Interactive Physics-Based Virtual Sculpting with Haptic Feedback

AVIRUP MANDAL, Indian Institute of Technology Bombay, India
PARAG CHAUDHURI, Indian Institute of Technology Bombay, India
SUBHASIS CHAUDHURI, Indian Institute of Technology Bombay, India

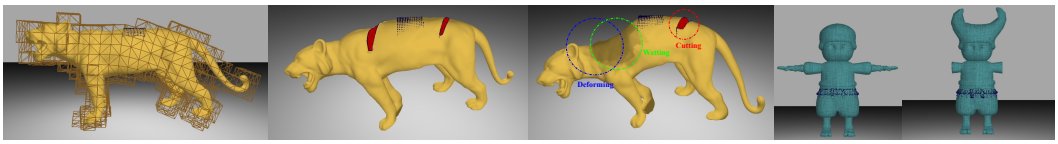

Fig. 1. From left to right we show surface visualization mesh embedded inside a volumetric simulation cage mesh, multiple cuts and an example with multiple sculpting operations on the Lioness model. Finally the last two images on the right depict original and sculpted versions of a Toy Ninja model.

Sculpting is an art form that relies on both the visual and tactile senses. A faithful simulation of sculpting, therefore, requires interactive, physically accurate haptic and visual feedback. We present an interactive physics-based sculpting framework with faithful haptic feedback. We enable cutting of the material by designing a stable, remeshing-free cutting algorithm called Improved stable eXtended Finite Element Method. We present a simulation framework to enable stable visual and haptic feedback at interactive rates. We evaluate the performance of our framework quantitatively and quantitatively through an extensive user study.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; *Haptic devices*; User studies; • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Virtual sculpting; Haptic feedback; Improved XFEM model; Cutting simulation.

ACM Reference Format:
Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2022. Interactive Physics-Based Virtual Sculpting with Haptic Feedback. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1, Article 5773229 (May 2022), 20 pages. https://doi.org/10.1145/3522611

1 INTRODUCTION

Sculpting is an art that relies on the perception of a material by touch, in addition to its visual appearance. Traditionally simulated virtual sculpting tools offer visual rendering of the object being sculpted but entirely miss the tactile aspect of the art form. Moreover, many of these tools perform shape editing in a purely geometric approach which does not capture accurate physical behaviour of material. Physically accurate mesh manipulation provides more natural and effective sculpting experience for digital artists. However, physics-based accurate simulation of deformable mesh manipulations is often computationally very expensive.

Authors' addresses: Avirup Mandal, avirupmandal@ee.iitb.ac.in, Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India, 400076; Parag Chaudhuri, paragc@cse.iitb.ac.in, Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India, 400076; Subhasis Chaudhuri, sc@ee.iitb.ac.in, Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India, 400076.

In this paper, we present a framework for stable simulation of interactive, physics-based virtual sculpting. Coupled with appropriate haptic feedback our framework allows for tactile interaction with the material being sculpted. The forces needed for the haptic feedback and the accurate behavior of the material being sculpted under these forces are generated using a robust physics-based simulation. Our sculpting framework can be used for reshaping 3D models starting from any arbitrary initial mesh.

Sculpting with clay consists of three main operations.

- **Deforming**: An artist deforms a lump a clay to shape it.
- **Wetting**: Adding water to the clay makes it malleable, which in turn helps to reshape parts of the same model differently.
- **Cutting**: Cutting is performed to make different shapes.

Recently, Mandal et al. [2021a] developed a framework for deforming and wetting an object mesh. We extend their work here to develop a fully functional sculpting framework with remeshing-free cutting. Our sculpting brushes allows a user to perform these sculpting operations virtually while receiving the appropriate haptic feedback for the same.

To perform accurate virtual sculpting simulation, at an interactive rate, we have to develop a robust and efficient algorithm for object cutting which is fast and real-time. For realistic user experience, we require high fidelity visual representation of virtual scenario coupled with accurate haptics force feedback. Moreover, visual and haptics feedback demand vastly different rate of update for smooth interaction and synchronization between them is a major challenge. So, in this work, we

- Present a robust and efficient simulation for object cutting that is remeshing-free. Our model is based on a Improved stable eXtended Finite Element Method (Section 4).
- Develop a multi-resolution, multi-timescale sculpting framework coupled with synchronized haptic and visual feedback (Section 6).

The rest of the paper is organized as follows. First we discuss the related works. Following that we delve into the detailed explanation of our remeshing-free cutting algorithm. Next, we present our haptic rendering solution in Section 5, that is used to implement the interactive virtual sculpting framework. We describe multi-resolution, multi-timescale simulation for cutting the mesh in Section 6. Finally, in Section 7 we show extensive qualitative, quantitative and user study results generated using our virtual sculpting framework.

2 RELATED WORK

In this section we review the existing works in the literature that are closely related to our works.

2.1 XFEM & Is-XFEM

There exists many algorithm for cutting and fracturing of meshes, most of which require remeshing the existing mesh. The two major disadvantages of remeshing are that it produces degenerate mesh elements which lead to instability and it is time consuming. We use a variation of remeshing-free eXtended Finite Element Method (XFEM) to simulate the dynamics of our cutting model.

Originally developed in material science [Moës et al. 1999] [Areias and Belytschko 2005] [Belytschko and Black 1999] [Zi and Belytschko 2003] [Zhu 2012], XFEM has found its way in computer graphics simulation [Koschier et al. 2017] [Chitalu et al. 2020]. The method is able to treat arbitrary cracks independent of the mesh and crack growth without any need for remeshing. All cracked or cut elements are enriched by a signed distance function which in turn provides extra degrees of freedom to the cracked nodes. The system matrix in original XFEM method can become singular, which makes the system unstable. When such a singular matrix arises, the authors [Koschier

et al. 2017] constrain the whole system dynamics and forcefully move the extra DOFs with the detached fragment for appropriate cut opening. In our work, we used another version of XFEM called Improved stable XFEM (Is-XFEM) [Wu and Li 2015] to improve system stability by avoiding the singular matrix generation. Recently, Mandal et al. [2021b] presented a novel graph-based remeshing-free fracture simulation algorithm that runs faster than the existing methods. However, their method is not real-time and thus not suitable for interactive applications.

2.2 Gaussian quadrature integration

Regular Gaussian quadrature rule [Gustafson and Hagler 1999] is not fit to solve the discontinuous integrals that originate from the Is-XFEM method. To solve the field equations over tetrahedral domains that are cut by implicit surfaces, we use a generalized and highly accurate surface and volume Gaussian quadrature integration presented in [Müller et al. 2013] [Koschier et al. 2017].

2.3 Haptic Rendering of Solids

Haptic rendering of mesh-based solid objects goes back to seminal work by Zilles and Salisbury [1995]. In this work, authors presented God Object based haptic rendering method. In this method, movement of a god object is constrained up to the outer surface of an object mesh while the corresponding haptic proxy penetrates the outer surface to go inside the object mesh. The difference of the acceleration of god object and haptic proxy generates haptic feedback. Ortega et al. [2007] improved the original god object method which was constrained to three degrees of freedom and extended it to all six degrees of freedom. Another popular method for rendering haptic force is penalty based rendering [Barbič and James 2009] [Otaduy and Lin 2005] [McNeely et al. 1999]. In this method, when two or more colliding objects penetrate each other, we calculate force feedback depending on the depth of penetration among these objects. However, discrete penalty based force rendering is often discontinuous and jerky specially when the contact stiffness is high. Continuous collision detection (CCD) [Tang et al. 2012] alleviates these problems by integrating the force over the contact time intervals between two colliding meshes. Xu and Barbič [2017] developed a method to compute haptic feedback between points and signed distance field using CCD. In our work, we have used the continuous penalty based method for our haptic feedback as it produces smoother force feedback [Xu and Barbič 2017].

2.4 Virtual sculpting

Early virtual sculpting methods [Blanch et al. 2004] [Chen and Sun 2002] use small cubic grid based field to build a rigid object model. These methods are not physically accurate in terms of material behaviour. Moreover, they are computationally expensive while creating a high resolution voxel model. Mass-spring based polygonal mesh deformation is explored in the works [Gunn 2006] [Dachille et al. 1999]. Object surface manipulation using B-spline is presented in [Gao and Gibson 2006]. Jagnow and Dorsey [2002] developed a method to perform virtual sculpting using displacement maps. However, both the B-spline and displacement maps based methods are limited to the interaction with 2D surfaces embedded into 3D space and does not explore the sculpting of volumetric 3D objects. Using low-cost pressure sensors, Callens et al. [2018] designed a tangible surface which can be used like a sculptor tool in virtual environment. In a similar manner, haptic stylus is used for mesh repairing [Turlapati et al. 2021]. However, all these methods discussed above perform sculpting in a strictly geometric way and do not support cutting. Moreover, none of these existing works preserve physical plausibility. Importance of physically realistic virtual sculpting has been investigated by De Goes and James [2017]. Using Kelvinlets, they render the accurate mesh deformation in real-time, but their work lacks the aspects of cutting and haptic feedback. Courtecuisse et al. [2010] presented an algorithm that simulates the cutting of soft

tissue with haptic feedback. Though their method is physically realistic, they use remeshing of tetrahedra for cutting, which is prone to instability. Jeřábková and Kuhlen [2009] presented a XFEM based framework for surgical simulation. However, their work does not employ any safeguard to handle the singular matrix problem which arises when a tetrahedral element gets cut into two highly skewed volumetric parts. We use Is-XFEM to mitigate this problem. Moreover, their work demonstrates the application of their algorithm on simple object meshes and lacks complex sculpting operations on high resolution object meshes. We present sculpting examples on high-resolution complex meshes using a multi-resolution simulation framework. Finally, the haptic feedback provided in all these works are based on discrete collision handling which suffers from jitters and vibration in feedback. We use continuous collision based smooth haptic feedback to tackle this problem. Recently, Mandal et al. [2021a] presented a framework to simulate deformation and wetting of material body. We used this work as baseline of our framework and developed on top of it to build a fully functional sculpting solution that is physically realistic, remeshing-free and render smooth haptic feedback.

3 DEFORMATION & WETTING

System dynamics of deformed object in Lagrange's form [Müller and Gross 2004] can be represented as

$$\begin{aligned} &\mathbf{M}\ddot{\overline{\mathbf{u}}} + \mathbf{f}^{int} = \mathbf{f}^{ext}, \quad \overline{\mathbf{u}} = \left(\mathbf{u}_1^T \dots \mathbf{u}_{n_v}^T\right)^T \\ &\sum_{e=1}^{n_e} \mathbf{m}_e \mathbf{u}_e + \sum_{e=1}^{n_e} \mathbf{f}_e^{int} = \sum_{e=1}^{n_e} \mathbf{f}_e^{ext}, \quad n_e = \#\text{tet} \end{aligned} \tag{1}$$

where $\mathbf{u}, \mathbf{M}, \mathbf{f}^{int}$ and $\mathbf{f}^{ext}$ denote global displacement function, global mass matrix, global internal and external force respectively. Parameters $\mathbf{u}_e, \mathbf{m}_e, \mathbf{f}_e^{int}$ and $\mathbf{f}_e^{ext}$ denote element displacement function, element mass matrix, element internal and external force respectively. Modelling of deformation and wetting of object mesh with haptic feedback have been explored in [Mandal et al. 2021a]. Moreover, in the same work, an algorithm is presented to change object property using water content inside the object mesh via wetting. This is the deformation part of sculpting. Now we have extended their work to sculpting object meshes using cutting along with deformation. Please refer to [Müller and Gross 2004] [Mandal et al. 2021a] for more details.

4 CUTTING OF MESH OBJECTS

In order to simulate cuts on mesh objects without remeshing, we take our inspiration from [Koschier et al. 2017] [Chitalu et al. 2020] to use XFEM. However, these XFEM methods are prone to instability due to the generation singular matrix. We enhanced their model to a Improved stable eXtended Finite Element Method [Wu and Li 2015]. To best of our knowledge, we are first to use Is-XFEM method in any kind of interactive framework.

4.1 Is-XFEM

The solid deformable object to be cut is represented as a tetrahedral mesh. The blade of the cutting brush is a triangular mesh. To begin the cutting process, we decide the path the cut brush follows on the tetrahedral mesh (red dotted lines in Figure 2). To determine the intersection path between tetrahedral mesh and the cut brush mesh, we use the algorithm developed by Möller [Möller 1997]. Each tetrahedron consists of four triangles. Using a traversal algorithm [Baraff et al. 2003] we check if the cut boundary forms a closed loop. Only the closed loop cut planes are considered. Subsequently extra degrees of freedom (DOF) are added to the corresponding nodes (Figure 2).

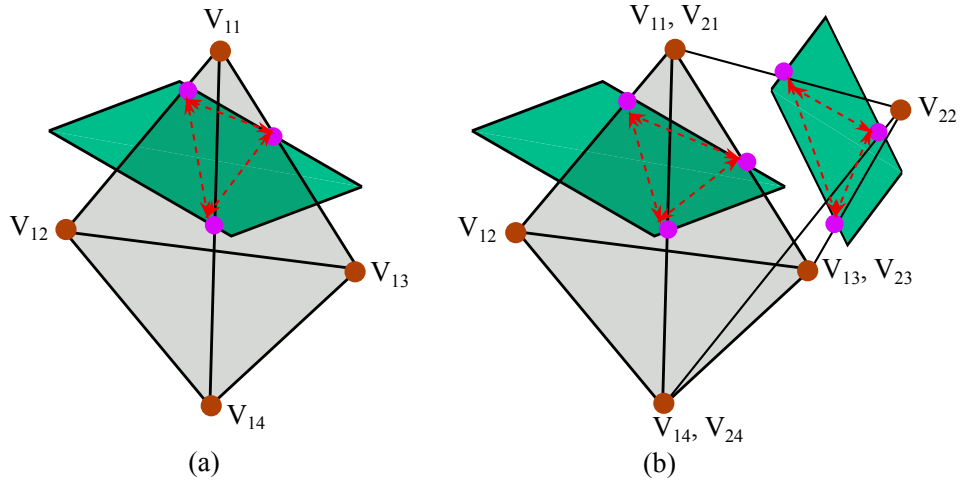

Fig. 2. Red dotted lines indicate the cut path followed by the cut brush on the tetrahedral mesh. Extra degrees of freedom are added to the nodes (shown with red points). The pink points are the intersection points.

If any tetrahedral element Δ_e has m numbers of cut planes, defined as $\Lambda_1, \Lambda_2 \ldots \Lambda_m$, a signed distance function [Koschier et al. 2017] of a point w.r.t. the j^{th} cut plane can be defined as

$$\Phi^j(\zeta) = s(\zeta) \inf_{\zeta^* \in \Lambda^j} ||\zeta - \zeta^*|| \tag{2}$$

where $s : \mathbb{R}^3 \longrightarrow \{-1, 1\}$ denotes the sign of the distance and $\zeta \in \Delta_e$. Better stability is achieved in our algorithm by using an improved enrichment function ψ_i^j as expressed in Equation 3 [Wu and Li 2015]. The enrichment function ψ_i^j represents the enrichment or increase in the number of degrees of freedom of the node i due to the cut j.

$$\psi_i^j(\zeta) = \begin{cases} 1 & \text{No cut} \\ H_s\left(\Phi^j(\zeta)\right) - \left[k\gamma(\zeta) + (1-k)H_s\left(\Phi^j(\zeta_i)\right)\right] & \text{Cut} \end{cases} \tag{3}$$

where $0 < k \leq 1$ and H_s is the Heaviside function. Linear interpolant, γ, is defined as

$$\gamma(\zeta) = \sum_{i=1}^{n_v} \mathbf{N}_i(\zeta) H_s(\zeta_i) \tag{4}$$

The Is-XFEM enrichment function in Equation 3 reduces to XFEM for $k = 0$. We use a small value of $k = 0.1$ for our Is-XFEM simulation. Following the arguments presented in [Wu and Li 2015], it can be proved that Is-XFEM improves the condition number of the system considerably, rendering the system more stable. Using the definition of ψ_i^j from Equation 3, displacement function for Is-XFEM can be as

$$\mathbf{u}^*(\zeta, t) = \sum_{i=1}^{n_v} \mathbf{N}_i(\zeta)\mathbf{u}_i(t) + \sum_{j=1}^{m} \sum_{i=1}^{n_v^{enr}} \psi_i^j(\zeta)\mathbf{N}_i(\zeta)\mathbf{u}_i^j(t) \tag{5}$$

where $\mathbf{u}_i$ denotes the displacement of i^{th} node if no cut is present and $\mathbf{u}_i^j$ represents displacement of same i^{th} node when it gets enriched due to j^{th} cut. n_v^{enr} represents the number of enriched nodes. With this displacement function $\mathbf{u}^*(\zeta, t)$, the per-element mass and external force matrix can be

written as

$$\mathbf{m}_e^* = \int_{\Delta_e} \rho_0 \begin{bmatrix} \mathbf{N_e} & \mathbf{N_e^1} & \dots & \mathbf{N_e^m} \end{bmatrix}^T \begin{bmatrix} \mathbf{N_e} & \mathbf{N_e^1} & \dots & \mathbf{N_e^m} \end{bmatrix} d\zeta \tag{6}$$

$$\mathbf{f}_e^{ext^*} = \int_{\Delta_e} \begin{bmatrix} \mathbf{N_e} & \mathbf{N_e^1} & \dots & \mathbf{N_e^m} \end{bmatrix}^T \mathbf{b} d\zeta, \quad \mathbf{f}_e^{int^*} = \int_{\Delta_e} s_e^*(\bar{\mathbf{u}}, \zeta) d\zeta \tag{7}$$

$$\mathbf{N}_e^j = \begin{bmatrix} \psi_0^j \mathbf{N}_0 \mathbf{I}_3 & \psi_1^j \mathbf{N}_1 \mathbf{I}_3 & \psi_2^j \mathbf{N}_2 \mathbf{I}_3 & \psi_3^j \mathbf{N}_3 \mathbf{I}_3 \end{bmatrix} \tag{8}$$

where $\mathbf{m}_e^*$, s_e^*, $\mathbf{f}_e^{ext^*}$ and $\mathbf{f}_e^{int^*}$ are element mass matrix, elastic element force and external & internal element force vectors respectively for Is-XFEM. While $\mathbf{N}_e$ denotes shape function without any cut present, $\mathbf{N}_e^j$ denotes the updated shape function of tetrahedral element Δ_e corresponding to j^{th} cut. The same system dynamics for FEM deformable objects as defined in Equation 1 is still applicable for Is-XFEM with these updated parameters. The discontinuous integrands that arise in Is-XFEM are solved using a highly accurate Gaussian quadrature rule [Müller et al. 2013] [Koschier et al. 2017].

4.2 Gaussian Quadrature Rule

The integrals that appear in Is-XFEM are solved using the surface and volume Gaussian quadrature integration rules [Müller et al. 2013] [Koschier et al. 2017].

Let the cut plane, defined as $\mathcal{I}$, intersect a tetrahedral element Δ_e and split it into two distinct sub-domains. Let Δ_{e_i} denote the i^{th} sub-domain of Δ_e. Then by Gaussian quadrature integration rule of any function h over that sub-domain can be written as

$$\int_{\Delta_{e_i}} h(\zeta) d\zeta = \int_{\Delta_e} \chi_i h(\zeta) \, d\zeta \approx \sum_{j=1}^{N} w_{i,j} h(\zeta_j) \tag{9}$$

where ζ_j and $w_{i,j}$ are the quadrature points and weights corresponding to i^{th} sub-domain and j^{th} cut. χ_i is characteristic function for i^{th} sub-domain. Quadrature points of a tetrahedron are a set of predefined points, which can be calculated as per [Zhang et al. 2009]. The weights $w_{i,j}$ need to be calculated only once using a particular polynomial basis. Same weights can be reused to integrate any arbitrary bounded function.

4.2.1 Volume quadrature. Let a set of polynomial integrands be $\mathcal{F} = \{f_j\}_{j=1,\dots,M}$ and their anti-derivative be $\mathbf{f}_j$, where $\nabla \cdot \mathbf{f}_j = f_j \ \ \forall j$. Then using the divergence theorem, the quadrature rule to find out the weights for the tetrahedral volume element Δ_e can be written as

$$\begin{bmatrix} f_1(\zeta_1) & \dots & f_1(\zeta_N) \\ \vdots & \ddots & \vdots \\ f_M(\zeta_1) & \dots & f_M(\zeta_N) \end{bmatrix} \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,N} \end{bmatrix} = \begin{bmatrix} \int_{\partial\Delta_e} \chi_i \mathbf{f}_1 \cdot \mathbf{n} d\zeta \\ \vdots \\ \int_{\partial\Delta_e} \chi_i \mathbf{f}_M \cdot \mathbf{n} d\zeta \end{bmatrix} \tag{10}$$

To solve the the right hand side of Equation 10 we need to construct a surface quadrature rule because $\partial\Delta_e$ consists of surface triangular elements and $\mathbf{n}$ denotes the normal to that surface.

4.2.2 Surface quadrature. Following the arguments presented in [Müller et al. 2013], we construct surface quadrature rule, which closely resembles to volume quadrature except the choice of the set of polynomial integrands.

The expression $\int_{\partial\Delta_e} \chi_i \mathbf{f}_j \cdot \mathbf{n} d\zeta$ can be written as

$$\int_{\partial\Delta_e} \chi_i \mathbf{f}_j \cdot \mathbf{n} d\zeta = \underbrace{\int_{\partial\Delta_{e_i}\setminus\mathcal{I}} \mathbf{f}_j \cdot \mathbf{n} d\zeta}_{\text{I}} + \underbrace{\int_{\mathcal{I}} \mathbf{f}_j \cdot \mathbf{n} d\zeta}_{\text{II}} \tag{11}$$

where $\partial\Delta_{e_i}\setminus\mathcal{I}$ is surfaces of tetrahedral element and $\mathcal{I}$ is the cut plane that intersects the tetrahedron.

Following similar arguments presented in volume quadrature, Gaussian quadrature rule can be constructed over the four surfaces of a tetrahedral element, $\partial\Delta_{e_i}\setminus\mathcal{I}$, to evaluate the first part of Equation 11. During evaluating volume quadrature, we noticed that right side of Equation 10 consists of functions integrated over two dimensional area elements. Similarly in surface Gaussian quadrature, the right hand side will consist of one dimensional line elements which can be evaluated easily using standard Gaussian quadrature rule.

In the second part of Equation 11, as the cut plane $\mathcal{I}$ can be any arbitrary nonlinear surface, constructing an accurate surface is computationally expensive. This problem is solved by choosing a divergence-free basis of integrands $\mathcal{F}' = \{\mathbf{f}'_k\}_{k=1,\ldots,K}$. Now the quadrature rule for the cut surface can be written as

$$\begin{bmatrix} \mathbf{f}'_1(\zeta_1)\cdot\mathbf{n}_I(\zeta_1) & \ldots & \mathbf{f}'_1(\zeta_N)\cdot\mathbf{n}_I(\zeta_N) \\ \vdots & \ddots & \vdots \\ \mathbf{f}'_K(\zeta_1)\cdot\mathbf{n}_I(\zeta_1) & \ldots & \mathbf{f}'_K(\zeta_N)\cdot\mathbf{n}_I(\zeta_N) \end{bmatrix} \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,N} \end{bmatrix} = \begin{bmatrix} \int_{\mathcal{I}} \mathbf{f}'_1 \cdot \mathbf{n}_I d\zeta \\ \vdots \\ \int_{\mathcal{I}} \mathbf{f}'_K \cdot \mathbf{n}_I d\zeta \end{bmatrix} \tag{12}$$

where $\mathbf{n}_I$ is the normal to the cut surfaces.

The right hand side of the Equation 12 can be rewritten as

$$\begin{aligned} \int_{\mathcal{I}} \mathbf{f}'_k \cdot \mathbf{n}_I d\zeta &= \int_{\partial\Delta_{e_i}} \mathbf{f}'_k \cdot \mathbf{n} d\zeta - \int_{\partial\Delta_{e_i}\setminus\mathcal{I}} \mathbf{f}'_k \cdot \mathbf{n} d\zeta \\ &= \int_{\partial\Delta_{e_i}} \nabla \cdot \mathbf{f}'_k d\zeta - \int_{\partial\Delta_{e_i}\setminus\mathcal{I}} \mathbf{f}'_k \cdot \mathbf{n} d\zeta \\ &= 0 - \int_{\partial\Delta_{e_i}\setminus\mathcal{I}} \mathbf{f}'_k \cdot \mathbf{n} d\zeta = -\int_{\partial\Delta_e\setminus\mathcal{I}} \chi \mathbf{f}'_k \cdot \mathbf{n} d\zeta \end{aligned} \tag{13}$$

Substituting Equation 13, the integration domain of right side of the Equation 12 consists of the surfaces of tetrahedral element.

4.3 Stability of System Dynamics

While solving the system of Equations 1, we employ the conjugate gradient method. Note that when a tetrahedron gets cut in two parts, the ratio of the volume of these two parts may be highly skewed. This leads to a high condition number for the system, which indicates a loss of stability. In order to improve the stability, we apply a pre-conditioning with the volume ratios of cut tetrahedra and we constrain the system. The discretized version of the system Equation 1 is

$$\underbrace{\left(\mathbf{M} + \Delta t^2 \mathbf{K}\right)}_{\text{A}} \mathbf{v}^{i+1} = \underbrace{\mathbf{M}\mathbf{v}^i - \Delta t \left(\mathbf{K}\mathbf{x}^i + \mathbf{f}_0 + \mathbf{f}_p - \mathbf{f}_{ext}\right)}_{\text{c}} \tag{14}$$

where $\mathbf{f}_0, \mathbf{f}_p$ and $\mathbf{f}_{ext}$ denote initial force, plasticity force and external force respectively. To stabilize these system of equations, we applied two methods as discussed below.

4.3.1 Pre-conditioning. A pre-conditioner matrix $\mathbf{T}$ is applied to our system dynamics Equation 14 as shown in Equation 15.

$$\mathbf{Ax} = \mathbf{c} \implies \mathbf{T}^T\mathbf{ATy} = \mathbf{T}^T\mathbf{c} \implies \mathbf{x} = \mathbf{Ty} \tag{15}$$

The diagonal pre-conditioner matrix is constructed as

$$\mathbf{T} = \text{diag}\left(\frac{1}{\sqrt{v_{1,j}}}, \frac{1}{\sqrt{v_{2,j}}}, \ldots \frac{1}{\sqrt{v_{n,j}}}\right) \tag{16}$$

where $i = 1, 2, \ldots n$ are node numbers and j denotes the cut number. $v_{i,j}$ denotes the volume ratio and is defined as

$$v_{i,j} = \frac{V_{i,j,enr}}{V_{i,supp}} = \frac{\sum_{e \in C_i} \sum_{j=1}^{N} w_{e,j}}{V_{i,supp}} \tag{17}$$

where C_i is the set of all incident tetrahedra on i^{th} node and $w_{e,j}$ is the weights of Gaussian quadrature corresponding to j^{th} cut of tetrahedron e.

4.3.2 Constraining the system of equation. Along with pre-conditioning, to improve the condition number of the system of equations, when the ratio $v_{i,j}$ is too small, we constrain the system of equations like $(\mathbf{A} + \lambda\mathbf{I})\,\mathbf{x} = \mathbf{c}$.

5 HAPTIC RENDERING

We have rendered faithful haptic feedback to implement an interactive virtual sculpting. The haptic interaction process while sculpting an object consists of the following components:

- Continuous Collision Detection (CCD) between the haptic proxy and outer surface of the tetrahedral simulation mesh.
- Continuous penalty based haptic rendering while deforming and cutting the mesh.

All sculpting operations and haptic force feedback for them are performed on volumetric simulation mesh with tetrahedral elements. In order to visually render the sculpt in high quality, we transfer the deformations and all sculpting operations to a higher resolution visualization surface mesh as explained in Section 6.

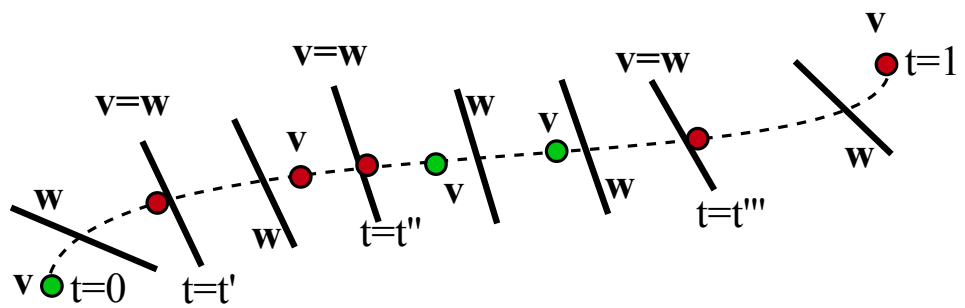

Fig. 3. Three contact times are $t^{'}, t^{''}, t^{'''}$. Penetration intervals are $[t^{'}, t^{''}]$ and $[t^{'''}, 1]$. Time step Δt is normalized to $[0, 1]$.

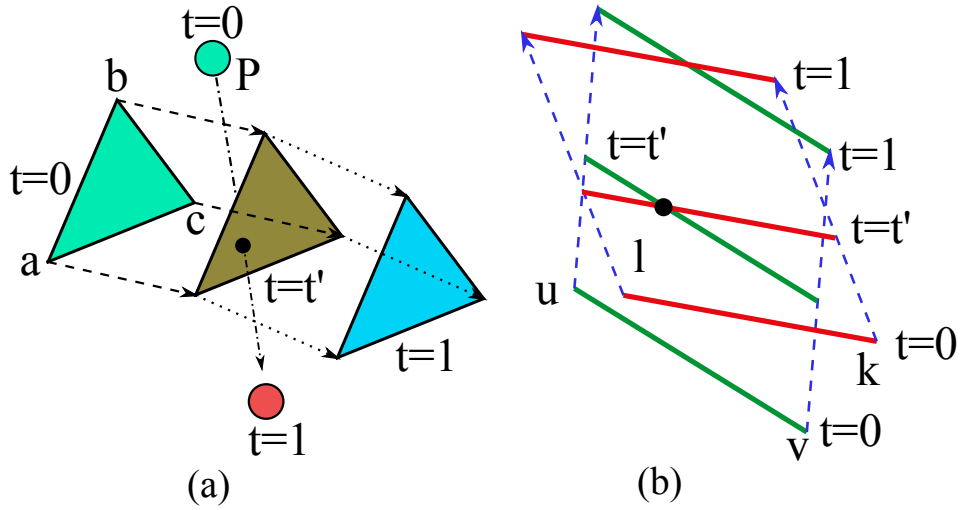

Fig. 4. Continuous collision between (a) vertex-face and (b)edge-edge

5.1 Continuous Collision Detection

Continuous collision detection method is performed between the outer boundary of the tetrahedral object mesh and haptic proxy, both of which consist of triangular primitives. When two triangular face elements collide, we check for two kinds of collisions: vertex-face and edge-edge. We begin by interpolating the positions of each primitive i.e., vertex, edge and face in the simulation time step, Δt, normalized to $[0, 1]$ (as shown in Figure 3). Finally, a 3^{rd} order polynomial equation in t is solved to deduce the number of collisions that occurs during that particular simulation time step. For faster detection of the collisions, we implement Axis Aligned Bounding Box method for each of the triangles along with the non-penetration filter based algorithm presented in [Tang et al. 2010].

5.2 Vertex-face penalty force

Vertex-face collision occurs when any vertex of the haptic proxy collides with any of the triangles of the object mesh boundary or vice-versa. In this case, we calculate a penalty force [Tang et al. 2012] as

$$\mathbf{I}_p^{VF} = k_{vf} \sum_{i=0}^{i<N} \int_{t_a^i}^{t_b^i} \mathbf{n}_t^T \left(\mathbf{p}_t - \mathbf{q}_t\right) \mathbf{n}_t dt \tag{18}$$

where k_{vf} is a scalar stiffness constant. Time intervals $[t_a^i, t_b^i] \in [0, 1]$ are called penetration time intervals which are defined as time segments while the vertex is inside the object mesh (see Figure 4(a)). The parameters i, $\mathbf{n}_t$, $\mathbf{p}_t$ and $\mathbf{q}_t$ denote the number of penetration time intervals, contact normal, position of vertex and contact point on boundary mesh respectively during Δt. Using barycentric coordinates of three vertices of a triangular primitive, the position of the collision point on the triangle can be expressed as $\mathbf{q}_t = w_a \mathbf{a}_t + w_b \mathbf{b}_t + w_c \mathbf{c}_t$. The penalty force $\mathbf{I}_p^{VF}$ is applied to the object mesh. A reaction force of the same magnitude but opposite direction is applied to the proxy.

5.3 Edge-edge penalty force

Similar to vertex-face penalty force, we calculate penalty force $\mathbf{I}_p^{EE}$ if collision occurs between the edge of the haptic proxy mesh and the edge of the simulation mesh boundary of the object ((see

Figure 4(b))). The penalty force is calculated as

$$\mathbf{I}_p^{EE} = k_{ee} \sum_{i=0}^{i<N} \int_{t_a^i}^{t_b^i} \mathbf{n}_{E_t}^T \left(\mathbf{p}_t - \mathbf{q}_t\right) \mathbf{n}_{E_t} dt \tag{19}$$

where k_{ee} is a scalar stiffness constant. Other parameters remain same as before. The position of the collision point on the two edges can again be expressed using barycentric coordinates of two vertices of the edge as $\mathbf{p}_t = w_a\mathbf{a}_t + w_b\mathbf{b}_t$ and $\mathbf{q}_t = w_c\mathbf{c}_t + w_d\mathbf{d}_t$. Like before, the penalty force and the corresponding reaction force are then applied to the object mesh and haptic proxy respectively.

5.4 Haptic Rendering: Cutting a Mesh

We employ the continuous penalty based haptic force feedback rendering method for cutting the object. We consider the collision between the tetrahedral element and the 2D cut brush mesh (see Figure 6). We use a direct coupling method for haptic rendering, i.e., we let the haptic proxy that is a cut brush in this case, penetrate into the object simulation mesh. As the penalty force is directly proportional to the depth of penetration, more the cut brush overlaps object, more the penalty force. When the penalty force increases beyond a certain threshold, the overlapping portion between the cut brush and the tetrahedral mesh is considered cut. For convenient user interaction, the entire cutting simulation runs in two distinct phases repeatedly.

- Marking the cut boundaries with the haptic cut brush. The movement of the object mesh is frozen during this interaction to get a clean cut boundary.
- Letting the object mesh move due to some kind of external force e.g. gravity, without damping kernel, as directed by the Is-XFEM simulation.

This is done because if the object moves considerably while cutting, the penalty force threshold to trigger the cut action is never reached.

6 MULTI-RESOLUTION, MULTI-TIMESCALE SIMULATION FRAMEWORK

For multi-resolution, multi-timescale simulation framework we have extended the method presented in [Mandal et al. 2021a] to cutting. The physical simulation runs on a coarse volumetric cage mesh while for visualization a high resolution surface mesh is used. Any manipulation performed on the simulation mesh gets transferred to the visualization surface mesh using a weight kernel. This sets up the multi-resolution component of our framework. Various sculpting operations transferred across the meshes as follows.

6.1 Transfer of Cut

When the cut brush penetrates inside the tetrahedral simulation mesh as well as the surface mesh, the cut gets initiated (Figure 5 left). Then the cut on the tetrahedral mesh gets projected only inside the secondary mesh for visualization purpose (Figure 5 middle). A user visualizes the cut-mesh without cage (Figure 5 right). Finally at the end of the sculpting operations, both the tetrahedral simulation mesh and surface visualization mesh get affected. User can save or export any of these sculpted meshes, whether simulation or visualization, for using them in other applications.

6.2 Transfer of Deformation and Wetting

We have used the same method as presented in [Mandal et al. 2021a] for transferring the deformation and wetting from simulation mesh to visual mesh.

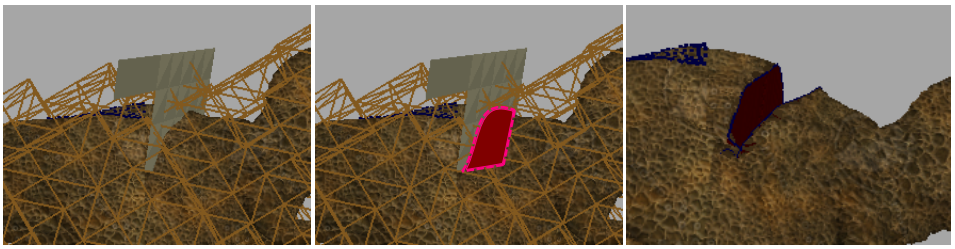

Fig. 5. The cut brush penetrates inside the mesh (left). The cut is projected inside the surface mesh for visualization (middle). The final cut without simulation mesh (right).

6.3 Multi-Timescale Haptic and Visual Feedback

Smooth haptic force feedback requires a minimum of refresh rate of 1000 frames/sec while smooth visual feedback demands a much lower refresh rate of 60 frames/sec. To achieve these disparate requirements, the whole simulation is run in two distinct threads. On one thread, physical simulations along with graphics rendering are performed while other thread is used for rendering haptic feedback. We forcefully kept the haptic thread refreshing at 1000 frames/sec. The visual thread gets updated at around 95 – 150 frames/sec. The synchronization between the two threads is obtained implicitly due to rapid update rate of the haptic thread, instead of using a blocking, explicit synchronization construct. Please refer to [Mandal et al. 2021a] for more details.

7 RESULTS

In this section we present results that help evaluate the performance of our sculpting solution and all its functionalities. First, we present a simulation overview of generating the cut surface for the purpose of visual rendering while cutting the mesh object, even though the original model is not remeshed. Further, we conducted a user study to evaluate the qualitative performance of our solution. A quantitative evaluation of our framework is also presented to affirm that we satisfy real-time interaction constraints.

All the experiments presented here are carried out in a Windows 7 operating system with Intel i7-4770K octa-core processor, 32GB DDR3 RAM, a single Nvidia Geforce GTX Titan GPU with 6 GB of graphics memory and a 6-DOF haptic device from Geomagic Touch. We do not handle self-collision in our work.

7.1 Cutting and Visualizing the Object Mesh

While cutting the object mesh, in order to get proper haptic feedback, we restrict the movement of object. After the object is cut, i.e., the cut surface plane is generated, Is-XFEM simulation is resumed that allows the object to move under external forces. The reason of doing so is that if the object moves while colliding with the cut brush then we will not get any proper overlap between cut brush and object mesh. This closely resembles interaction in real life where we need to hold onto an object firmly in order to cut it with a tool.

When the haptic force threshold is reached, the cut plane boundary gets generated in the overlapping region between cut brush and object mesh. The boundary then gets projected on the surface mesh for proper visualization. This can be seen in Figure 6 (top). Subsequently due to effect of gravity the cut part of the object mesh dangles (shown in Figure 6 (bottom)). In the figure we show the boundary of the cut on the surface mesh with a blue solid line. In Figure 1 (second image from left) we have shown multiple cuts on a Lioness model. The haptic force feedback for cutting the object mesh is shown in Figure 7. It can be seen, the feedback force is low when cut

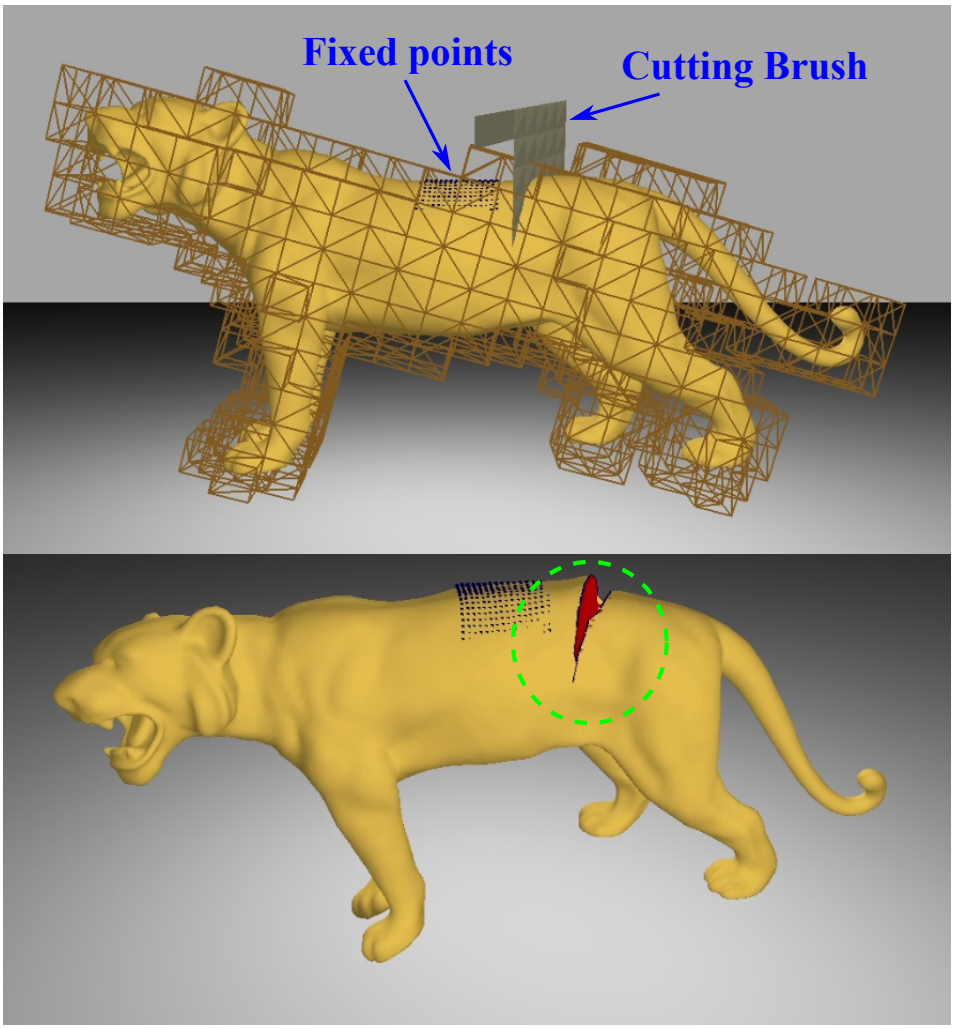

Fig. 6. The cut brush colliding with the object mesh (top). The opening of cut (circled in green) on the mesh (bottom).

brush is not interacting with object mesh. As it penetrates inside the object, marked by red double headed arrows in the figure, the force increases sharply and once the force threshold is crossed, the object gets cut. The force feedback falls to a low value again after that. In the Figure 7 haptic force feedback of four cuts are shown.

For proper visualization, after the penalty force threshold is reached, the intersection of boundary mesh of both the simulation and visualization meshes and the cutting tool mesh is determined first. The intersection plane is then duplicated to generate both sides of the cut. Finally the boundary of both the simulation and visualization meshes are re-meshed with these cut planes embedded into it. This ensures that we always have a closed boundary mesh. The motion of the cut planes are simulated using the Equation 5. Our framework is capable of rendering effects of multiple sculpting operations on the same model. In Figure 1 (third image from left) cutting, deforming and wetting

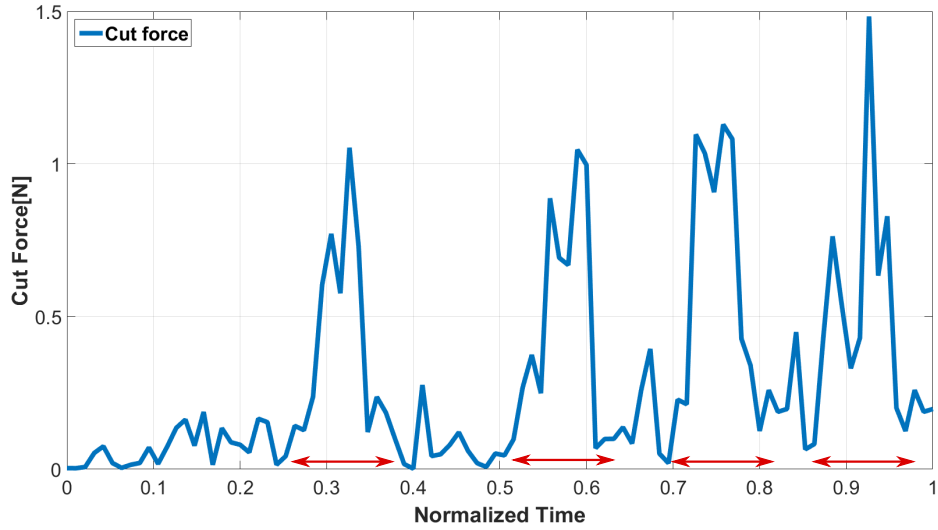

Fig. 7. Illustration of haptic force feedback while cutting the object.

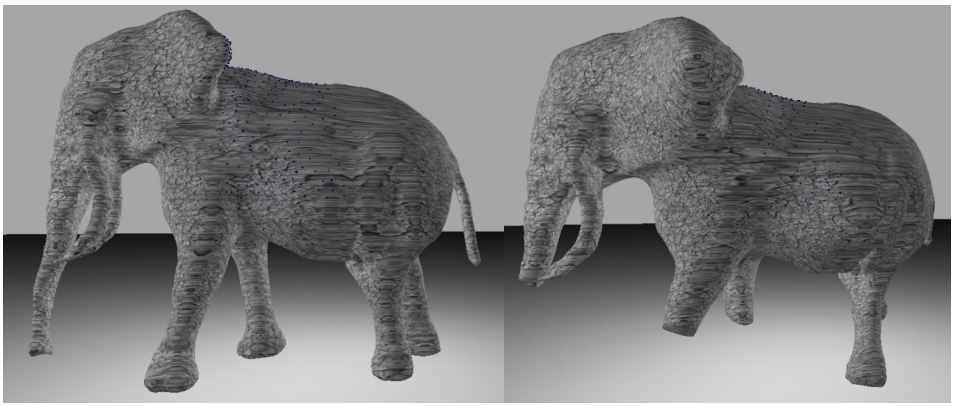

Fig. 8. Original (left) and sculpted (right) Elephant model.

operations are performed on a Lioness model. In Figure 8, Figure 9 and Figure 10 we present an Elephant, a Lioness and a Spider model respectively sculpted by different volunteers. Similarly in Figure 1, the last two images on the right show an original and a sculpted Toy Ninja model. In Figure 11 an amateur volunteer sculpted a scary mask starting from a simple sphere model.

7.1.1 Discussion. Our interactive framework can handle multiple sculpting operations being performed on an object mesh in real time. At the end of the sculpting operations, both the volumetric tetrahedral mesh and visualization triangular mesh get affected. User can save, export or import any of these sculpted meshes, whether tetrahedral or triangular, for using them in the same or other applications. Moreover, after sculpting, our method always generates a water-tight visualization mesh which is convenient for further use in different applications.

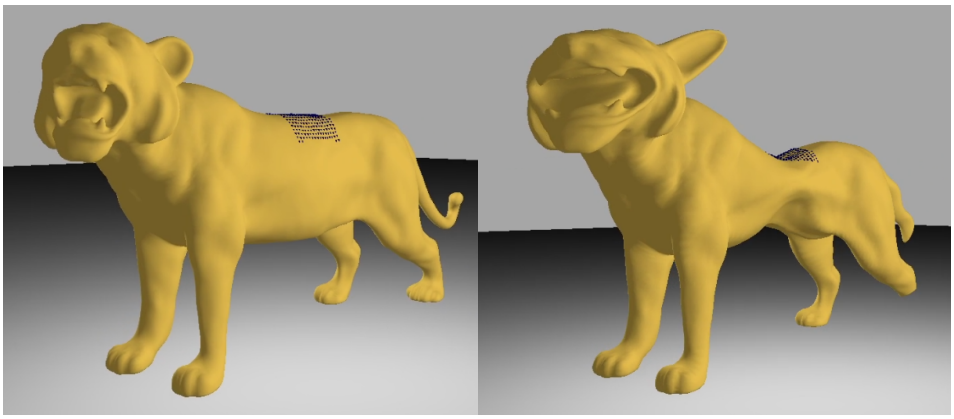

Fig. 9. Original (left) and sculpted (right) Lioness model.

7.2 XFEM vs Is-XFEM

We have compared the results rendered using Is-XFEM and XFEM in Figure 12. We use the method presented in [Koschier et al. 2017] for the XFEM simulation. We perform the exact same cutting operation on a T-Rex model using both Is-XFEM and XFEM. While Is-XFEM remains stable, XFEM becomes unstable. Is-XFEM improves the condition number of the system considerably [Wu and Li 2015], rendering the system more stable.

7.3 User Study

We have performed an extensive user study for all the sculpting operations of our framework, i.e., cutting of mesh along with deformation and wetting from [Mandal et al. 2021a]. Our user study consists of following two experiments.

- **Haptics-Visual Feedback Study** to analyse the effect of haptics and visual feedback in virtual sculpting. We perform an ANOVA analysis for this study.
- **Double Stimulus Comparison Study** to determine how close virtual sculpting experience is compared to the real world sculpting.

7.3.1 Study Subjects. Our user study was conducted with 20 subjects with 3 : 1 male-female ratio. All the subjects are between 22 and 40 years old. All the participants confirmed that they are in sound health both physically or mentally at the time of experiment. None of the participants used a haptic setup before.

7.3.2 Experimental Setup. Our experimental setup is shown in Figure 13. A user sculpting a virtual Lioness model is shown in the figure. Figure 14 visually compares the result of cutting a virtual clay cylinder with a real clay cylinder.

7.3.3 Haptics-Visual Feedback Study. The Analysis of Variance (ANOVA) [Fisher 1954] is a commonly used tool to analyze whether the differences between groups of data are statistically significant. In our work, we use ANOVA to determine the relevance and usefulness of rendering haptics and visual feedback during virtual sculpting. The users are asked to perform all the sculpting operations available in our framework in the following manner and rate their experience on a scale of 1 (very poor) to 5 (very good) for each case.

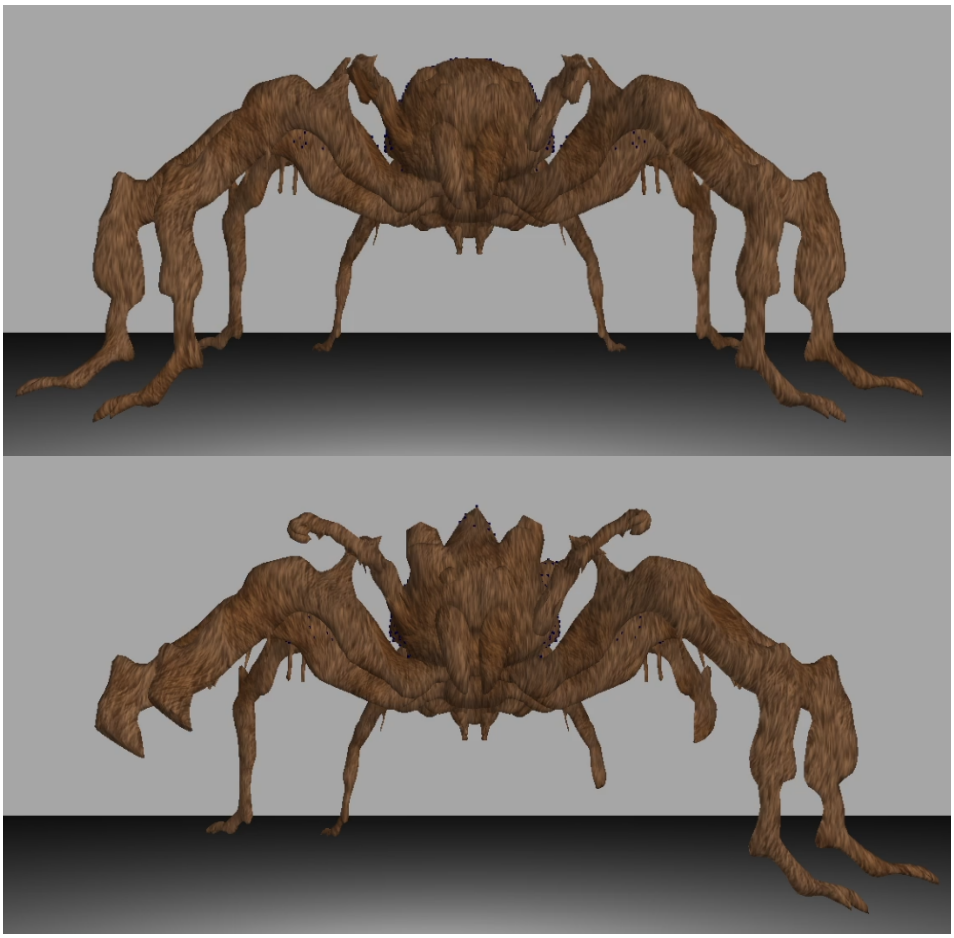

Fig. 10. Original (top) and sculpted (bottom) Spider model.

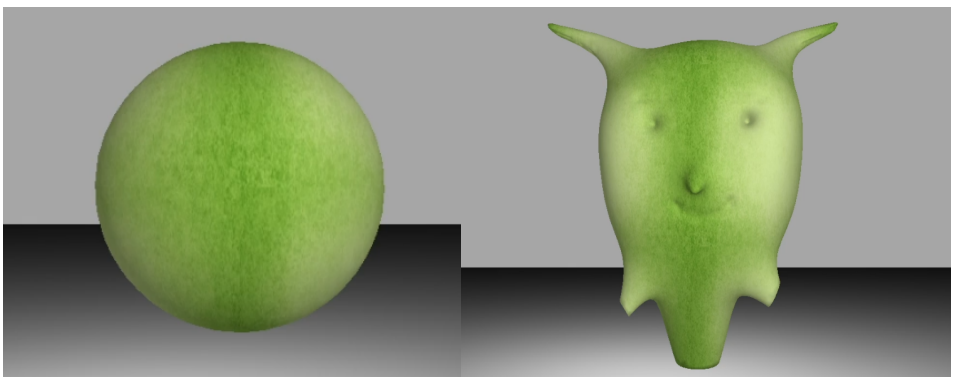

Fig. 11. Original (left) and sculpted (right) Sphere model.

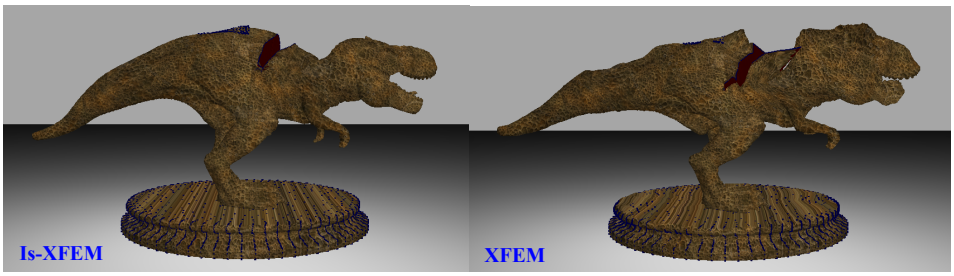

Fig. 12. Cutting operation performed with Is-XFEM (left) and XFEM (right). Simulation rendered with Is-XFEM is more stable compared to XFEM.

- **Strategy 1 - Visual On & Haptics On**: Perform all the sculpting operations available in our framework with both visual and haptic feedback.
- **Strategy 2 - Visual On & Haptics Off**: Perform all the sculpting operations available in our framework with visual feeback but without haptic feedback.
- **Strategy 3 - Visual Off & Haptics On**: Perform all the sculpting operations available in our framework without visual feedback but with haptic feedback.

Note that "without visual feedback" does not denote complete absence of the virtual scene. It means that the effect of the sculpting operations (e.g., deformation, opening up of the cut plane) are not rendered on the high resolution visualization mesh. All the sculpting operations are performed only on the outer cage mesh.

Table 1. p-value for ANOVA study.

Compared strategies	***p-value***
1 vs 2	0.00065
1 vs 3	0.00001

The null hypothesis in ANOVA is that all groups are random samples from the same population, which in our work means that all the sculpting strategies are equally effective. Thus, any observed difference between them is due to random noise. Now, the p-value is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct. So, if the p-value is below a certain threshold, the null hypothesis is rejected. In our work, we use p-value of 0.05, which is a widely accepted choice. The p-value (see Table 1) for Strategy 1 vs Strategy 2 is $0.00065 < 0.05$, which denotes significant difference between the two strategies rejecting the null hypothesis. However, as reported in the top half of Table 2, the higher mean and median score for Strategy 1 compared to Strategy 2 denotes that user experience for virtual sculpting improves when haptic feedback is on. Similarly p-value for Strategy 1 vs Strategy 3 is $0.00001 < 0.05$, which again rejects the null hypothesis. The much higher mean and median score for Strategy 1 compared to Strategy 3 proves that visual feedback is paramount for faithful user experience.

Therefore if we compare the various strategies then based on the results of the above study we can make the following observations.

Strategy 1 vs Strategy 2: According to our data, users rate their virtual sculpting experience most favorably for Strategy 1. It reveals that if the visual feedback remains same, turning on the haptic feedback while sculpting improves user experience.

Table 2. Mean and standard deviation of user feedback.

Parameter	*Mean*	*Median*	*Std*
Visual On, Haptics On	4.71	4.65	0.23
Visual On, Haptics Off	4.27	4.35	0.26
Visual Off, Haptics On	2.79	2.85	0.43
Realism	4.68	4.70	0.34
Visual-haptic sync	4.85	5.00	0.22
Physical consistency	4.51	4.50	0.29

Strategy 1 vs Strategy 3: As the results show Strategy 3 performs very poorly compared to Strategy 1 which asserts the utmost importance of appropriate visual feedback for virtual sculpting. This is not surprising given that in real life also, visual cues and motions are most dominant among all the six sensory senses used for perception.

7.3.4 Double Stimulus Comparison Study. We have performed a double stimulus comparison [Union 2013] study to compare our framework with real world sculpting experience. For this study we use the following parameters.

- **Realism**: The participants are asked to evaluate how realistic their virtual sculpting experience is compared to real world sculpting.
- **Visual-haptic synchronization**: The users are asked to report if they experienced any delay between visual change and haptic force feedback while using our framework.
- **Physical consistency**: Finally, the users are asked to rate how consistent is the visual simulation of our framework with the real world.

In the study, each subject experiences two stimuli one after the other.

- **First stimulus**: First the subjects are asked to mould a ball of clay for free-form sculpting. They are at liberty to use their hands, small sticks and a knife to sculpt a shape of their choice.
- **Second stimulus**: The subjects are then asked to try each of the sculpting tools available in our framework on different object models, including a virtual sphere.

Then the subjects are asked to rate their experience on a scale of 1 (very poor) to 5 (very good) after finishing the experiment.

The mean and standard deviation of the scores of the user feedback opinions for our framework are listed in the bottom half of Table 2. As evident from the Table, the users rate positively for their virtual sculpting experience. Moreover, the low value of standard deviation denotes little difference of opinion among users.

7.4 Quantitative Evaluation

The average interactive frame rates (fps) for cutting and haptic feedback are presented in Table 3. As shown in the Table, in all the cases the update rate for both the visual and haptic threads remain higher than the minimum frame rate (60 for visual and 1000 for haptic) required for smooth user interaction.

8 CONCLUSION AND FUTURE WORK

We present a novel approach for stable interactive virtual sculpting framework enhanced with haptic feedback and physically accurate material simulation. We evaluate the appeal and interactivity of our solution via a user study and a variety of simulation results. One of the major limitations of our

Fig. 13. Set up for our experiment.

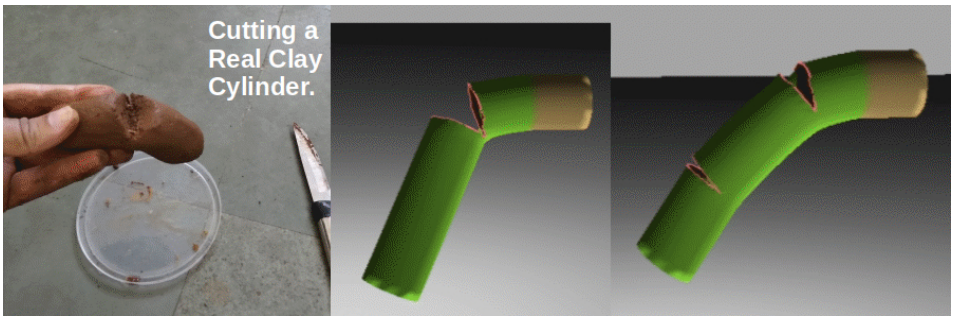

Fig. 14. The three images (from left to right) show cutting a cylinder made of real clay, simulating single and multiple cuts on a virtual clay cylinder.

work is that with the increase in the number of cuts in the object mesh the size of system matrix for Is-XFEM increases. We wish to work in that direction in future.

REFERENCES

Pedro M. A. Areias and Ted Belytschko. 2005. Analysis of three-dimensional crack initiation and propagation using the extended finite element method. *Internat. J. Numer. Methods Engrg.* 63, 5 (2005), 760–788.

David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph.* 22, 3 (July 2003), 862–870.

J. Barbič and D. L. James. 2009. Six-DoF haptic rendering of contact between geometrically complex reduced deformable models: Haptic demo. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. 393–394.

Table 3. Frame rates (fps) for interaction with various models during cutting.

Model	*Cut*	*Haptic*
Lioness	109.7	1002.6
T-Rex	98.1	1001.7
Spider	121.5	1005.5
Elephant	145.8	1009.2
Toy Ninja	163.4	1000.9
Cylinder	117.3	1010.1
Sphere	138.2	1007.4

T. Belytschko and T. Black. 1999. Elastic crack growth in finite elements with minimal remeshing. *Internat. J. Numer. Methods Engrg.* 45, 5 (1999), 601–620.

Renaud Blanch, Eric Ferley, Marie-Paule Cani, and Jean-Dominique Gascuel. 2004. *Non-Realistic Haptic Feedback for Virtual Sculpture.* Technical Report RR-5090. INRIA.

Edouard Callens, Fabien Danieau, Antoine Costes, and Philippe Guillotel. 2018. A Tangible Surface for Digital Sculpting in Virtual Environments. In *Haptics: Science, Technology, and Applications*, Domenico Prattichizzo, Hiroyuki Shinoda, Hong Z. Tan, Emanuele Ruffaldi, and Antonio Frisoli (Eds.). Springer International Publishing, Cham, 157–168.

Hui Chen and Hanqiu Sun. 2002. Real-time Haptic Sculpting in Virtual Volume Space. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '02)*. ACM, 81–88.

Floyd M. Chitalu, Qinghai Miao, Kartic Subr, and Taku Komura. 2020. Displacement-Correlated XFEM for Simulating Brittle Fracture. *Computer Graphics Forum* 39, 2 (2020), 569–583.

Hadrien Courtecuisse, Hoeryong Jung, Jérémie Allard, Christian Duriez, Doo Yong Lee, and Stéphane Cotin. 2010. GPU-based Real-Time Soft Tissue Deformation with Cutting and Haptic Feedback. *Progress in Biophysics and Molecular Biology* 103, 2-3 (Dec. 2010), 159–168.

Frank Dachille, IX, Hong Qin, Arie Kaufman, and Jihad El-Sana. 1999. Haptic Sculpting of Dynamic Surfaces. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics (I3D '99)*. ACM, New York, NY, USA, 103–110.

Fernando De Goes and Doug L. James. 2017. Regularized Kelvinlets: Sculpting Brushes Based on Fundamental Solutions of Elasticity. *ACM Trans. Graph.* 36, 4, Article 40 (July 2017), 11 pages.

Ronald Aylmer Fisher. 1954. *Statistical methods for research workers; 20th ed.* Oliver and Boyd, Edinburgh.

Zhan Gao and Ian Gibson. 2006. Haptic Sculpting of Multi-Resolution B-Spline Surfaces with Shaped Tools. *Comput. Aided Des.* 38, 6 (jun 2006), 661–676.

Chris Gunn. 2006. Collaborative virtual sculpting with haptic feedback. *Virtual Reality* 10, 2 (01 Oct 2006), 73–83.

Philip E. Gustafson and Brian A. Hagler. 1999. Gaussian quadrature rules and numerical examples for strong extensions of mass distribution functions. *J. Comput. Appl. Math.* 105, 1 (1999), 317 – 326.

Robert Jagnow and Julie Dorsey. 2002. Virtual Sculpting with Haptic Displacement Maps. In *Proceedings of the Graphics Interface 2002 Conference, May 27-29, 2002, Calgary, Alberta, Canada.* 125–132.

Lenka Jeřábková and Torsten Kuhlen. 2009. Stable Cutting of Deformable Objects in Virtual Environments Using XFEM. *IEEE Computer Graphics and Applications* 29, 2 (2009), 61–71.

Dan Koschier, Jan Bender, and Nils Thuerey. 2017. Robust eXtended Finite Elements for Complex Cutting of Deformables. *ACM Trans. Graph.* 36, 4 (July 2017), 55:1–55:13.

Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2021a. Physics-based Mesh Deformation with Haptic Feedback and Material Anisotropy. arXiv:cs.GR/2112.04362

Avirup Mandal, Parag Chaudhuri, and Subhasis Chaudhuri. 2021b. Remeshing-Free Graph-Based Finite Element Method for Ductile and Brittle Fracture. *CoRR* abs/2103.14870 (2021). arXiv:2103.14870 https://arxiv.org/abs/2103.14870

William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. 1999. Six Degree-of-freedom Haptic Rendering Using Voxel Sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., 401–408.

Nicolas Moës, John Dolbow, and Ted Belytschko. 1999. A finite element method for crack growth without remeshing. *Internat. J. Numer. Methods Engrg.* 46, 1 (1999), 131–150.

Tomas Möller. 1997. A Fast Triangle-triangle Intersection Test. *J. Graph. Tools* 2, 2 (Nov. 1997), 25–30.

B. Müller, F. Kummer, and M. Oberlack. 2013. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Internat. J. Numer. Methods Engrg.* 96, 8 (2013), 512–528.

Matthias Müller and Markus Gross. 2004. Interactive Virtual Materials. In *Proceedings of Graphics Interface 2004 (GI '04)*. Canadian Human-Computer Communications Society, 239–246.

M. Ortega, S. Redon, and S. Coquillart. 2007. A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties. *IEEE Transactions on Visualization and Computer Graphics* 13, 3 (May 2007), 458–469.

M. A. Otaduy and M. C. Lin. 2005. Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. 247–256.

Min Tang, Dinesh Manocha, Miguel A. Otaduy, and Ruofeng Tong. 2012. Continuous Penalty Forces. *ACM Trans. Graph.* 31, 4 (July 2012), 107:1–107:9.

Min Tang, Dinesh Manocha, and Ruofeng Tong. 2010. Fast Continuous Collision Detection Using Deforming Non-penetration Filters. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*. ACM, 7–13.

Sri Harsha Turlapati, Dino Accoto, and Domenico Campolo. 2021. Haptic Manipulation of 3D Scans for Geometric Feature Enhancement. *Sensors* 21, 8 (2021).

International Telecommunication Union. 2013. ITU-R The double-stimulus continuous quality-scale. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf. Accessed: 2021-04-17.

Jian-Ying Wu and Feng-Bo Li. 2015. An improved stable XFEM (Is-XFEM) with a novel enrichment function for the computational modeling of cohesive cracks. *Computer Methods in Applied Mechanics and Engineering* 295 (2015), 77 – 107.

H. Xu and J. Barbič. 2017. 6-DoF Haptic Rendering Using Continuous Collision Detection between Points and Signed Distance Fields. *IEEE Transactions on Haptics* 10, 2 (April 2017), 151–161.

Linbo Zhang, Tao Cui, and Hui Liu. 2009. A Set of Symmetric Quadrature Rules on Triangles and Tetrahedra. *Journal of Computational Mathematics* 27, 1 (2009), 89–96.

Qi-Zhi Zhu. 2012. On enrichment functions in the extended finite element method. *Internat. J. Numer. Methods Engrg.* 91, 2 (2012), 186–217.

Goangseup Zi and Ted Belytschko. 2003. New crack-tip elements for XFEM and applications to cohesive cracks. *Internat. J. Numer. Methods Engrg.* 57, 15 (2003), 2221–2240.

C. B. Zilles and J. K. Salisbury. 1995. A constraint-based god-object method for haptic display. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, Vol. 3. 146–151.