VR-Doh: Hands-on 3D Modeling in Virtual Reality Supplemental Document

ZHAOFENG LUO*, Carnegie Mellon University, USA and Peking University, China
ZHITONG CUI*, Carnegie Mellon University, USA and Zhejiang University, China
SHIJIAN LUO, Zhejiang University, China
MENGYU CHU, Peking University, China
MINCHEN LI, Carnegie Mellon University, USA

CCS Concepts: • **Computing methodologies** → **Virtual reality**; *Point-based models.*

Additional Key Words and Phrases: Virtual Reality, Human-Computer Interactions, 3D Modeling, Elastoplasticity Simulaton, Material Point Method

Contents

Contents 1
1 Technical Background 1
1.1 MLS-MPM 1
1.2 PhysGaussian 3
2 Technical Details of MLS-MPM 3
2.1 First Piola-Kirchoff Stress 3
2.2 Plasticity 4
3 User Study 1: Usability and Effectiveness of VR-Doh 5
3.1 Semi-Structured Interview Results (Experts, P1-P6) 5
3.2 Semi-Structured Interview Results (Novices, P7-P12) 7
4 User Study 2: Comparison with Blender 7
4.1 Semi-Structured Interview Results 8
References 8

1 TECHNICAL BACKGROUND

Based on our design rationale, we selected PhysGaussian [Xie et al. 2024] as the foundation of our system to enable real-time simulation of elastoplastic objects and achieve photorealistic rendering. This section provides a technical background on the Moving Least Squares (MLS) Material Point Method (MPM) [Hu et al. 2018] and 3D Gaussian Splatting [Kerbl et al. 2023], which PhysGaussian builds upon.

1.1 MLS-MPM

The Moving Least Squares Material Point Method (MLS-MPM) [Hu et al. 2018] is a hybrid Lagrangian-Eulerian approach well suited for simulating multi-material phenomena. In this framework, Lagrangian particles track

*Both authors contributed equally to this research.

Authors' Contact Information: Zhaofeng Luo, Carnegie Mellon University, USA and Peking University, China, roushelfy@stu.pku.edu.cn; Zhitong Cui, Carnegie Mellon University, USA and Zhejiang University, China, zhitongcui@zju.edu.cn; Shijian Luo, Zhejiang University, China, sjluo@zju.edu.cn; Mengyu Chu, Peking University, China, mchu@pku.edu.cn; Minchen Li, Carnegie Mellon University, USA, minchernl@gmail.com.

Algorithm 1 Sim_substep()

```
Particle_to_Grid()
Update_Grid_Velocity()
Grid_to_Particle()
Particle_Projection()                ▷ Main Ppaer
Apply_Plasticity()
```

the geometry and material properties of the simulated object, while a Eulerian background grid facilitates force computation and time integration. This dual representation enables MPM to efficiently handle large deformations, topology changes, and contact by transferring physical quantities, such as mass and momentum, between particles and the grid, leveraging the strengths of each spatial discretization [Jiang et al. 2016; Sulsky et al. 1995].

Each MPM simulation time step (Algorithm 1) begins with the `Particle_to_Grid` operation, transferring particle mass and momentum to neighboring grid nodes:

$$\begin{aligned} m_i^n &= \sum_p w_{i,p}^n m_p, \\ m_i^n \mathbf{v}_i^n &= \sum_p w_{i,p}^n \left(m_p \mathbf{v}_p + (m_p \mathbf{C}_p^n - \mathbf{E}_p^n)(\mathbf{x}_i - \mathbf{x}_p^n)\right). \end{aligned} \tag{1}$$

Here, subscripts p and i correspond to particle and grid quantities, respectively, while the superscript indicates the time step. m is the mass, and w is the weight for the transfer, which is nonzero only for particles and grid nodes that are close. $\mathbf{v}$ is the velocity, $\mathbf{x}$ is the position, $\mathbf{C}_p$ stores information about the local velocity field around particle p, and $\mathbf{E}_p^n = -\frac{4\Delta t}{\Delta x^2} \sum_p V_p^0 \mathbf{P}_p^n \left(\mathbf{F}_p^n\right)^T$ is the elasticity stress term. Here, Δt is the time step size, Δx is the grid spacing, V_p^0 is the initial volume of particle p, $\mathbf{F}$ is the deformation gradient, and $\mathbf{P}$ is the first Piola-Kirchhoff stress calculated using $\mathbf{F}$.

In the `Update_Grid_Velocity` step, grid velocities are updated to incorporate external forces $\mathbf{f}_{ext}$, such as gravity and air damping:

$$\hat{\mathbf{v}}_i^n = \mathbf{v}_i^n + \frac{1}{m_i^n} \mathbf{f}_{ext,i}^n \cdot \Delta t. \tag{2}$$

To handle simulation domain boundaries and forces from passive objects, such as tracked human hands, *slip* or *sticky* boundary conditions are enforced. For slip boundaries, the normal component of the relative velocity near the boundary is set to zero, while for sticky boundaries, both tangential and normal components are set to zero to simulate friction.

Following this, the `Grid_to_Particle` operation updates particle states based on nearby grid nodes:

$$\begin{aligned} \mathbf{v}_p^{n+1} &= \sum_p w_{i,p}^n \hat{\mathbf{v}}_i^n, \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \hat{\mathbf{v}}_i^n, \\ \mathbf{C}_p^{n+1} &= \frac{\Delta t}{\Delta x^2} \sum_p w_{i,p}^n \hat{\mathbf{v}}_i^n \left(\mathbf{x}_i - \mathbf{x}_p^n\right)^T, \quad \mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t \mathbf{C}_p^n\right) \mathbf{F}_p^n. \end{aligned} \tag{3}$$

Since boundary conditions are enforced at the grid level, particles may still penetrate solid boundaries, especially for fine geometries not adequately resolved by the grid. To address this, our system introduces a particle-level collision handling via an additional `Particle_Projection` step, ensuring more accurate handling of particle-boundary interactions (details in the main paper).

Finally, the `Apply_Plasticity` step updates the deformation gradient $\mathbf{F}_p^{n+1}$ through the return mapping function $Z(\cdot)$, which projects stresses outside the elastic zone back onto the yield surface of elastoplastic materials. The resulting change in $\mathbf{F}_p^{n+1}$ represents permanent plastic deformations that are not recovered by elastic forces. Further details are provided in the supplemental document.

1.2 PhysGaussian

3D Gaussian Splatting (GS) [Kerbl et al. 2023] employs a set of unstructured 3D Gaussian kernels to efficiently represent and render a scene. Each Gaussian kernel is characterized by its center $\mathbf{x}_k$, covariance matrix $\mathbf{A}_k$, and density function:

$$G_k(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T \mathbf{A}_k^{-1}(\mathbf{x}-\mathbf{x}_k)}, \tag{4}$$

opacity σ_k, and spherical harmonic coefficients C_k. Unlike neural radiation fields (NeRFs) [Mildenhall et al. 2021], which represent scenes using neural implicit functions and render views by casting camera rays, GS directly projects 3D Gaussians onto a 2D image plane, enabling highly efficient rendering and training. Additionally, the explicit representation of 3D GS facilitates convenient scene editing, as demonstrated in works such as Chen et al. [2024]; Gao et al. [2024].

During rendering, the final color C of each pixel is computed as a weighted sum of the projected Gaussian kernels' colors:

$$C = \sum_{k \in P} \alpha_k SH(\mathbf{d}_k; C_k) \prod_{j=1}^{k-1} (1 - \alpha_j), \tag{5}$$

where P is the set of all Gaussians contributing to the pixel color, ordered by view depth; α_k is the effective opacity, calculated as the product of σ_k and the density projected onto the pixel; SH represents the spherical harmonic function; and $\mathbf{d}_k$ is the view direction.

While GS primarily focuses on visual appearance, it does not incorporate physical properties. PhysGaussian [Xie et al. 2024] extends GS by integrating MPM, treating each Gaussian kernel as a Lagrangian particle to track displacement and deformation. This enables the simulation of elastoplastic behaviors under external forces or boundary conditions, creating a unified simulation-rendering framework. In PhysGaussian, at time t, the density of a deformed Gaussian k is calculated as:

$$G_k(\mathbf{x}, t) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_k(t))^T (\mathbf{F}_k(t)\mathbf{A}_k\mathbf{F}_k(t)^T)^{-1}(\mathbf{x}-\mathbf{x}_k(t))}, \tag{6}$$

where $\mathbf{x}_k(t)$ and $\mathbf{F}_k(t)$ are the current position and deformation gradient, respectively, of the corresponding MPM particle, provided by the simulation.

We observed that properly simulating the deformation and dynamics of elastoplastic materials for geometric modeling often requires significantly fewer degrees of freedom (DOFs) than rendering their complex appearance. To address this, we customized the framework by decoupling the appearance and physical representations, allowing fewer MPM particles than Gaussian kernels, thereby enabling real-time simulation. See our main paper for more details.

2 TECHNICAL DETAILS OF MLS-MPM

2.1 First Piola-Kirchoff Stress

To calculate the first Piola-Kirchoff stress $\mathbf{P}_p^n$, two forms are available: StVK and neo-Hookean, defined respectively as:

$$\mathbf{P}_p^n = 2\mu \left(\mathbf{F}_p^n - \mathbf{U}\mathbf{V}\right) + \lambda \left(J_p^n - 1\right) J_p^n \left(\mathbf{F}_p^n\right)^{-T}, \tag{7}$$

$$\mathbf{P}_p^n = \mu \left(\mathbf{F}_p^n \cdot \mathbf{F}_p^{nT}\right) + \mathbf{I} \cdot (\lambda \log(J) - \mu). \tag{8}$$

Here, μ and λ are the Lamé constants, representing the shear stiffness and incompressibility of the material, and $J = \det(\mathbf{F})$, measuring local volume change. The terms $\mathbf{V}$ and $\mathbf{U}$ are obtained through singular value decomposition of the deformation gradient tensor: $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}$.

2.2 Plasticity

Three types of plasticity models are often applied: Drucker-Prager, von Mises, and clamp-based plasticity. Here, we first provide the return mapping functions $Z(\cdot)$ of each plasticity model, and then explain all the symbols in the equation.

For Drucker-Prager plasticity:

$$Z(\mathbf{F}_p)_{druncker} = \begin{cases} \mathbf{F}_p, & \delta\gamma \leq 0 \\ \mathbf{U}\exp\left(\boldsymbol{\epsilon} - \delta\gamma \frac{\hat{\boldsymbol{\epsilon}}}{\|\hat{\boldsymbol{\epsilon}}\|}\right)\mathbf{V}^T, & \text{otherwise} \end{cases} \tag{9}$$

with

$$\delta\gamma = \begin{cases} \|\hat{\boldsymbol{\epsilon}}\|, & \operatorname{tr}(\boldsymbol{\epsilon}) > 0 \\ \|\hat{\boldsymbol{\epsilon}}\| + \alpha \frac{d\lambda + 2\mu}{2\mu}\operatorname{tr}(\boldsymbol{\epsilon}), & \text{otherwise} \end{cases} \tag{10}$$

For Von Mises plasticity:

$$Z(\mathbf{F}_p)_{von} = \begin{cases} \mathbf{F}_p, & \delta\gamma \leq 0 \\ \mathbf{U}\exp\left(\boldsymbol{\epsilon} - \delta\gamma \frac{\hat{\boldsymbol{\epsilon}}}{\|\hat{\boldsymbol{\epsilon}}\|}\right)\mathbf{V}^T, & \text{otherwise} \end{cases} \tag{11}$$

where

$$\delta\gamma = \|\hat{\boldsymbol{\epsilon}}\| - \frac{\tau_Y}{2\mu} \tag{12}$$

Clamp-based plasticity is defined as:

$$Z(\mathbf{F}_p)_{clamp} = \mathbf{U} \cdot \operatorname{Clamp}(\Sigma, \Sigma_{\min}, \Sigma_{\max}) \cdot \mathbf{V}^T. \tag{13}$$

Explanation of Symbols.

- $\mathbf{F}_p$: Plastic deformation gradient tensor, representing the irreversible plastic component of the deformation.
- $Z(\mathbf{F}_p)$: Updated plastic deformation gradient after applying the plasticity return mapping. Subscripts (e.g., "druncker", "von", or "clamp") specify the model being applied (Drucker-Prager, Von Mises, or Clamp-based).
- $\mathbf{U}, \Sigma, \mathbf{V}$: Components from the Singular Value Decomposition (SVD) of the deformation gradient $\mathbf{F}_p = \mathbf{U}\Sigma\mathbf{V}^T$, where:
  - $\mathbf{U}$: Left singular vectors (orthogonal matrix).
  - Σ: Diagonal matrix of singular values, representing principal stretches.
  - $\mathbf{V}^T$: Transposed right singular vectors (orthogonal matrix).
- $\boldsymbol{\epsilon} = \log(\Sigma^{\text{tr}})$: Hencky strain tensor, computed as the logarithm of the trial singular values Σ^{tr}.
- $\hat{\boldsymbol{\epsilon}}$: Deviatoric part of the Hencky strain tensor, removing the volumetric component:

  $$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\epsilon})\mathbf{I},$$

  where $\mathbf{I}$ is the identity matrix.
- $\|\hat{\boldsymbol{\epsilon}}\|$: Frobenius norm of the deviatoric Hencky strain tensor, representing its magnitude.
- $\delta\gamma$: Plastic multiplier, indicating the magnitude of plastic flow during the return mapping.
- $\operatorname{tr}(\boldsymbol{\epsilon})$: Trace of the Hencky strain tensor, representing the volumetric strain.
- α: Material parameter in the Drucker-Prager model, related to the friction angle and material properties.
- μ: Shear modulus, a material constant characterizing resistance to shear deformation.

- λ: First Lamé parameter, characterizing resistance to volumetric deformation.
- τ_Y: Yield stress in shear, a material parameter for Von Mises plasticity.
- $d\lambda$: Material-dependent parameter, often related to elastic properties such as the bulk modulus.
- $\Sigma_{\min}, \Sigma_{\max}$: Minimum and maximum limits for singular values in Clamp-based plasticity.
- $\text{Clamp}(\Sigma, \Sigma_{\min}, \Sigma_{\max})$: Function that clamps the singular values of Σ to lie within the range $[\Sigma_{\min}, \Sigma_{\max}]$.

3 USER STUDY 1: USABILITY AND EFFECTIVENESS OF VR-DOH

We present detailed subjective feedback from both experienced and novice participants below.

3.1 Semi-Structured Interview Results (Experts, P1-P6)

(1) Do you think the designs created using this tool align with your initial expectations?

P1 found the design mostly satisfactory, with minor areas for improvement. P2 stated that 80% of their expectations were fulfilled, with the pose being largely accurate, though fractures on the model surface hindered achieving a fully realized design. P3 expressed satisfaction with the details, particularly in the eyes of the SpongeBob model, and found the material texture consistent with the intended character image. P4 estimated that 70% of their expectations were met, as the realistic deformations matched their vision, but the finer details lacked completeness. Similarly, P5 found the results to align with their expectations despite minor fractures in the model. P6 noted that the outcome met basic expectations, though the initial design was not overly complex, and significant deformations led to fractures in the object.

(2) During your experience with the tool, what aspects were the most satisfying and the least satisfying? Additionally, what do you consider to be the greatest difficulty encountered while using the tool?

Most Satisfying: P1 highlighted the simplicity and enjoyment the tool brought to tasks, such as using a stick to create holes while designing strawberries, making the process feel immersive. The pinch gesture interaction using both hands was also found to be very practical. P2 appreciated the tool's ease of use without requiring extensive expertise, along with its high efficiency. The realistic deformation process, especially while editing the mermaid model, provided a visually engaging experience, akin to real swimming. Additionally, the ability to switch material parameters during editing was highly valued. P3 praised the intuitive interaction with 3D models, such as touching and squeezing, and also found the ability to adjust material parameters during editing particularly useful. P4 noted the tool's capability to provide real-time feedback on deformation with realistic rendering, which is often time-consuming in conventional modeling software. P5 found the tool effective for naturally and easily altering 3D model poses, with good continuity between joints, and appreciated the ease of adjusting movements and outfits to make designs more vivid. P6 valued the ability to merge objects seamlessly and highlighted the pinch gesture for shape editing without damaging surface details as especially useful.

Least Satisfying: P1 noted that the limited number of MPM simulation grids caused the texture of the strawberry object, which initially had a perforated texture, to become less apparent after merging. P2 expressed concern about potentially damaging the object too significantly during operations, which often led to surface fractures. P3 highlighted the absence of an undo function as a major limitation. P4 pointed out that the pinch gesture lacked proper visualization of the applied force, as well as recommended values for different materials and a clear indication of its maximum area of influence. P5 desired more detailed designs but found that insufficient particle counts made surfaces prone to fracturing. P6 also mentioned the lack of an undo function as a key drawback.

Greatest Difficulty: P1 found that the lack of tactile feedback made it challenging to adapt to deforming objects in a virtual reality environment, requiring some time to adapt to the new operational logic. P2 initially struggled with controlling the size of the selected area for the pinch gesture, leading to unsatisfactory results, but improved after multiple attempts. P3 identified difficulties in precisely adjusting the relative position of two merged objects using hands, as well as the issue of 3D Gaussian-based model surfaces easily fracturing. P4 noted that the pinch gesture lacked intuitive force control, making it hard to determine the appropriate force for different materials and avoid unexpected deformations. Additionally, 3D Gaussian-represented model surfaces were prone to breaking, with no automated surface repair function, and manual repairs proved difficult. P5 found it challenging to create symmetrical, streamlined shapes using hands. P6 highlighted the difficulty of precisely adjusting the relative position of two merged objects manually.

(3) What 3D modeling software do you use most frequently? How long have you been using it? Compared to that software, what do you think are the key differences between our tool and your preferred software? What are the respective advantages and disadvantages?

P1 highlighted the advantages of the tool, including its ability to make objects soft for modeling, allowing for more flexible and realistic deformations compared to setting control points manually in traditional software, although the manual method is more precise. Additionally, the system offers stronger realism, enabling users to see rendered changes during deformation and creating natural irregularities that better match real-world shapes. However, the flexibility comes at the cost of precision, and the system does not support splitting objects into two parts. P2 emphasized the tool's higher design efficiency compared to traditional modeling software, making it better suited for quickly expressing and brainstorming creative ideas. It also facilitates presentations and enhances communication with other designers. The ability to rotate perspectives and clearly view 3D models in a virtual reality environment is more intuitive than the traditional three-view method. P3 noted the stronger sense of immersion and the low learning curve, making it suitable for children. The real-time texture rendering during shape editing enhances the shape editing process. However, compared to traditional modeling software, the resulting objects are less detailed. P4 compared the tool to Blender, noting that the key advantage of the VR tool lies in its realism. However, Blender offers more diverse editing options, making it better suited for complex modeling needs. P5 discussed the fundamental difference in 3D modeling logic: traditional tools like 3DMax and ZBrush are mesh-based, requiring adjustments from points to lines to surfaces, while this system mirrors natural manipulation and plasticity, avoiding the need to learn internal operational logic. Advantages include intuitive and convenient operation, which better supports creative expression in 3D design, and realistic deformation without requiring rigging. Traditional tools struggle with deformation due to fixed meshes. However, this system is less capable of fine detailing compared to 3DMax, struggles with producing symmetrical, streamlined shapes, and assigns vertex colors directly, which negatively impacts rendering quality when modeling from scratch. P6 highlighted the low learning curve of the system, which allows users to achieve desired deformations quickly while benefiting from excellent real-time rendering. However, precise control is challenging, such as when adjusting the relative position of merged objects.

(4) What aspects do you find unsatisfactory, and do you have any suggestions for improvements? Are there any features you think should be added?

P1 suggested adding support for splitting objects into two parts and increasing the density of modeling objects to enhance surface detail during reconstruction. P2 recommended introducing a sculpting-style workspace with a fixed object position that only allows rotation along the y-axis, enabling the use of more precise tools, such as small chisels, for editing. They also suggested adding an undo function and providing the ability to repair 3D Gaussian-based model surfaces after fracturing. P3 proposed implementing a grid-based snapping feature when merging objects to counteract hand-tracking instability and ensure precise placement. They also suggested adding functionality to repair fractured 3D Gaussian-based model surfaces. P4 highlighted the need for more guidance on the appropriate force to apply to different materials and suggested adopting a pinch gesture mechanism similar to Blender's sculpting tools, where the applied force decreases as the area increases. They also emphasized the importance of enabling repairs for fractured 3D Gaussian-based model surfaces. P5 recommended adding an undo function, supporting the creation of symmetrical, streamlined shapes by smoothing nearby points, and introducing a feature to split objects into two parts. P6 proposed adding a grid-based snapping system for precise positioning during object merging to mitigate the effects of hand-tracking instability. They also suggested reducing the likelihood of fractures in 3D Gaussian-based model surfaces.

3.2 Semi-Structured Interview Results (Novices, P7-P12)

(1) Do you think the designs created using this tool align with your initial expectations?

P7 observed minor discrepancies in physical properties but found the designs largely met expectations. P8 stated their projects generally met expectations. P9 found the outcomes satisfactory but saw room for improvement. P10 noted the results met expectations but merging objects was cumbersome. P11 expressed satisfaction with the watermelon model's results. P12 confirmed the designs met expectations.

(2) During your experience with the tool, what aspects were the most satisfying and the least satisfying? Additionally, what do you consider to be the greatest difficulty encountered while using the tool?

Most Satisfying: P7 found the intuitive operation, such as "modeling like clay," to be highly satisfactory due to its low learning curve. P8 appreciated the wide range of hands-on modeling options and material adjustments. P9 highlighted the tool's diverse and comprehensive features, including gravity-assisted functionality. P10 valued the high usability of the tool, noting its support for high frame rates and smooth real-time performance. P11 praised the freedom in detailed operations and the smooth functionality of basic features. P12 appreciated the realistic object rendering provided by the tool.

Least Satisfying: P7 noted performance issues such as insufficient particle numbers and surface material effects that need improvement. P8 highlighted the lack of split and undo functionalities. P9 pointed out imprecise hand tracking as a significant drawback. P10 mentioned that the object deformation behavior was overly sensitive, leading to accidental triggers. P11 expressed dissatisfaction with the limited primitive shape options. P12 observed that insufficient particle numbers caused positional errors after merging objects.

Greatest Difficulty: Hand-tracking issues were a recurring challenge, with imprecision, significant errors, and instability making position adjustments and operations difficult for users.

(3) From the perspective of usability (e.g., whether the tool is easy to learn and use), smoothness (e.g., whether operations are seamless and free from noticeable delays), realism (e.g., whether physical simulations of shape deformations meet your expectations), or any other aspects, please provide your feedback.

The tool was widely regarded as user-friendly, with low learning curves and easy-to-use interfaces across all participants (P7-P12). The system offered smooth and stable interactions with high frame rates, providing a seamless experience without noticeable lags or motion sickness (P7, P8, P10, P11, P12). In terms of realism, participants appreciated the adherence to physical rules and the overall accuracy in modeling (P8, P9, P10), though some noted areas for improvement, such as material diversity, detailed physical behaviors, and surface smoothing (P7, P8, P9, P11). Despite its strengths, hand-tracking performance was identified as a consistent issue, with imprecision and instability affecting precise operations and adjustments (P9, P12).

(4) What aspects do you find unsatisfactory, and do you have any suggestions for improvements? Are there any features you think should be added?

Suggestions for improvement included adding an undo function (P7, P9, P12), enhancing particle numbers for better surface reconstruction (P7, P12), and introducing more geometric presets like ellipsoids and rectangular prisms (P8, P12). Other proposals involved supporting single-dimensional shape adjustments (P9), optimizing hand-tracking, and improving collision box handling (P10). Participants also recommended adding object separation and text insertion (P8).

4 USER STUDY 2: COMPARISON WITH BLENDER

Below, we present detailed subjective feedback from our six participants.

4.1 Semi-Structured Interview Results

(1) What do you think are the key differences in operational experience between VR-Doh and Blender?

Participants highlighted several key differences between VR-Doh and Blender. P1 noted that VR-Doh's 3D perspective makes macroscopic object shaping in a spatial environment more convenient, while Blender's 2D perspective is better suited for detailed operations using a mouse. P2 emphasized that VR-Doh is easy to learn, with a simple operational logic that is highly accessible. Although its modeling logic may not be immediately clear, it becomes easy to understand after watching a tutorial, enabling one to know the next steps to take instinctively. However, the lack of tactile feedback makes fine control over objects more difficult. Blender, in contrast, has a higher learning curve but offers greater precision from a professional modeling perspective. P3 likened VR-Doh to the experience of working with real clay models, but noted its limitations in fine detail control, whereas Blender excels in precision. P4 observed that VR-Doh focuses on holistic, overall manipulation, while Blender is better for localized, detailed adjustments. Lastly, P5 pointed out that VR-Doh's operations are more intuitive and easier to start, while Blender's quantitative approach is more precise, reducing unintended errors. P6 noted VR-Doh does not require the abstraction of objects into vertices, edges, and surfaces for manipulation, whereas Blender necessitates the mental reconstruction of objects into vertices, edges, and surfaces in order to perform operations. Therefore, VR-Doh aligns more intuitively with the process of modeling objects

(2) What do you think are the key advantages and disadvantages of VR-Doh compared to Blender?

P1 highlighted that VR-Doh enables quick modeling using hands, efficiently realizing design concepts and avoiding the issue of "hands lagging behind the mind." However, it has disadvantages in terms of precision during object shaping. P2 noted that while VR-Doh's operations are more intuitive, its lack of precision makes it less suitable for professional users. Conversely, Blender offers higher precision but does not align well with habitual operations for modifying geometric objects. P3 emphasized VR-Doh's intuitive usage but mentioned that it lacks fine operational control. Similarly, P4 observed that VR-Doh is more intuitive, with realistic physical deformation that corresponds to real-world experiences, but it struggles with fine object adjustments. Blender, in contrast, allows direct manipulation of the mesh for better geometric control, but it requires significant experience to use effectively, and its outputs may not align with physical realism. Lastly, P5 pointed out that VR-Doh facilitates quick comprehension of modeling tasks but is limited by its lack of precise control and the absence of an undo function. P6 found VR-Doh operates on a WYSIWYG (What You See Is What You Get) basis, eliminating the need for frequent switching between edit and render modes. However, the selected manipulation area in VR-Doh is relatively large, which limits the ability to precisely control what can or cannot be selected. Additionally, Blender offers a reversible modeling process through the use of Modifiers, enabling users to preview geometric changes without permanently altering the object.

REFERENCES

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. 2024. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21476–21485.

Xinyu Gao, Ziyi Yang, Bingchen Gong, Xiaoguang Han, Sipeng Yang, and Xiaogang Jin. 2024. Towards Realistic Example-based Modeling via 3D Gaussian Stitching. *arXiv preprint arXiv:2408.15708* (2024).

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*. 1–52.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Deborah Sulsky, Shi-Jian Zhou, and Howard L Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Computer physics communications* 87, 1-2 (1995), 236–252.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4389–4398.