer Conference*, volume 18, page 25, 2003.

[41] F. Wan, Y. Yin, Q. Zhang, and X. Peng. Analysis of parallel multigrid methods in real-time fluid simulation. *International Journal of Modeling, Simulation, and Scientific Computing*, page 1750042, 2017.

[42] G. Wesche and H.-P. Seidel. Freedrawer: a free-form sketching system on the responsive workbench. In *Proc. of the ACM VRST*, pages 167–174. ACM, 2001.

[43] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM TOG*, 24(3):965–972, July 2005.

[44] D. Zielasko, S. Freitag, D. Rausch, Y. C. Law, B. Weyers, and T. W. Kuhlen. Blowclick: A non-verbal vocal input metaphor for clicking. In *Proc. of the 3rd ACM SUI*, pages 20–23. ACM, 2015.

[45] D. Zielasko, N. Neha, B. Weyers, and T. W. Kuhlen. Blowclick 2.0: A trigger based on non-verbal vocal input. In *Proc. of IEEE VR*, pages 319–320. IEEE, 2017.