Phys-Sketch: Sketching 3D Dynamic Objects in Immersive Virtual Reality

Jose Abel Ticona[1(✉)], Steeven Villa[1], Rafael Torchelsen[2], Anderson Maciel[1], and Luciana Nedel[1]

[1] Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
{jatlarico,dsvsalazar,amaciel,nedel}@inf.ufrgs.br

[2] Universidade Federal de Pelotas (UFPel), Pelotas, Brazil
rafael.torchelsen@inf.ufpel.edu.br

http://inf.ufrgs.br/~jatlarico, http://inf.ufrgs.br/~dsvsalazar, http://rafaeltorchelsen.wordpress.com, http://inf.ufrgs.br/~amaciel, http://inf.ufrgs.br/~nedel

Abstract. Sketching was traditionally a 2D task. Even when the new generation of VR devices allowed to sketch in 3D, the drawn models remained essentially static representations. In this paper, we introduce a new physics-inspired sketching technique built on the top of Position-based Dynamics to enrich the 3D drawings with dynamic behaviors. A particle-based method allows interacting in real time with a wide range of materials including fluids, rigid bodies, soft bodies and clothes. Users can interact with the dynamic sketches and sculpt them while they move, deform and fall. We analyze the expressiveness of the system from the regard of two experienced artists. Thus, this paper also gives a starting point to move towards an improved generation of physics-enabled sketching applications.

Keywords: 3D sketching · Real-time physics-based simulation · Human-computer interaction · Immersive-environments

1 Introduction

Virtual and Augmented Realities (VR and AR) are in part tricking our senses to improve the feeling of presence. Several factors directly influence how successful the user experience is, such as the quality of the graphics and 3D models. However, this static part by itself does not produce a full VR or AR experience. Elements' behavior, as well as the way they move and how the user interacts with them, also play a fundamental role. Usually, looking for minimizing

This study was partly funded by the Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001, and partly by CNPq. We also acknowledge FAPERGS (project 17/2551-0001192-9) and CNPq-Brazil (project 311353/2017-7) for their financial support.

M. Gavrilova et al. (Eds.): CGI 2019, LNCS 11542, pp. 119–130, 2019.
https://doi.org/10.1007/978-3-030-22514-8_10

the computation-time, designers choose to not use physically-accurate objects. Therefore, adopting kinematic behaviors is quite popular. Having said that, for a true VR experience, one has to consider physical behaviors of objects made of diverse materials, not to mention the ways people interact with them.

Virtual modeling is a growing area due to the development of new VR, interaction, and visualization techniques. This area is promising in the sense that it can improve creativity and reduce the development time of virtual models, environments, and animations.

Some popular applications were recently proposed to allow 3D creations employing immersive interaction, such as the TiltBrush [3] released in 2016 by Google. This application addresses several immersive interaction issues when painting in 3D. In the same way, Quill [2], from Facebook, lets the user create and animate virtual models. These approaches were successful in allowing users on materializing ideas, since both introduced tools to sculpt, draw, and paint.

In spite of that, the more noticeable weakness on the mentioned tools lies in their kinematic nature, this being reflected on the impossibility of choosing a particular material for a given object. Some of the applications let the user associate materials to their models. Then again, this is limited to rendering properties. A lack of physics-based behavior is thus noticeable: drawings or objects created are static along all the experience. The interaction between bodies is entirely kinematic and awkwardly real. However, the literature offers well developed, stable and interactive methods to deal with objects dynamics. Position-based dynamics, for instance, significantly reduces the computation time of physically plausible simulations. Recent developments in this area let us simulate significant natural phenomena [13].

Considering the issues mentioned above, we propose a novel immersive sketching application to create elements with different materials. Our solution allows for physics-based interactions between objects in real-time using position-based dynamics (PBD). This model is stable and permits to create expressive dynamic-sketches involving several types of physical behaviors such as rigid solid bodies, liquids, gases, soft solids, and clothes. As an example, it is possible to change the flow of a river by adding or digging into the ground while the river is flowing, or even to create beautiful waterfalls using the same methods. Moreover, this approach also allows the user to see in real time the behavior of recently created soft bodies or even clothes, in a rich direct manipulation environment.

2 Related Works

Recently, some works have proposed the use of immersive methods to let artist sketch, draw and even sculpt. GravitySketch [1], is a recent application which gives an immersive experience and additionally let the user interact without controllers by using leap motion. Works as Canvox by Kim et al. [6] proposes the division of the whole canvas in smaller volumes of interest to give more details using octrees. Although This work notably improves the sketching stage, they neglect the after-sketching interaction. Likewise, Multiplanes by Barrera et al. [4] aid the user to sketch by automatically generating planes as the user draws a line. This strategy is a 3D immersive analogous of the conventional CAD sketching pipeline but, once again, the authors focused on the sketching omitting further physical interactions with the already drawn objects.

Recently, Seo et al. [15] presented Aura Garden in 2018, a collaborative sculpting environment for light. This environment lets users draw and animate in mid-air with different materials, but all those materials are non-physical (excluding wood), so it is not possible to simulate interactions neither among them or the user. Eroglu et al. [5] presented a successful physic-based model to sketch in VR. This work focused on fluid modeling, letting the user change the fluid properties and freely draw in space. Although they did an outstanding work, their method is limited to fluids, and do not take into account solids or soft bodies. Lately, in 2018, Claybook [14] was released for most of the video-game consoles, this game let users generate dynamic content made of clay. Objects phases are liquid or solid, but it doesn't allow users to create rigid bodies. Besides, it is not immersive at the date of publication of this manuscript. To sum up, a common failure of the above-commented studies is the few or even nonexistent physics-based animation on the process, limiting the dynamism of the creations. Thus, the interactivity of the objects decreases. To solve that, we put our efforts in developing a possible real-time physics-enabled environment to both, sketch and animate bodies.

3 System Overview

The proposed system is divided into two main modules: First of all, the *Sketching* module which is the creation stage, where the user draws objects. Users are allowed to select materials and sketch in any possible position using the available brush shapes. However, for the sake of controllability, objects are unprovided of physical properties during the sketching step. Additionally, for performance reasons bodies being sketched, are represented as a set of spheres (see Fig. 1 left)

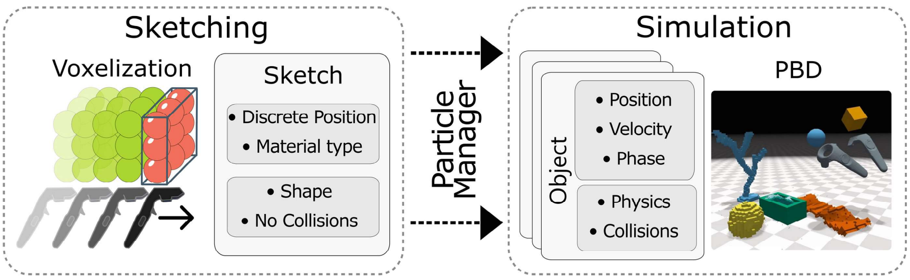

Fig. 1. The interaction technique in our design switches between two states: sketching and simulation. The first state is used to create new particle-elements into the scene, and the dynamic behavior is given to the bodies on the second stage depending on the properties of the material.

through a process called *Voxelization*. Section 4 explain in detail the processes involved in the sketching stage; Afterwards, in the *Simulation* stage, physics is assigned to the bodies (see Fig. 1 left). In this step, the animated bodies interact with both; user and environment. Moreover, users can touch, stretch or move any of them as desired. The system is built over the standard position-based dynamics (PBD) framework: We update both velocities and positions of the virtual tool using the external tracking system once the main PBD calculations were finished.

4 Sketching

Primarily, to create an object within our framework, the user must follow two steps: first sketch the body (*Sketching*); and next provide it with physical behavior (*Simulation*). Although both stages are well demarcated for the user all along the creative process, the simulation continues running on background for the existing bodies.

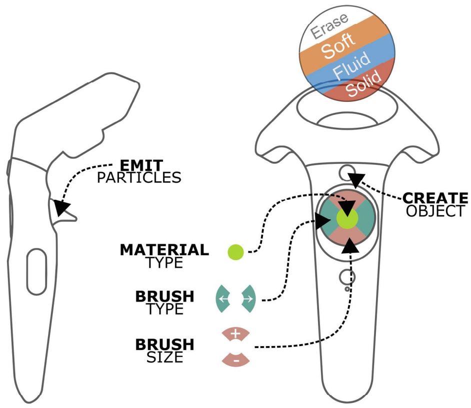

Fig. 2. Functionalities of the tool for interactions while sketching

Separation of stages is, thus, important to improve the sketching experience; without this discrimination, newly introduced particles would start falling after its creation, reducing usability to the system and making the creation of consistent sketches difficult. Therefore, in the beginning, the generated particles do not have any animation. Furthermore, the primary interaction tool is the brush, which has three functions: create, remove and move objects. Whats more, the user can easily select the object properties through the controller buttons as shown in Fig. 2(a). Additionally, the amount of particles introduced in the environment is given by the size and shape of the brush; new particles are evenly generated through voxelization (See Sect. 4.1).

Algorithm 1. Voxelization

procedure VOXELIZATION(P)
  $p = discretization(P)$ (Using Eq. 4)
  if $p \in M$ & $M[p] < 0$ **then**
    $P' = interpolation(p)$ (Using Eq. 5)
    $M[p] = index$
    $Particles[index] = P'$
    $index + +$

Therefore, to emit particles the user must press and hold the trigger. Similarly, the circular trackpad is used to navigate through brush shape and size options. Finally, pressing the center of the trackpad changes the material type (2). Furthermore, movement through the scene is allowed during all the simulation. However, there are physical (Room size) and computational (Simulation Size) constraints to the movement through the scene, even though the virtual space is not limited.

4.1 Voxelization

The manner as the particles are created plays a fundamental role in the simulation: placing more than one particle at the same position could generate overshoots at the first timesteps. In order to solve this issue, we use voxelization in the sketching stage; where each particle represents a voxel. The environment divided into a 3D grid $M \in R3$. Subsequently, M is initialized with -1 values. When a body is placed into the scene, the values of M are changed to 0 for the position of occupied by the thew body. This proceeding is shown in Algorithm 1 Finally, to remove particles, the Algorithm 2 is applied. We sync the coordinates of the space and the coordinates of our binary 3D grid using the following relations:

$$p' = p - c \tag{1}$$
$$Q' = Q - C \tag{2}$$
$$\frac{p'}{Q'} = \frac{|w|}{|W|} \tag{3}$$

Algorithm 2. Remove Particles

procedure PARTICLEREMOTION(P)
  $p = discretization(P)$ (Using Eq. 4)
  if $p \in M$ & $M[p] \geq 0$ **then**
    $indexRemove = M[p]$
    $M[p] = -1$
    $removeParticle(indexRemove)$

Where p is a positive integer of the 3D grid, p' the virtual coordinate system (VCS), translated by the center grid (c), and Q is the (Physic) World Coordinate System (WCS). Similarly, (C) is the center of the tracking system. $|w|$ and $|W|$ are the width of the 3D virtual grid and the physical workspace respectively. Consequently, this gives us the Correlation 3. Moreover, in order to translate world positions to the discrete space (VCS), we use the Eq. 4, while for bringing back the particles to the world space (WC), we use the Eq. 5.

$$p = (Q - C)\frac{|w|}{|W|} + c \tag{4}$$

$$Q = (p - c)\frac{|W|}{|w|} + C \tag{5}$$

4.2 Particle Interpolation

Brush movement tend to be fast during the simulation, it creates non-connected curves in the space. To avoid holes in the sketch, the brush shape is used as extrusion plane in each timestep. To do this, we calculate the velocity vector opposed to the controller movement. Secondly we find the traveled distance ($||V||\Delta t$), and next we calculate how many particles (i_{max}) fit in that distance (see Eq. 6).

$$i_{max} = Round(\frac{||V||\Delta t}{d_{particle}}) \tag{6}$$

Moreover, we divide this distance ($||V||\Delta t$) by the diameter of a particle ($d_{particle}$) to get the maximum number of particles (i_{max}) to project along the opposite velocity direction. Finally, in Eq. 7 we proceed to calculate the exact positions of the new particles.

$$P'(i) = P - i\frac{V}{||V||}d_{particle} \qquad \forall i \in [0, i_{max}] \tag{7}$$

At the end of this stage, we have position information, but particles are not dynamic so far. However, in the next section when we discuss particle dynamics through our particle management.

5 Simulation

Lagrangian models are widely used to perform real-time simulations. Methodologies as Position-based dynamics (PBD) [10] or Smoothed-particle hydrodynamics became more and more popular lately. Likewise, recent publication model physical behaviors as solids, fluids [8], gases [12] and even complex phenomena as phase transitions [13] PBD as starting point. Lagrangian models in contrast with Eulerian approaches, treat the bodies as the origin of its calculations. Consequently, it is unnecessary to calculate properties along a grid. This is useful especially when the domain of the simulation is unknown. There are several meshless methods used to model continuum mechanics; we chose Position-Based Dynamics [10] by reason of its unconditional stability and its low computational cost.

PBD is a particle-based animation technique that uses a set of constraints to calculate the positions of the particles in each timestep. Namely, it tries to fit the position of each particle based on a set of constraints. A given particle can admit an arbitrary number of constraints, but a constraint must comprise at least two particles. Furthermore, the solver must iterate to set the positions of the particles based on the constraints, reducing the value of each constraint. Thus, we can look at this process as an optimization problem. This is a successful way to simulate a wide variety of bodies such as clothes, deformable, rods, and elastics. However, it is not adequate for a rigid body or fluid simulations. Hence, techniques as Smoothed-particle hydrodynamic (SPH) are used to resolve such weakness. Indeed in 2013 the SPH-PBD integration was named Position-based fluids (PBF) [8] and made it possible to simulate fluids in the same frameworks as PBD. Finally, to simulate rigid bodies, we used shape matching technique [9].

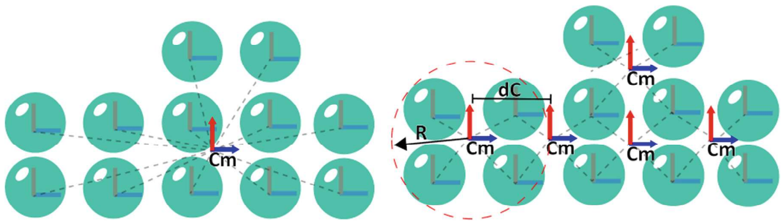

Fig. 3. (Left) Solid body: the coordinates of the particles are computed based on the mass center (Cm) of the body. (Right) Soft bodies: several coordinate systems are created based on two properties: the radius r and the distance dC between clusters, each cluster has a Mass center Cm, allowing articulated behavior.

5.1 Particle Management

We simulate a contrasting set of materials in this work; as a consequence, the procedure to simulate each material differs. In the following lines, we describe in detail the management of the particles/bodies, based on their materials.

Solids. Solid bodies are generated, firstly, calculating the center of mass of the entire body by summing up all the particle positions and later dividing by the total number of particles in the body. The transformations to a rigid body are directly applied to its mass center, and finally, applied to each particle using the relative positions (Fig. 3 left). This is know as Shape Matching [11]

Soft Bodies. In contrast, Soft bodies contain more mass centers. In this case, several CMs are calculated from a cluster of particles given a radius R and a minimum distance between clusters dC. This results in a structure of articulated bodies contained inside a total body. Similarly, as in solid bodies, every particle belonging to a cluster has a relative position to the CM of the cluster [9]. Figure 3(Right) shows how the coordinate system is created. A brief look to Fig. 3(left) and (right) shows the main differences between structures.

Liquids. We simulate fluids using Smoothed-particle hydrodynamics technique (SPH). Further information could be found in Macklin and Müller [8].

Collision Handling. We based the collision detection on the *Flex* Model (by Nvidia). Consequently, our system detects only particle-particle but not particle-mesh collision. Such limitation forces us to model everything in the scene with particles, including the tools (controllers). Hence, we modify the radius of solid particles constrained to $R_{Solid} < R_{Fluid}$ in order to handle solid-fluid collisions, this prevent possible leaks of fluid particles through spaces between solid particles. Controller-objects collisions are managed common particle interactions. However, the dynamic of those particles is given by the external tracking and not by the simulation dynamics. The collision system remains stable under normal conditions, but there is a maximum velocity where the particles would pass through an object because of the simulation timestep, due to this value is considerably high, it is not an issue.

6 Rendering

The rendering is divided into two parts: Fluids and Rigid-Soft bodies. Fluid rendering was done using Anisotropic Kernels because this is the standard system used by Flex. This approach render the objects based on the neighboring particles, namely performing Principal Component Analysis (PCA) over the neighbors. More information about this method could be found in the Yu & Turk paper published in 2013 [16]. For solid and soft rendering applied marching cubes (MC) algorithm [7] to the model voxelization to arrange particles through space, is a straightforward way to get a conceptual visualization of the objects. During the simulation, we know from the neighborhood search, which particles are on the surface of the object, so we apply MC only on outer particles, reducing computational cost.

7 Results

The sketching system was implemented and tested in a Dell workstation with an i7 3.2 Ghz Processor, 16 Gb Ram, and NVIDIA GeForce GTX 1070 Graphic Card, running Windows 10 and CUDA 9.2. All images and videos used in this paper were generated using HTC Vive headset and controllers for visualization and interaction (see Fig. 4). However, the sketching system is also compatible (and indeed was tested) with Oculus Rift headset and touch controllers. We also successfully tested Leap Motion for interaction.

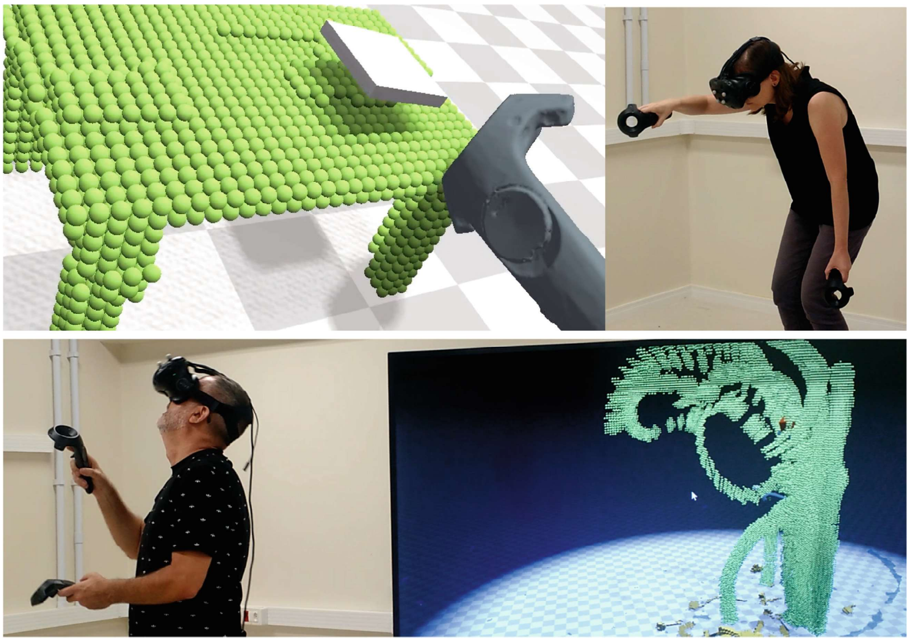

Fig. 4. Top: one of the invited artists is refining her sketch by removing particles. Bottom: sketching process of a three

Regarding performance, we got average timesteps between 15 ms and 33 ms; the late was reached when the amount of particles raises over 100k. In this case, the rendering is not fluid anymore. The system represents the workspace in a grid of $151 \times 151 \times 151$, where each cell potentially corresponds to one particle. As occupied cells are saved in memory and performance is mandatory, we restricted the max number of particles to 20% of the grid, or 688,590 particles. To run the scene of the Fig. 5, a total of 80k particles with a 28 ms for frame was used. There were 8 iterations for the run simulation and with a timestep of 16.6 ms.

7.1 Expressiveness

We invited two artists to informally test the immersive sketching application to create some sketches using the tools presented in Sect. 4. Both had no previous experience with VR. The test was divided into three stages and artists were free to spend as many time as they wanted in each stage:

1. Training stage: The system was introduced to the participants, letting them become familiar with the controls, and the virtual tool for 3D sketching in virtual environments. So, they first need to choose a material and a shape; Here the brush was introduced to them. We showed them how to change between shapes, sizes, and materials, to conclude we requested them to do it themselves drawing a free sketch.

2. Drawing stage: The artists were asked to draw a table and a container to be filled with water. Then, users were requested to interact with the objects they previously created.
3. Expression stage: The rtists were asked to sketch something original. For this stage, the gravity was set to zero. The sketched objects still reacts according to the materials they are made, but can freely float in the air, what can be seen as an interesting feature for artistics expression.

The experiments took approximately 40 min, where 15 min were spend for training and 25 min for executing the given tasks. Precisely, the first artist spent 40 min on the application while the second one spent 35 min. None of them reported any sympthon of cyberseekness, and both were very comfortable in the virtual environment.

The artists executed a lot of gestures with their arms, also walking and jumping in the real world. During all the process, they changed their positions – walking and crouching – looking for a better point of view to continue their sketches. An interesting behavior we perceived in both was that they moved in such a way to avoid to collide or even to cross the virtual objects. From observing this behavior and as the result of an informal post-test interview, we can conclude they experienced a high sense of presence in the virtual environment.

Subjects also highlighted the comfort of working with physical objects referring to the soft bodies and water behavior. A drawback reported by one artist was the lack of a strong haptic feedback. Currently, our model only conveys vibrotactile feedback by means of the VIVE controllers when the controller strikes an object in the scene. In the expression stage, the subjects reported an improvement in the experience due to zero gravity. Even though this behavior is non-physical it let them draw objects without building supports.

Another comment of the users is that the best strategy to sketch is using the dominant hand to create particles and the other hand to do different actions, like erasing, for instance. As a suggestion, they highlighted the potential of mixing different materials into a single object, for example, to allows the creation of a solid object with a soft part. They also mentioned that the behavior of the objects is credible, but the rendering of solid and soft objects must be improved. Finally, they also stated that the choice of the color and texture is relevant in the artistic process. However, despite these suggestions for future improvements, they were very excited to keep using the sketching system to create new scenarios and artistic installations.

8 Conclusions and Future Work

In this paper we proposed and prototyped a particle-based system to sketch and simulate virtual objects with physical behavior in a VR environment. Results shown the performance achieved is sufficient to support 3D interaction and fluid animation of the objects. An example is shown step by step in the sequence of Fig. 5.

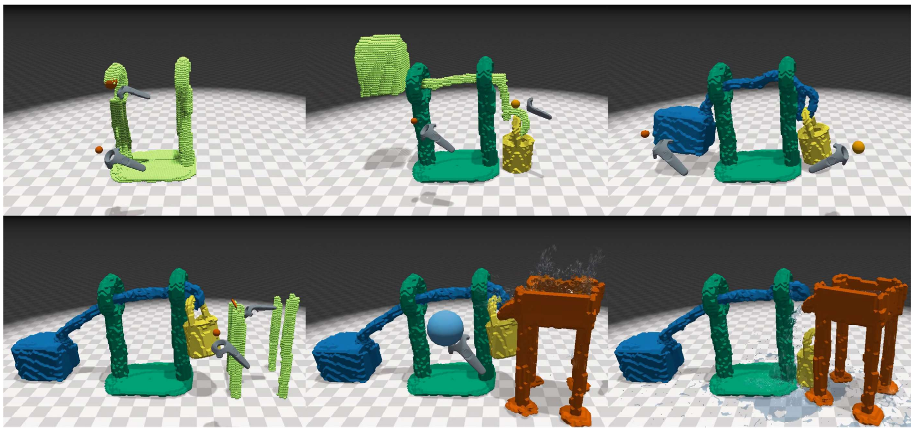

Fig. 5. In these sequence, we show how our tool allows us to use physics during simulation. This figure shows a pail, a bucket and a rope that joins the pail and the bucket, during the simulation we see how the bucket that has more weight lifts the pail, then we create a container of water that will fill the pail and then match the weights.

The first informal experiments with target users has shown promising results. The users, even if they did not have any previous experience with VR, felt comfortable to move and interact in the virtual environment proposed. They also did not report any cybersickness symptom during the 40 min each one spent fully immersed. Regarding the scenes sketched until now, they are interesting and sufficient to demonstrate the expressive power of the sketching system proposed. However, even if the main concepts about immersive sketching of dynamic objects have been verified, we are aware that the system needs to be improved to be regularly used for artists.

Notice that the rendering of the objects has a "legolized" appearance. An improvement of the render by smoothing the shapes would, we suppose, significantly improve the user experience. The marching cubes method generates smoother forms when the voxels are smaller. However, smaller voxels will demand more precision from the artists to sketch, as well as more computer power to simulate the objects behavior.

Path-constrained sketches are highly limited in terms of accuracy since the straight lines and geometries, in general, depend on the user ability to draw. Next works must move ahead introducing other tools, including the possibility of defining primitive bodies as cubes, spheres, rectangles, stars, etc. Additionally, the possibility of creating bodies using extrusions and revolutions would expand the range of applications of our tool. Further sketching tools as resizing or mixing up materials must be explored and evaluated. However, this work illustrates how physics-based interactions reinforce the sketching experience and provides tools to create *living sketches*.

A fully immersive experience should include tactile and force feedback. Besides, this could improve the performance of digital artists when sculpting or modeling their artwork, as drawing in the air lacks a support for accuracy. This is a possible future work.

References

1. Gravity Sketch - Bringing Virtual Reality Into Your Design Workflow. https://www.gravitysketch.com/
2. Quill VR illustration and animation tool built to empower artists and creators. https://quill.fb.com/
3. Tilt Brush by Google. https://www.tiltbrush.com/
4. Barrera Machuca, M.D., Asente, P., Lu, J., Kim, B., Stuerzlinger, W.: Multiplanes: assisted freehand VR drawing. In: Adjunct Publication of the Annual ACM Symposium on User Interface Software and Technology (UIST 2017 Adjunct), pp. 1–3 (2017). https://doi.org/10.1039/b924500f
5. Eroglu, S., Gebhardt, S., Schmitz, P., Hausch, D., Kuhlen, T.W.: Fluid sketching immersive sketching based on fluid flow. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 475–482. IEEE (2018)
6. Kim, Y., Kim, B., Kim, J., Kim, Y.J.: CanvoX: High-resolution VR Painting in Large Volumetric Canvas (2017)
7. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. In: ACM Siggraph Computer Graphics, vol. 21, pp. 163–169. ACM (1987)
8. Macklin, M., Müller, M.: Position based fluids. ACM Trans. Graph. **32**(4), 1 (2013). https://doi.org/10.1145/2461912.2461984
9. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. ACM Trans. Graph. **33**(4), 1–12 (2014). https://doi.org/10.1145/2601097.2601152
10. Muller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. J. Vis. Commun. Image Representation **18**(2), 109–118 (2007). https://doi.org/10.1016/j.jvcir.2007.01.005
11. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. In: ACM Transactions on Graphics (TOG), vol. 24, pp. 471–478. ACM (2005)
12. Ren, B., Yan, X., Yang, T., Li, C.F., Lin, M.C., Hu, S.M.: Fast SPH simulation for gaseous fluids. Vis. Comput. **32**(4), 523–534 (2016). https://doi.org/10.1007/s00371-015-1086-y
13. Salazar, S.V., Ticona, J.A., Torchelsen, R., Nedel, L., Maciel, A.: Heat-based bidirectional phase shifting simulation using position-based dynamics. Comput. Graph. **76**, 107–116 (2018). https://doi.org/10.1016/j.cag.2018.09.004
14. Sebastian, A., Sami, S.: GPU-based clay simulation and ray-tracing tech in Claybook. https://www.claybookgame.com/, www.secondorder.com/
15. Seo, J.H., Bruner, M., Ayres, N.: Aura garden: collective and collaborative aesthetics of light sculpting in virtual reality. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pp. Art 12:1–Art12:6 (2018). https://doi.org/10.1145/3170427.3177761
16. Yu, J., Turk, G.: Reconstructing surfaces of particle-based fluids using anisotropic kernels. ACM Trans. Graph. (TOG) **32**(1), 5 (2013)