

Natural Tactile Interaction with Virtual Clay

Héctor Barreiro, Joan Torres and Miguel A. Otaduy

Abstract—Despite many past efforts to develop haptic experiences of virtual clay modeling, natural interaction with virtual clay remains a hard challenge. In this work, we propose a computational solution for the interactive simulation of clay-like materials with unprecedented realism, coupled with free-air tactile rendering that provides a natural tangible experience. Our solution includes a novel particle-based model of viscoplasticity for efficient interactive simulation, and an optimization-based ultrasound rendering algorithm that takes as input the interaction forces between a virtual hand model and the clay-like material. We demonstrate the effectiveness of our method through expressive creative experiences.

I. INTRODUCTION

One of the ambitions of virtual reality (VR) is to let people create 3D forms without the constraints of real-world objects, materials and procedures [1]. Today, VR provides commodity immersive display and hand tracking, two of the major requirements for effective VR-based 3D modeling applications. However, simulation models and interaction techniques have not reached the maturity necessary for virtually modeling complex materials such as clay in a natural way.

Virtual modeling of clay-like materials should include the following features: realistic behavior of the material, natural hands-on interaction, and tactile feedback. In this paper, we present a simulation and interaction model that includes all these features, and therefore enables natural and tangible VR-based modeling of clay. We rely on an existing natural hand simulation model [2] to enable bidirectional coupling between the user and the virtual clay simulation. We use hand tracking to command the simulated virtual hand, we simulate the physical interaction between this hand and the clay-like material, and we feed the interaction forces back to the hand simulation. We also use these forces to command a tactile rendering algorithm. The two major technical contributions in our work, which enable the overall system, are the simulation model of clay and the tactile rendering algorithm.

Clay is a complex material that combines properties of solids (permanent shape) with properties of fluids (plastic flow). To date, no interactive simulation method represents well this complex behavior. We present a particle-based model, following the position-based dynamics (PBD) and position-based fluids (PBF) methods [3], [4], [5], which captures well and efficiently the main features of clay-like materials. Our model, described in Section III, includes novel formulations of constraints for viscosity, elastoplasticity, and friction, which together produce clay-like behavior.

Clay is modeled with bare hands, and contact may occur both at finger pads as well as on large areas of the palm.



Fig. 1. Arts & crafts in the virtual classroom. We introduce a novel model of clay that allows interactive and highly realistic deformations, merging, and splitting. We also introduce an ultrasound rendering algorithm that enables a tangible interactive experience.

Ultrasound rendering, despite its power limitations, offers the ability to stimulate the full hand in free air, with a good trade-off between coverage and resolution [6]. In Section IV, we present a rendering algorithm that takes as input the contact forces between the virtual hand model and the clay material, and outputs focal pressure point commands for an ultrasound array. Our algorithm solves an optimization formulation that can accommodate perceptual weight maps.

We demonstrate the effectiveness of our simulation model and rendering algorithm on several examples of creative experiences with complex and rich clay material, such as the one shown in Fig. 1.

II. RELATED WORK

VR Modeling of Clay Clay is a viscoplastic material whose behavior resembles both a solid and a fluid. It exhibits microscopic material bonds that preserve shape, but these bonds are fragile and the material flows even under small stress, although with very high viscosity. Due to this complexity, simulation of clay is a computationally challenging problem, and existing interactive methods barely approximate its true behavior. Nevertheless, multiple works have attempted to partially model the behavior of clay in VR applications, including haptic feedback.

Early models for VR-based modeling of clay used combinations of spline surfaces and voxelized volumes [7]. To counteract the high cost of voxelization, McDonnell et al. proposed adaptive volumetric representations [8]. They combined interactive editing capabilities with 3-Degree-of-Freedom (DoF) haptic feedback.

A critical challenge of clay-like materials is the combination of elastic behavior with plastic flow, which is essential

for its modeling capabilities. Several works combine global (i.e., mesh-based elastic models) with local deformation (i.e., voxel-based flow models) [9]. These models have also been integrated with 3-DoF haptic feedback, and augmented with end-effectors with pressure sensors, to better mimic the interaction of the hand with the clay material [10].

There are also models that address modeling clay on a potter’s wheel, and leverage the revolution shape of objects, but limit general interaction. Some methods use spline surfaces combined with haptic feedback and augmented reality [11], and others define circular sector elements, which also drive haptic feedback [12]. It is also possible to add volume preservation to cylindrical elements [13].

Position-Based Viscoelasticity Simulation Despite all the efforts discussed above, the physical behavior of clay is only partially approximated in previous VR applications. No previous interactive simulation method reproduces the highly viscous flow and ductile fracture of clay-like materials. PBD and PBF methods [3], [4], [5], due to their efficiency and flexibility, offer a potential solution.

One possible approach to model clay with PBD would be to consider it an elastic solid. Some works add efficient cutting to viscoelastic PBD solids [14], [15]. However, clay exhibits a fluid-like behavior that allows material to merge, and these methods do not support merging. The other approach to model clay with PBD would be to consider it a viscous fluid. Again some works propose models for efficient viscosity simulation within PBF [16], [17], but they do not reach the extreme viscous behavior of clay. Our simulation model allows the user to conform, split, and merge virtual clay much like in the real world.

Ultrasound Haptic Rendering Ultrasound phased arrays produce an acoustic pressure field in the surrounding volume, which can be controlled by commanding the phase of the various ultrasound transducers [18]. Mapping a desired haptic stimulus to phase commands is however very challenging, which has motivated the design of simpler control metaphors. One method, denoted *amplitude modulation* consists of controlling the intensity and location of a few focal pressure points [19]. Another method, denoted *spatiotemporal modulation* consists of controlling trajectories of focal points that produce a persistent tactile sensation [20], [21].

To render the interaction of the user with the clay material, we must control the ultrasound device as a function of a temporally and spatially varying pressure field. Barreiro et al. formulated optimization-based solutions to this rendering problem for both amplitude and spatiotemporal modulation. Their solution for amplitude modulation formulates and solves rendering as a clustering problem [22]. Their solution for spatiotemporal modulation, on the other hand, formulates and solves rendering as a path routing problem [23]. Jang and Park [24] also solved amplitude modulation as an optimization, following a hill-climbing algorithm.

Our ultrasound rendering algorithm follows the clustering-based amplitude-modulation approach of Barreiro et al. [22], with two notable differences. First, the target pressure field is extracted from a PBD simulation of clay. Second, we add

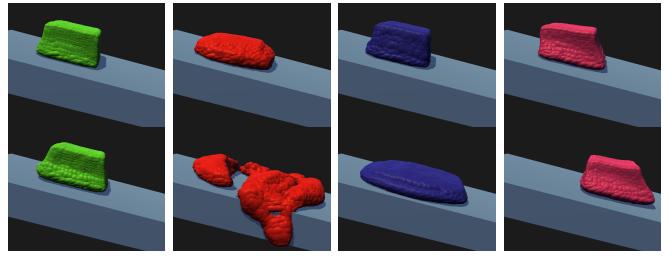


Fig. 2. Ablation study of the novel constraints in our PBF clay model. We drop a block of clay on an incline, and we show its deformation during the impact (top) and one second later (bottom). From left to right: our full clay model (green); without viscosity constraints, the material flows fast and fractures (red); without elastoplasticity constraints, the material drifts (blue); and without friction constraints the block slides (magenta).

a perceptual weight map to the clustering algorithm.

III. VISCOPLASTIC MODEL OF CLAY

Particle-based Lagrangian representations offer the best compromise for the simulation of complex viscoplastic materials such as clay. They support robust strain metrics for elasticity through particle bonds, and they also support efficient plastic flow by dynamically (de)activating such bonds. In particular, we build on PBF simulation method [4]. PBF formulates particle dynamics as a constrained minimization problem, and constraints are resolved using a relaxation algorithm. In PBF, solid and fluid constraints can be seamlessly handled, and the main difference is whether they use (semi-)permanent or temporary particle bonds. We start this section with a summary of the PBF algorithm for the simulation of incompressible fluids, which sets the baseline for our method.

To apply the PBF algorithm to highly viscoplastic materials such as clay, we propose novel constraint formulations that capture the major effects of the material. First, we model viscosity by expressing a constraint on strain rate. To this end, we must turn the velocity-based formulation into position-based constraints. Second, we model elastoplasticity using semi-permanent distance constraints. We achieve plasticity by integrating a hysteresis threshold on the (de)activation of the constraints. And third, we model frictional contact using anchor constraints. All in all, these constraint formulations produce the characteristic clay behavior of extreme viscoplasticity. Fig. 2 compares the effect of each of the constraints on the behavior of clay.

A. PBF Simulation Model

In PBF, the material is discretized by a set of particles, and the state of each particle is defined by its position \mathbf{x}_i and velocity \mathbf{v}_i . Particles also have mass m_i . PBF uses a smoothed particle hydrodynamics (SPH) [25] approach to define variables in the continuum, and hence the value of a variable a at an arbitrary position \mathbf{x}_i is evaluated as:

$$a(\mathbf{x}_i) = \sum_j \frac{m_j}{\rho_j} a_j W_{ij}, \quad (1)$$

with $W_{ij} = W(\mathbf{x}_{ij}) = W(\mathbf{x}_i - \mathbf{x}_j)$ the evaluation at \mathbf{x}_i of a smoothing kernel centered at \mathbf{x}_j with support radius h ,

ALGORITHM 1: PBF step

Input: Previous particle states $\{\mathbf{x}_i^0\}, \{\mathbf{v}_i^0\}$.
Output: Updated particle states $\{\mathbf{x}_i\}, \{\mathbf{v}_i\}$.

```

/* Compute unconstrained positions */
foreach particle i do
     $\mathbf{x}_i = \mathbf{x}_i^0 + \Delta t \mathbf{u}_i^0 + \frac{\Delta t^2}{m_i} \mathbf{f}_i(\mathbf{x}_i^0)$ 
end

/* Relaxation iterations */
foreach iteration do
    /* Loop over constraint types */
    foreach constraint type k do
        /* Solve constraints in parallel */
        foreach constraint j do
            project  $\{\mathbf{x}_i\}$  such that  $\mathbf{C}_{k,j}(\{\mathbf{x}_i\}) = \mathbf{0}$ 
        end
    end
end

/* Update velocities */
foreach particle i do
     $\mathbf{v}_i = \frac{\mathbf{x}_i - \mathbf{x}_i^0}{\Delta t}$ 
end

```

a_j the value of the variable for the j^{th} particle, and ρ_j the density field evaluated at \mathbf{x}_j .

As noted above, in PBF the mechanical behavior of the material is defined and solved using constraints. In the following subsections we detail the formulation of the different types of constraints of our viscoplastic material model. First, let us define some general notation, where C_k represents each one of k types of constraints, and $\mathbf{J}_{k,i} = \frac{\partial C_k}{\partial \mathbf{x}_i}$ is its Jacobian with respect to the position of a particle.

Constraints are solved one at a time, projecting the particle positions such that linearized constraints are fulfilled. The particle projection for a constraint C_k is computed as [5]:

$$\Delta \mathbf{x}_i = -\frac{\frac{1}{m_i} \mathbf{J}_{k,i}^T C_k}{\sum_j \frac{1}{m_j} \|\mathbf{J}_{k,j}\|^2}. \quad (2)$$

The overall PBF algorithm proceeds as outlined in Algorithm 1. First, it computes unconstrained motion given by inertial and gravity forces. Then, it iterates over the constraints, projecting the particles to the closest valid configuration. To maximize the efficiency of a GPU implementation, we iterate over constraint types in Gauss-Seidel fashion, but we execute all constraints of the same type in fully parallel Jacobi fashion. To conclude, the PBF algorithm computes particle velocities using finite differences.

We characterize clay as an incompressible extremely viscoplastic material. To model incompressibility, we use a constraint on the particle density, as done regularly in PBF methods [26], [4]. Next, we discuss the novel types of constraints used in our model.

B. Viscosity

Viscosity damps the differences in local velocities within a medium, and these velocity differences can be best charac-

terized by the symmetric part of the strain rate tensor [27]. Then, we design a viscosity constraint as:

$$\mathbf{C}_{viscosity}(\mathbf{v}_i) = \nabla \mathbf{v}_i + (\nabla \mathbf{v}_i)^T = \mathbf{0}. \quad (3)$$

Our constraint can be regarded as a modified version of the viscoelasticity constraint of Barreiro et al. [17]. Unlike them, we care only about viscosity, not viscoelasticity, and we model the elastic behavior of the material through an explicit elastoplastic constraint discussed later.

The velocity gradient in (3) is obtained by differentiating the SPH kernel (1) to obtain:

$$\nabla \mathbf{v}_i = \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \nabla W_{ij}^T. \quad (4)$$

To avoid damping rotational motion, we define particle velocities using a corotational finite-difference approximation as $\mathbf{v}_i = \frac{\mathbf{x}_i - \mathbf{x}_i^r}{\Delta t}$, where \mathbf{x}_i^r represents the rotational part of the displacement [17].

In Fig. 2 we compare the simulation with (green) and without (red) the viscosity constraint. With viscosity, the material becomes rigid soon after an impact, and without viscosity it flows fast and quickly fractures.

C. Elastoplasticity

Viscosity alone cannot represent the behavior of materials such as clay. The material exhibits a resistance to deviate from its stable configuration (i.e., elasticity), although it soon flows into a different configuration. We model this behavior using an elastoplastic model at low velocities, combined with the previous viscous model at larger velocities.

In PBF, elasticity can be handled using distance constraints between pairs of particles \mathbf{x}_i and \mathbf{x}_j . The constraint is formulated as:

$$\mathbf{C}_{elasticity}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij} = \mathbf{0}, \quad (5)$$

where d_{ij} is the rest length of the constraint.

To model plastic flow, we (de)activate elasticity constraints based on the relative velocity of particle pairs projected to their connecting segment. If the projected relative velocity of a particle pair falls below a threshold, we activate an elasticity constraint. On the other hand, if the relative velocity grows over a threshold, we deactivate the elasticity constraint to allow the particles to flow. We achieve stable behavior through the use of distinct (de)activation thresholds (i.e., hysteresis).

In Fig. 2 we compare the simulation with (green) and without (purple) the elastoplasticity constraint. Without elastoplasticity the material drifts slowly, and with elastoplasticity it soon stops deforming.

D. Contact and Friction

We model frictional contact with arbitrary objects, such as the dynamically deforming hand, by rasterizing ghost particles on these objects, and then setting particle-based constraints. We use two types of constraints, to model impenetrability and friction, as we detail next.



Fig. 3. Our tactile rendering algorithm proceeds according to these steps, from left to right: (i) The particle-based simulation computes forces and deformation on clay particles due to the interaction with ghost particles on the hand (Section III). (ii) We compute a pressure field on the ghost particles on the hand (Section IV-A). (iii) We compute the location and pressure of focal points such that the reconstructed pressure field is optimal, and these focal points are commanded to an ultrasound array for amplitude-modulation rendering (Section IV-B).

For each solid object, we rasterize ghost particles in its rest configuration, and transform these particles according to the motion of the object. For the hand, we mesh its interior with tetrahedra, and transform the particles using the barycentric coordinates of the enclosing tetrahedra.

We execute collision detection at the beginning of each time step. For each ghost particle that falls within a radius R of a clay particle, we set an impenetrability constraint with the same formulation of the distance constraint (5), with $d_{ij} = R$. As done by Macklin et al. [28], we ensure that impenetrability only repels and does not attract particles, by designing one-sided contact normals.

To handle Coulomb friction, we use a sliding-anchor model, inspired by spring-based frictional contact models [29], but adapted to the constraint-based PBF simulation algorithm. Every time a particle \mathbf{x}_i suffers a new contact, we set an anchor \mathbf{a}_i at an offset R from the location of the corresponding ghost particle. Then, we add a zero-distance constraint between \mathbf{x}_i and \mathbf{a}_i . However, when applying the particle projection (2) due to the friction constraint, we limit it based on the friction coefficient μ and the particle projection due to the impenetrability constraint, i.e., $\|\Delta\mathbf{x}_{friction,i}\| \leq \mu \|\Delta\mathbf{x}_{impenetrability,i}\|$. Furthermore, at the end of each step we slide each anchor, such that the distance to the colliding particle is given by the total particle projection of the friction constraint.

In Fig. 2 we compare the simulation with (green) and without (magenta) the friction constraint. Without friction the material slides down the incline.

IV. ULTRASOUND HAPTIC RENDERING

Based on the interactive simulation of clay presented in the previous section, in this section we describe how we provide feedback to the user, in the form of ultrasound-based tactile rendering. We divide this task in two steps, outlined in Fig. 3. First, we obtain a target pressure field from the contact forces between the hand model and the clay material. Second, we design an amplitude-modulation rendering algorithm to compute focal pressure points that are commanded to the ultrasound device. Next, we describe these two steps in detail.

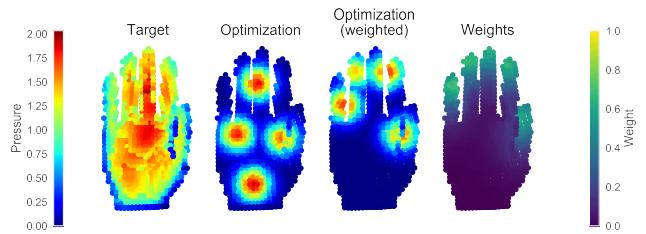


Fig. 4. Our rendering algorithm supports perceptual weight maps to favor higher accuracy on more sensitive areas of the hand, such as the finger pads. On the left, we show particles color-coded according to their target pressure; on the right, we show the weight map. The two optimizations indicate the reconstructed pressure with and without weight map.

A. Interaction Pressure Field

Similar to previous works on ultrasound rendering of fluids, we extract a pressure field from the interaction between the simulated hand and the clay material. However, since our PBF simulation model differs from the Eulerian model of Barreiro et al. [22] or the SPH model of Jang and Park [24], we require a different method to compute the pressure field on the simulated hand.

As described in Section III-D, we set ghost particles on the hand to handle contact with the clay material. We use these same ghost particles to recover the pressure field on the hand. On every projection step due to an impenetrability contact constraint, we compute the contact force on the corresponding ghost particle. The force $\Delta\mathbf{f}_i$ of a projection step (2) can be obtained as:

$$\Delta\mathbf{f}_i = -\frac{\frac{1}{\Delta t^2} \mathbf{J}_{k,i}^T C_k}{\sum_j \frac{1}{m_j} \|\mathbf{J}_{k,j}\|^2}, \quad (6)$$

with $k = \text{impenetrability}$.

By adding up the forces from all constraint iterations in a time step, we obtain the total collision force \mathbf{f}_i on each ghost particle. Then, same as Jang and Park [24], we obtain the target pressure p^* by computing the component of the force normal to the surface at every ghost particle \mathbf{x}_i :

$$p^*(\mathbf{x}_i) = \max(-\mathbf{n}_i^T \mathbf{f}_i, 0), \quad (7)$$

where \mathbf{n}_i is the outward surface normal. Note that we only consider ghost particles that are facing the location of the ultrasound array.

The middle image in Fig. 3 shows a sample interaction, with ghost particles in the hand color-coded according to their pressure.

B. Amplitude-Modulation Rendering

Once the target pressure field is defined, we optimize a set of focal pressure points and their corresponding pressure, which are commanded to the ultrasound device for amplitude-modulation rendering. We opt for amplitude modulation instead of spatiotemporal modulation to accommodate a user-defined perceptual weighting function $\alpha(\mathbf{x})$, as we see next. This function allows us to focus attention on more sensitive areas of the hand, as shown in Fig. 4.

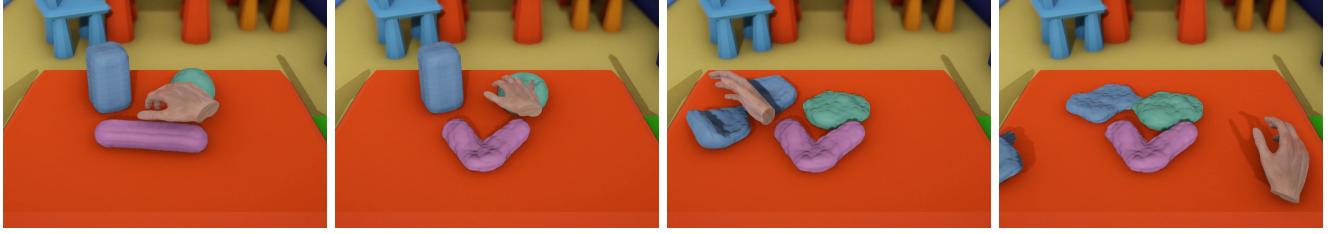


Fig. 5. Interactive modeling of colorful clay figures. Our clay model supports robust viscoplastic deformations, which are key for generating arbitrary stable shapes, and for merging and splitting material.

We follow the clustering algorithm of Barreiro et al. [22], augmented with the perceptual weighting function. Specifically, we set weights $\alpha = 1$ at finger pads, and smoothly decay the weight function based on the distance to the closest finger pad, following roughly mechanoreceptor densities [30].

The algorithm of Barreiro et al. first finds the locations of focal points through a k-means clustering problem. We search for the $\{\mathbf{x}_k\}$ focal point locations, using the weighted target pressure $\alpha(\mathbf{x}) p^*(\mathbf{x})$:

$$\{\mathbf{x}_k\} = \arg \min_k \sum_{\mathbf{x}_i \in Vor(\mathbf{x}_k)} \alpha(\mathbf{x}_i) p^*(\mathbf{x}_i) \|\mathbf{x}_i - \mathbf{x}_k\|^2, \quad (8)$$

where $Vor(\mathbf{x}_k)$ is the Voronoi region of focal point \mathbf{x}_k .

Then, the algorithm computes the rendered pressures $\{p_k\}$ of the focal points, by minimizing the difference between the target pressure and the reconstructed pressure. This reconstructed pressure assumes a Gaussian fall-off with standard deviation σ given by the wavelength of the ultrasound signal.

$$p_k = \arg \min_{\mathbf{x}_i \in Vor(\mathbf{x}_k)} \left(p_k e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma^2}} - p^*(\mathbf{x}_i) \right)^2 \Rightarrow \\ p_k = \frac{\sum_{\mathbf{x}_i \in Vor(\mathbf{x}_k)} p^*(\mathbf{x}_i) e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma^2}}}{\sum_{\mathbf{x}_i \in Vor(\mathbf{x}_k)} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{\sigma^2}}}, \forall k. \quad (9)$$

The computed focal point positions and pressures are commanded to the ultrasound device on each rendering frame, resulting in tangible feedback of the user's interaction.

V. EXPERIMENTS

We have tested our simulation and rendering methods on two interactive modeling scenarios. Fig. 5 shows a kindergarten table with colorful blocks of modeling clay. The user deforms, splits and merges the blocks as shown in the images and the video. Fig. 7, on the other hand, shows a

potter's wheel with a block of clay. In this case, the user interacts with both hands to create a solid of revolution as the wheel rotates. Both scenes demonstrate the robustness of the clay-like material, which exhibits the extreme viscoplasticity of real-world clay. We render clay graphically using a screen-space ellipsoid splatting method. The depth buffer is smoothed to produce a continuous surface, but some bumpiness and interpenetrations may appear.

The ultrasound rendering algorithm provides tangible feedback of the interaction. As shown in Fig. 6, this feedback depends on the properties of the clay material. With a more plastic material, contact forces are smaller under the same user actions, which turn into lower rendered pressures.

In the kindergarten and pottery scenes, the number of particles is respectively 2 114 and 4 169. The full simulation and tactile rendering runs at approximately 25 fps. The cost is dominated by the simulation of the clay material (27 ms per frame in the kindergarten scene, 30 in the pottery scene), while the rendering algorithm takes only 2 ms per frame. The simulation of the hand takes 6 ms per frame, but it is executed in parallel on a different thread. Within the clay simulation, the cost is dominated by the elastoplasticity constraint. We use a time step of 1/60, which is smaller than the actual update rate. This, together with the numerical damping introduced by the PBF solver, makes the simulated physics appear slightly slower than real-world physics. All examples were executed on an AMD Ryzen 7 2700 8-core 3.20 GHz PC with 32 GB of RAM and a Nvidia GeForce GTX 1080 Ti GPU with 11 GB of RAM. For ultrasound rendering, we have used a STRATOS Explore (USX) device from Ultraleap, running at 40kHz.

VI. DISCUSSION AND FUTURE WORK

In this work, we have demonstrated a solution for tangible interactive clay modeling, based on a clay simulation method and an ultrasound-based rendering algorithm. This work enables novel rich creative experiences, thanks to the robust handling of extreme viscoplasticity, and to the optimization of the rendering output for arbitrary interactions. Given the working solution, now it is possible to turn the attention to the investigation of choices and improvements.

We identify three limitations, which could motivate future work. One is the dominant cost of the elastoplasticity constraint, which may produce 10 to 20 constraints per particle. This cost limits the number of particles in the scene, and therefore the resolution of the clay material.

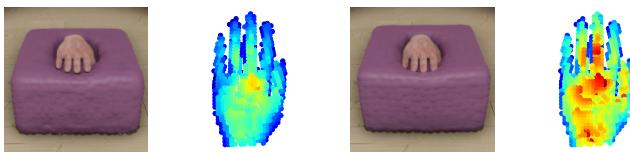


Fig. 6. Rendered pressures vary depending on the properties of the material. With less plasticity (right), the material flows less and imposes a higher pressure field on the hand.

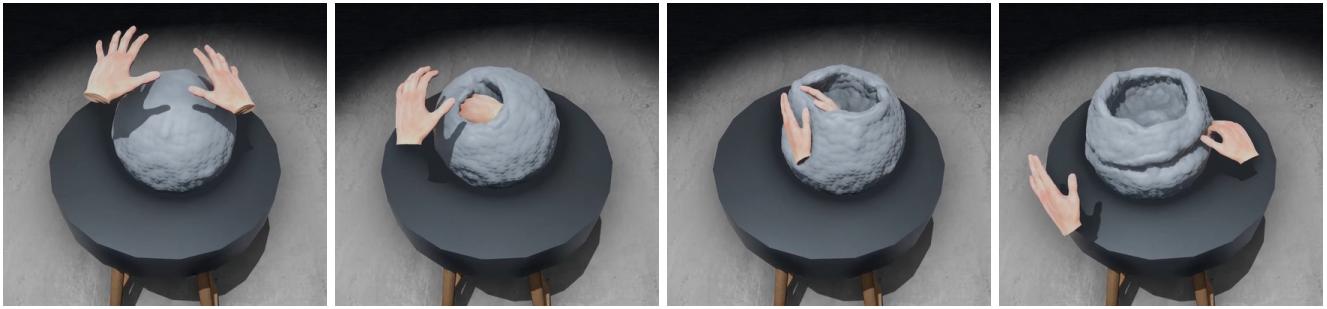


Fig. 7. Tangible modeling of clay on a potter’s wheel. We simulate two-handed interaction with clay, providing a natural hands-on experience.

Another limitation is the coverage of the amplitude-modulation rendering method. As evidenced in Fig. 4, the quick decay of the focal points prevents matching large pressure areas on the hand. Nevertheless, the weight map allows us to focus on perceptually relevant locations. A possible alternative is to investigate spatiotemporal rendering methods to reach larger coverage [20], [21], [23].

Last but not least, the dexterity of the interaction is limited when trying to execute detailed modeling with the fingers. The reasons are multiple, including the low resolution of the particles, but probably also the limited fidelity of the tactile stimulation. We chose ultrasound-based stimulation due to its free-air operation and ease of use, but it is worth investigating what type of tactile feedback is best suited to enable dexterous modeling with the fingers.

ACKNOWLEDGMENT

We wish to thank Dani Lobo and Mickeal Verschoor for help with the integration of CLAP. This work was funded in part by the European Research Council (ERC Consolidator Grant 772738 TouchDesign) and the Spanish Ministry of Science (grant RTI2018-098694-B-I00 VizLearning).

REFERENCES

- [1] E. Bouzbib, G. Bailly, S. Haliyo, and P. Frey, ““can i touch this?”: Survey of virtual reality interactions via haptic solutions,” in *Proceedings of IHM Conference (To appear)*, 2021.
- [2] M. Verschoor, D. Lobo, and M. A. Otaduy, “Soft hand simulation for smooth and robust natural interaction,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018, pp. 183–190.
- [3] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, “Position based dynamics,” in *Workshop on Virtual Reality Interaction and Physical Simulation*, 2006.
- [4] M. Macklin and M. Müller, “Position based fluids,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 104:1–104:12, 2013.
- [5] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin, “A survey on position-based simulation methods in computer graphics,” *Comput. Graphics Forum*, vol. 33, no. 6, pp. 228–251, 2014.
- [6] M. A. Otaduy, A. Okamura, and S. Subramanian, “Haptic technologies for direct touch in virtual reality,” in *ACM SIGGRAPH 2016 Courses*, ser. SIGGRAPH ’16. ACM, 2016, pp. 13:1–13:123.
- [7] F.-L. Krause and J. Lüddemann, “Virtual clay modelling,” in *Proceedings of the Fifth IFIP TC5/WG5.2 International Workshop on Geometric Modeling in Computer Aided Design on Product Modeling for Computer Integrated Design and Manufacture*, ser. GMCAD ’96. GBR: Chapman & Hall, Ltd., 1997, p. 162–175.
- [8] K. T. McDonnell, H. Qin, and R. A. Włodarczyk, “Virtual clay: A real-time sculpting system with haptic toolkits,” in *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ser. I3D ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 179–190.
- [9] G. Dewaele and M.-P. Cani, “Interactive global and local deformations for virtual clay,” in *Proc of Pacific Graphics Conf*, 2003.
- [10] A. Pihuit, P. G. Kry, and M. Cani, “Hands on virtual clay,” in *2008 IEEE Int Conf on Shape Modeling and Applications*, 2008.
- [11] G. Han, J. Hwang, S. Choi, and G. J. Kim, “Ar pottery: Experiencing pottery making in the augmented space,” in *Virtual Reality*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [12] J. Lee, G. Han, and S. Choi, “Haptic pottery modeling using circular sector element method,” in *Haptics: Perception, Devices and Scenarios*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [13] S. Chaudhury and S. Chaudhuri, “Volume preserving haptic pottery,” in *2014 IEEE Haptics Symposium (HAPTICS)*, 2014, pp. 129–134.
- [14] I. Berndt, R. Torchelsen, and A. Maciel, “Efficient surgical cutting with position-based dynamics,” *IEEE Computer Graphics and Applications*, vol. 37, no. 3, pp. 24–31, 2017.
- [15] L. Xu, Y. Lu, and Q. Liu, “Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time,” *R Soc Open Sci*, vol. 14, no. 5, 2018.
- [16] T. Takahashi, T. Nishita, and I. Fujishiro, “Fast simulation of viscous fluids with elasticity and thermal conductivity using position-based dynamics,” *Computers & Graphics*, vol. 43, pp. 21–30, 2014.
- [17] H. Barreiro, I. García-Fernández, I. Alduán, and M. A. Otaduy, “Conformation constraints for efficient viscoelastic fluid simulation,” *ACM Trans. Graph.*, vol. 36, no. 6, 2017.
- [18] S. Inoue, Y. Makino, and H. Shinoda, “Active touch perception produced by airborne ultrasonic haptic hologram,” in *2015 IEEE World Haptics Conference (WHC)*, 2015, pp. 362–367.
- [19] B. Long, S. A. Seah, T. Carter, and S. Subramanian, “Rendering volumetric haptic shapes in mid-air using ultrasound,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 181, 2014.
- [20] G. Korres and M. Eid, “Haptogram: Ultrasonic point-cloud tactile stimulation,” *IEEE Access*, vol. 4, pp. 7758–7769, 2016.
- [21] B. Kappus and B. Long, “Spatiotemporal modulation for mid-air haptic feedback from an ultrasonic phased array,” *The Journal of the Acoustical Society of America*, vol. 143, no. 3, pp. 1836–1836, 2018.
- [22] H. Barreiro, S. Sinclair, and M. A. Otaduy, “Ultrasound rendering of tactile interaction with fluids,” in *2019 IEEE World Haptics Conference (WHC)*, 2019, pp. 521–526.
- [23] ——, “Path routing optimization for stm ultrasound rendering,” *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 45–51, 2020.
- [24] J. Jang and J. Park, “Sph fluid tactile rendering for ultrasonic mid-air haptics,” *IEEE Transactions on Haptics*, vol. 13, no. 1, 2020.
- [25] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annu. Rev. Astronomy and Astrophysics*, vol. 30, pp. 543–574, 1992.
- [26] K. Bodin, C. Lacoursiere, and M. Servin, “Constraint fluids,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 3, pp. 516–526, 2012.
- [27] J. Bender and D. Koschier, “Divergence-free sph for incompressible and viscous fluids,” *IEEE Trans. Visual Comput. Graphics*, vol. 23, no. 3, pp. 1193–1206, 2017.
- [28] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, “Unified particle physics for real-time applications,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 153:1–153:12, July 2014.
- [29] K. Yamane and Y. Nakamura, “Stable penalty-based model of frictional contacts,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 1904–1909.
- [30] G. Corniani and H. P. Saal, “Tactile innervation densities across the whole body,” *Journal of Neurophysiology*, vol. 124, no. 4, 2020.