The Visual Computer (2026) 42:346
https://doi.org/10.1007/s00371-026-04570-3

RESEARCH

VR-Deform: real-time visual and haptic interaction with deformable bodies using XPBD in virtual reality

Mingzhao Zhou[1] · Nadine Aburumman[1] · Zhenxing Li[2] · Roope Raisamo[2]

Received: 5 May 2026 / Accepted: 31 May 2026

Abstract
In this paper, we simulate a lifelike interaction with deformable bodies in virtual reality (VR) and provide force haptic feedback. Our method achieves physically compelling responses and stable feedback, which are essential for natural interaction with soft virtual components. Existing systems often suffer from instability or latency and rely on rigid body or primitive soft body models, which limit their application. We present a real-time VR framework that employs Extended Position Based Dynamics (XPBD) to enable deformable body simulation alongside stable haptic feedback. In our scenes, the proposed system achieves up to 96.3 FPS visual rendering on the VR system's head-mounted display (HMD) while maintaining stability without frame loss or lag. Designed for lightweight deployment on consumer-grade hardware, our implementation integrates a haptic glove to capture finger motion and deliver multi-point tactile feedback. The simulation uses a unified solver that incorporates a set of position-based constraints and accounts for haptic force feedback. Our method supports scenes of up to 500k vertices and requires no external tracking, relying solely on the HMD and controllers for positional input. We conducted a user study in which participants interacted with the system and rated deformations and haptic feedback. Results showed the visuals and the haptic responses were convincing and plausible. We demonstrate that our framework enables immersive, stable, deformable body interaction with synchronised visual and haptic feedback, supporting applications in medical simulation, physical skill training, and entertainment.

Keywords Virtual reality (VR) · Haptic · Deformable body simulation · Extended position-based dynamics (XPBD)

1 Introduction

Deformable bodies play a vital role in everyday real-world interaction, whether manipulating textiles, biological tissues, or flexible components in manufacturing settings. While immersive technologies such as virtual reality (VR) continue to advance, various applications such as medical training, industrial simulation, and gaming, increasingly demand natural interaction with deformable virtual elements [2]. Due to their simplicity and ease of implementation, rigid body interactions remain dominant in VR, as most systems rely on rigid body dynamics for manipulation. However, rigid bodies are only sufficient for basic interactions and fail to capture the elastic behaviour of deformable materials [34]. Deformable body simulations offer more realistic interactions but are still limited in practice because of their computational cost [11]. Moreover, the mathematical complexity of such simulations, particularly in real-time scenarios, poses a significant challenge, and the addition of force haptic feedback further constrains computational speed [32].

Nonetheless, in VR applications, tactile and force feedback are essential for creating believable virtual interactions, as haptic forces combined with visual input enhance presence and make deformable body manipulation feel compelling. However, delivering stable haptic feedback with low latency during deformable body interaction remains largely unexplored due to mismatches update rates between visual and haptic rendering, aggravated by the high computational demands of deformation simulations. Addressing these technical issues is important to maintain immersion in VR experiences.

In this paper, we propose a novel real-time framework that enables stable simulation and believable deformable

✉ Nadine Aburumman
Nadine.Aburumman@brunel.ac.uk

Mingzhao Zhou
Mingzhao.Zhou@brunel.ac.uk

1 Brunel University London, Uxbridge, UK

2 Tampere University, Tampere, Finland

Published online: 19 June 2026

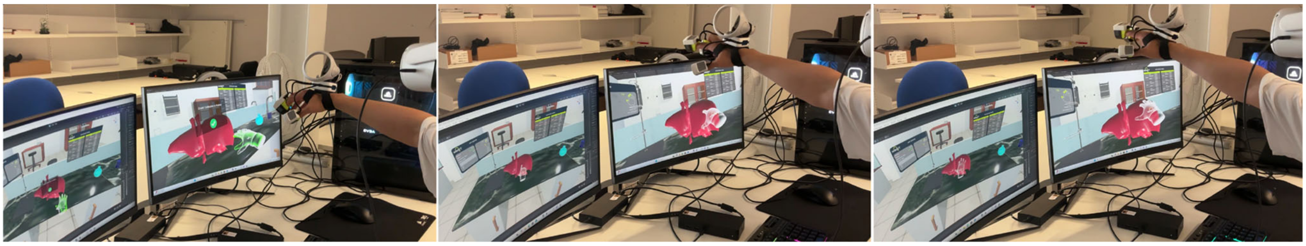

Fig. 1 User interacting with a deformable liver model in VR using an HMD headset and a force-based wearable haptic glove. The system delivers real-time visual rendering and haptic force feedback, enabling the user to perceive soft tissue deformation and haptic force during manipulation

body manipulation with tactile haptic feedback in VR (see Fig. 1). Our method provides synchronised visual and haptic feedback at low latency by employing Extended Position Based Dynamics (XPBD) [21]. This constraint-based simulation method enables computing particle positions directly instead of calculating velocities and subsequently updating positions, offering a controllable and physically compelling interaction. This configuration enables timely tactile feedback via a wearable haptic glove while maintaining immersive visuals. The deformable body simulation is driven by a constraint-based scheme, where contact handling is used to compute haptic forces that are transmitted to the glove in real-time. By aligning visual and tactile feedback, the system enables users to physically feel what they see, enhancing realism and immersion during interaction with deformable virtual objects.

This work makes the following contributions:

- *A real-time framework for visual and haptic interaction with deformable objects in VR using XPBD.*
- *A set of geometric constraints is formulated to compute and deliver haptic feedback via a wearable device.*
- *A deployable implementation in Unity3D that operates on consumer-grade hardware without external tracking systems, where we validated our system through a user study.*

2 Related work

In this section, we review the relevant literature in two main areas: Deformable body simulation and haptic rendering techniques. For a comprehensive overview of deformable body simulation, we refer the reader to the excellent survey by Zhang et al. [42]. Moreover, for haptic rendering techniques, we direct readers to the detailed review presented in [18, 36].

2.1 Deformable body simulation

Physics-based simulation of soft deformable materials remains a core challenge for interactive graphics and immersive applications. Finite Element Methods (FEM) have been widely used to model non-linear elastic behaviour in medical simulation and robotics [12, 19]. In soft robotics, real-time control of deformable systems using FEM was demonstrated by Duriez et al. [6], and later extended to surgical training scenarios, including cataract simulation [5]. To address the limitations related to mesh quality, Chen et al. [4] introduced a mesh-free FEM formulation more tolerant to topological changes and distortion. Despite these advances, FEM-based methods remain computationally demanding [40]. An alternative technique is the Material Point Method (MPM), which combines Lagrangian particles with an Eulerian background grid to simulate large deformation dynamics. Since its introduction to Computer Graphics [38], MPM has been adopted for handling challenging scenarios involving severe frequent topological change. Extensions to the method have introduced a range of plasticity models to support realistic material behaviour [10]. While the use of implicit integration schemes is used to offer improved stability under larger time steps, they also give rise to large non-linear systems that are computationally intensive to solve [16]. Recent work in physics-based animation has increasingly prioritised robustness, stability, and computational efficiency over strict physical realism, particularly in applications requiring interactive performance. Therefore, mass-spring models (MSMs) are commonly adopted in interactive scenarios due to their simplicity and speed. However, MSMs exhibit limited energy conservation and lack accurate correspondence with stress–strain relationships, reducing their fidelity in simulation [37].

Position-based approaches then came to introduce an easy and efficient solver for real-time and interactive applications. Shape matching [27] simplifies deformation by aligning particles with their goal configurations, but is limited in its ability to preserve volume. Position Based Dynamics (PBD) [28] improves simulation stability by directly enforcing positional constraints, allowing large time steps and avoiding force integration. Macklin et al. [21] introduced XPBD, a compliant constraint formulation that enhances stability while enabling physically plausible deformation under interactive manipulation. Later, XPBD Small Steps [22] reduces the simulation step size by dividing it by the iteration count in order to improve the convergence of the core algorithm. Recently, GPU-accelerated implementations [33] have further extended the scalability of position-based methods, and their integration into game engines such as Unity3D has been demonstrated [24].

In this work, we employ XPBD as the core simulation engine for its balance between control, efficiency, and visually compelling deformation. Although XPBD is increasingly used for visual simulation, its application to stable haptic feedback in VR remains untackled. This work addresses that gap by integrating XPBD into a real-time visual and force haptic interaction framework.

2.2 Haptic rendering

Haptic interaction enables users to perceive force, material properties, and shape during virtual object manipulation. In VR systems, natural hand-based input is typically captured through kinematics-based methods using controllers, gloves, or tracking systems [20, 41]. These devices stream real-time data such as position, orientation, and joint angles, which are used to trigger interaction events in the virtual environment [43]. While vibrotactile feedback is widely adopted due to its low cost and ease of integration [13, 23], it lacks the capacity to convey sustained or directional forces. Force feedback systems address this limitation by providing mechanical resistance, enabling more realistic and physically driven interactions. To support such realism, force feedback devices must transmit continuous force information that respond to user input in real-time. Recent advances include wearable solutions with multi-degree-of-freedom actuation designed for fine motor control. For example, Batik et al. [1] and Kudry et al. [15] developed haptic systems capable of supporting precise manipulation tasks, while Lim et al. [17] showed that force feedback can enhance user performance during tool-based interactions. Central to the functionality of these systems is collision detection. Although detecting rigid-body contacts is well-established, interactions with deformable objects are more complex due to continuous changes in geometry during manipulation [24]. Guoxin et al. [7] approached this challenge by framing collision detection as a geometry-constrained optimisation problem, enabling robust contact handling. Similarly, Mesit et al. [25] proposed a multilayer spring-boundary model that accounts for both external and self-collisions, and Yang et al. [39] introduced a hybrid framework combining Lagrangian particles and Eulerian grids to improve efficiency in contact simulation [8].

Our system integrates a wearable haptic input with a deformable simulation pipeline to enable real-time visual-haptic interaction. Contact with deformable virtual objects is detected using a geometry-based collision model that continuously adapts to changes in object shape. Various constraints are formulated and incorporated into the XPBD solver. These constraints simultaneously drive mesh deformation and are used to compute force feedback signals rendered through the haptic glove. This tight coupling between visual and haptic responses allows users to perceive resistance during interaction with deformable bodies.

3 Overview

We present a framework for simulating visual and haptic interactions with deformable bodies in real-time using Extended Position Based Dynamics (XPBD) [21] while integrating force-based wearable haptic gloves. XPBD extends traditional PBD by introducing constraint compliance, enabling the simulation of soft body behaviour with improved physical fidelity, especially under stiff constraints or large time steps. These properties make XPBD well-suited for applications requiring haptic feedback. In our framework, we employ XPBD to model deformable bodies as particle-based systems governed by constraints, including stretching, volume preservation, shape matching, strain and bending. Constraints are enforced using a Gauss–Seidel solver, with per-constraint compliance parameters and accumulated Lagrange multipliers. This compliant formulation leads to well-defined estimates of constraint forces, enabling plausible force feedback and robust stiffness control. Additionally, structuring the particle system with tetrahedral elements allows for lifelike volumetric deformation. In our framework, users interact with the simulated deformable body using a *WEART TouchDIVER G1* haptic glove [35], which provides position input to the simulation and receives force feedback in return. The system computes interaction forces based on the constraint response of particles, and delivers this feedback in real-time to enhance the realism of the manipulation. The deformed configuration of the particle system is then used to update the high-resolution surface mesh, enabling compelling representation of the soft body. Tetrahedral meshes are generated from the surface meshes using the external tool *TetGen* [31], where tetrahedral are used for simulation. The surface mesh is updated continuously during the simulation to reflect real-time deformation and the final visual rendering.

A schematic overview of the system pipeline is shown in Fig. 2, illustrating the flow of data between user input, the XPBD simulation core and haptic feedback computation, in which this integration of XPBD with real-time haptic and graphical feedback supports intuitive and plausible interaction with deformable objects in immersive environments.

4 Deformable body simulation

We simulate deformable objects as tetrahedral meshes that consist of particles, constrained by geometric relationships.

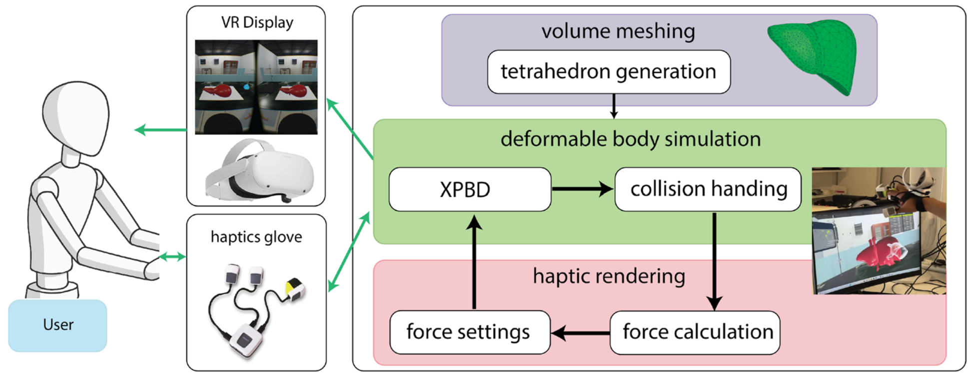

Fig. 2 Illustrates the integration of an HMD (VR headset) and a wearable force haptic glove. Our system captures deformation and collisions, with force calculations enabling haptic feedback in real-time, where volumetric meshes are generated for deformable bodies simulation using the XPBD scheme

To ensure stability and realism under varying simulation rates, while maintaining responsive haptic interaction during deformation, we adopt XPBD, which enhances traditional PBD by incorporating a well-defined concept of elastic potential energy. In general, constraints are functions of position and orientation variables, linear and angular velocities, and their derivatives. They impose kinematic restrictions in the form of equations, equalities, and inequalities that constrain the relative motion of bodies. The aim of using these constraints is to model both the internal behaviour of the simulated body and its interactions with external contacts and forces.

In XPBD [21], each constraint is associated with a compliance parameter $\alpha \geq 0$, which defines the material's flexibility. The vector $\mathbf{p}$ represents the positions of a set of n particles, and $C_j(\mathbf{p})$ is a function that represents the j'th constraint among a set of constraints that work on the particles. Unlike regular PBD, which directly adjusts particle positions based on a stiffness value, XPBD computes force-driven *Lagrange multipliers* (λ_j) for each constraint, at every iteration, that guide more physically accurate particle position updates. For a detailed introduction of PBD, we refer the reader to Müller et al. [28].

The core of the XPBD technique is captured by the following equation, which updates the Lagrange multiplier λ_j for each constraint:

$$\Delta\lambda_j = \frac{-C_j(\mathbf{p}) - \tilde{\alpha}_j\lambda_j}{\nabla C_j(\mathbf{p})\mathbf{M}^{-1}\nabla C_j(\mathbf{p})^T + \tilde{\alpha}_j}, \tag{1}$$

where M is a diagonal matrix of which the diagonal entries represent the particle masses and $\nabla C_j(\mathbf{p})$ is the gradient of the constraint C_j evaluated at the current predicted particle positions $\mathbf{p}$. To account for the time discretisation used in the simulation, XPBD introduces a scaled compliance term $\tilde{\alpha}_j$, defined as:

$$\tilde{\alpha}_j = \frac{\alpha_j}{\Delta t^2}, \tag{2}$$

where Δt is the time step size and α_j introduces a physically meaningful interpretation of stiffness, decoupling it from solver settings and enabling consistent behaviour across varying time steps and iteration counts. Furthermore, the accumulated multiplier λ_j serves as an estimate of the internal constraint force, which we use to drive haptic feedback.

In XPBD, constraint corrections are applied in position space using the following update rule:

$$\Delta\mathbf{p} = \mathbf{M}^{-1}\nabla C_j(\mathbf{p})^T\Delta\lambda_j. \tag{3}$$

Thus, $\Delta\mathbf{p}$ is the position correction for the particles involved in constraint C_j, and $\Delta\lambda_j$ is the Lagrange multiplier update for the constraint, as defined above.

The particle position prediction $\mathbf{p}$ for the next time step of the simulation is then modified by $\Delta(\mathbf{p})$:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}. \tag{4}$$

Moreover, damping is incorporated via a Rayleigh dissipation model:

$$\lambda_{\text{damp}} = -\tilde{\beta}\nabla C_j(\mathbf{p})\mathbf{v}, \tag{5}$$

where $\tilde{\beta}$ is the time-step-scaled damping parameter and $\mathbf{v}$ the particle velocity. The total constraint force is then:

$$\lambda = \lambda_{\text{elastic}} + \lambda_{\text{damp}}, \tag{6}$$

where $\lambda_{\text{elastic}} = -\tilde{\alpha}^{-1}\nabla C_j(\mathbf{p})$. Therefore, $\tilde{\alpha}_j$ is a block diagonal compliance matrix corresponding to the inverse of the stiffness constants.

We summarise the XPBD simulation process for a deformable body in Algorithm 1.

4.1 Stretch constraint

To model tension and compression behaviour in elastic materials, we use stretch constraints that resist changes in edge length. Stretch constraints enforce a target distance between

Algorithm 1 XPBD Simulation for Deformable Body

for each particle i **do**
  Initialise: $\mathbf{x}_i \leftarrow \mathbf{x}_0$, $\mathbf{v}_i \leftarrow \mathbf{v}_0$, $w_i \leftarrow 1/m_i$
  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \cdot w_i \cdot \mathbf{f}_{\text{gravity}}(\mathbf{x}_i)$
end for
while simulating **do**
  for each particle i **do**
    Predict position: $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t\, \mathbf{v}_i + \Delta t^2\, w_i \cdot \mathbf{f}_{\text{ext}}(\mathbf{x}_i)$
  end for
  Initialise multipliers $\lambda_0 \leftarrow 0$
  Perform collision detection using $\mathbf{x}_i$, $\mathbf{v}_i$
  Generate constraints from current predicted positions
  for each solver iteration **do**
    for each constraint C_j **do**
      Compute $\Delta\lambda_j$ and update λ_j using Eq. 1
      Apply position correction $\Delta\mathbf{p}$ to related particles using Eq.3 and Eq.4
    end for
  end for
  for each particle i **do**
    Update: $\mathbf{x}_i \leftarrow \mathbf{p}_i$
    Update: $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/\Delta t$
  end for
end while

two particles. Given current particle positions $\mathbf{p}_1$ and $\mathbf{p}_2$, and l_0 representing the rest length associated to a stretch constraint C_{stretch}, the constraint is defined as:

$$C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\| - l_0. \tag{7}$$

Denote by $\nabla_{\mathbf{p}_i} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2)$ the gradient of the constraint restricted to the coordinates of particle $\mathbf{p}_i$, for $i \in \{1, 2\}$. Then we have

$$\nabla_{\mathbf{p}_1} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = \frac{\mathbf{p}_1 - \mathbf{p}_2}{\|\mathbf{p}_1 - \mathbf{p}_2\|} \tag{8}$$

$$\nabla_{\mathbf{p}_2} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = -\nabla_{\mathbf{p}_1} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2). \tag{9}$$

Using XPBD, the constraint update step results in the following Lagrange multiplier increment:

$$\Delta\lambda_{\text{stretch}} = \frac{-C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) - \tilde{\alpha}_{\text{stretch}}\lambda_{\text{stretch}}}{w_1 + w_2 + \tilde{\alpha}_{\text{stretch}}}, \tag{10}$$

where λ_{stretch} is the Lagrange multiplier associated with the stretch constraint, and $\tilde{\alpha}_{\text{stretch}}$ is the compliance of the constraint. We then apply the position correction to the two particles:

$$\Delta\mathbf{p}_1 = -w_1 \Delta\lambda_{\text{stretch}} \nabla_{\mathbf{p}_1} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2), \tag{11}$$

$$\Delta\mathbf{p}_2 = w_2 \Delta\lambda_{\text{stretch}} \nabla_{\mathbf{p}_2} C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2), \tag{12}$$

where w_1 and w_2 are the inverse masses for particles $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively. Thus:

$$\Delta\lambda = \frac{-(\|\mathbf{p}_1 - \mathbf{p}_2\| - l_0) - \tilde{\alpha}_{\text{stretch}}\lambda_{\text{stretch}}}{w_1 + w_2 + \tilde{\alpha}_{\text{stretch}}} \tag{13}$$

$$\Delta\mathbf{p}_1 = -w_1 \cdot \Delta\lambda \cdot \mathbf{n} \tag{14}$$

$$\Delta\mathbf{p}_2 = +w_2 \cdot \Delta\lambda \cdot \mathbf{n} \tag{15}$$

$\mathbf{n}$ is the normal.

4.2 Tetrahedral volume constraint

To model volumetric elasticity and resist uniform compression and expansion, we use volume constraints. Volume constraints ensure that deformable elements, such as soft tissues, do not collapse or expand unnaturally during simulation, thus minimising volume loss. Consider a tetrahedral element defined by the current position of the particles $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$. The signed volume of the tetrahedron is given by:

$$V = \frac{1}{6}\left[(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\right] \cdot (\mathbf{p}_4 - \mathbf{p}_1). \tag{16}$$

Let V_0 denote the rest volume. Then, the volume constraint is defined as:

$$C_{\text{vol}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = V - V_0. \tag{17}$$

Where $\nabla_{\mathbf{p}_i} C_{\text{vol}}$ denotes the gradient of the constraint with respect to particle $\mathbf{p}_i$, for $i \in \{1, 2, 3, 4\}$. The gradients are given by:

$$\nabla_{\mathbf{p}_1} C_{\text{vol}} = \frac{1}{6}\left[(\mathbf{p}_2 - \mathbf{p}_3) \times (\mathbf{p}_4 - \mathbf{p}_3)\right], \tag{18}$$

$$\nabla_{\mathbf{p}_2} C_{\text{vol}} = \frac{1}{6}\left[(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)\right], \tag{19}$$

$$\nabla_{\mathbf{p}_3} C_{\text{vol}} = \frac{1}{6}\left[(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_4 - \mathbf{p}_2)\right], \tag{20}$$

$$\nabla_{\mathbf{p}_4} C_{\text{vol}} = \frac{1}{6}\left[(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\right]. \tag{21}$$

Based on the XPBD formulation, the Lagrange multiplier update is computed as:

$$\Delta\lambda_{\text{vol}} = \frac{-C_{\text{vol}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) - \tilde{\alpha}_{\text{vol}}\lambda_{\text{vol}}}{\sum_{i=1}^{4} w_i \|\nabla_{\mathbf{p}_i} C_{\text{vol}}\|^2 + \tilde{\alpha}_{\text{vol}}}, \tag{22}$$

where λ_{vol} is the Lagrange multiplier associated with the volume constraint, and $\tilde{\alpha}_{\text{vol}} = \alpha_{\text{vol}}/h^2$ is the scaled compliance. Therefore, the position updates for each particle are then given by:

$$\Delta\mathbf{p}_i = -w_i \cdot \Delta\lambda_{\text{vol}} \cdot \nabla_{\mathbf{p}_i} C_{\text{vol}}, \quad i \in \{1, 2, 3, 4\}, \tag{23}$$

where the position correction moves each particle in the direction that reduces the volume constraint violation, scaled by the inverse mass and the local gradient contribution.

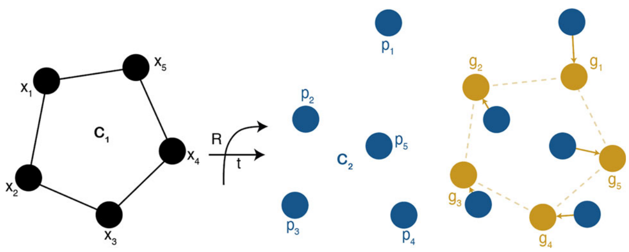

Fig. 3 An illustration of shape matching: The initial shape $\mathbf{x}_i$ is aligned to the target shape $\mathbf{p}_i$ via rotation $\mathbf{R}$ and translation $\mathbf{t}$ to produce the goal shape $\mathbf{g}_i$

4.3 Dihedral bending constraint

To model resistance to folding and changes in curvature, we use dihedral bending constraints. These constraints help keep the angles between connected triangles consistent. For a triangle made of particles $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, and a fourth particle $\mathbf{p}_4$ that forms a bending quad, we calculate the rest distance R_ℓ from $\mathbf{p}_4$ to the centre of the triangle. Therefore, we can define bending constraints as:

$$C_{\text{bend}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) - \varphi_0, \tag{24}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are the unit normals of the adjacent triangles $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ and $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$, respectively, and φ_0 is the rest (initial) dihedral angle between them. The arccos is the inverse function of the cosine function.

The constraint function can also be expressed in terms of cross products:

$$\arccos\left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|} \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)\|}\right) - \varphi_0$$

The gradient of this constraint with respect to each particle position can be derived analytically, as shown in Appendix A of [28]. Then, to project this constraint using XPBD, we then compute $\Delta\lambda$ and the position correction $\Delta\mathbf{p}_i$ for each particle according to Eqs. 1 and 3, respectively.

4.4 Shape matching constraint

To model elastic strain energy and global shape preservation, we use shape matching constraints. These constraints are defined within our framework to enforce a rigid transformation between the rest configuration and the current particles' positions (see Fig. 3).

Given two sets of points, $\mathbf{x}_i$ and $\mathbf{p}_i$, our objective is to find the rotation matrix $\mathbf{R}$ and translation vectors $\mathbf{t}$ and $\mathbf{t}_0$ that minimise the following expression:

$$\sum_i w_i(\mathbf{R}(\mathbf{x}_i - \mathbf{t}_0) + \mathbf{t} - \mathbf{p}_i)^2, \tag{25}$$

where w_i denotes the weight associated with each point, and the optimal translation vectors correspond to the centres of mass of the rest shape $\mathbf{c}_1$ and the deformed shape $\mathbf{c}_2$, respectively. The optimal choices for $\mathbf{t}_0$ and $\mathbf{t}$ can de derived to be $\sum_i m_i\mathbf{x}_i / \sum_i m_i$ and $\sum_i m_i\mathbf{p}_i / \sum_i m_i$, respectively, relatively straightforwardly. We then find $\mathbf{R}$ by first finding a matrix $\mathbf{A}$ substituted in place of $\mathbf{R}$, that we allow to be any linear transformation (not necessarily a rotation one) that maps the rest configuration of a particle system to its deformed state, see [27] for more details. This yields the optimal choice.

$$\mathbf{A} = \left(\sum_i m_i\left((\mathbf{p}_i - \mathbf{c}_2)(\mathbf{x}_i - \mathbf{c}_1)^T\right)\right) \left(\sum_i m_i\left((\mathbf{x}_i - \mathbf{c}_1)(\mathbf{x}_i - \mathbf{c}_1)^T\right)\right)^{-1} \tag{26}$$

Thus, the goal position is then computed as follows: The polar decomposition of matrix $\mathbf{A}$ yields the rotation matrix $\mathbf{R}$, and the target position is given by:

$$\mathbf{g}_i = \mathbf{c}_2 + \mathbf{R}(\mathbf{x}_i - \mathbf{c}_1). \tag{27}$$

Therefore, we can define the shape matching constraints as:

$$C_{\text{matching}}(\mathbf{p}_i) = \mathbf{p}_i - \mathbf{g}_i \tag{28}$$

The corrections are applied using XPBD's projection equations Eqs. 1 and 3.

4.5 Tetrahedral strain constraint

To model continuum elastic behaviour and shear deformation, we use tetrahedral strain constraints based on the Green–Lagrange strain tensor within our framework. These constraints are suitable for soft body modelling and we applied them to triangles and tetrahedral meshes. The strain constraints model elastic deformation, where a tetrahedron is defined by four particles $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$. This constraint involves all four particles and can drive the configuration to a state in which the components of Green's strain tensor assume given values [26]. This can be formulated in terms of the positions of the four particles adjacent to the tetrahedron. We define the matrix $\mathbf{D}_{\text{rest}}$, which contains the edge vectors from $\mathbf{p}_1$ to the other points in the undeformed state

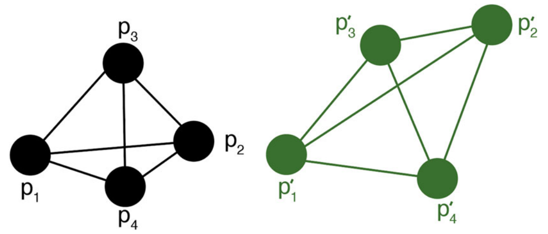

Fig. 4 Tetrahedral strain constraint illustrated on undeformed (black, left) and deformed (green, right) configurations. The constraint operates on four particles $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$, capturing elastic deformation via Green's strain tensor. For strain calculation, $\mathbf{p}_1$ and $\mathbf{p}'_1$ are assumed to be at the origin

(rest). Since translation does not contribute to strain, we can assume that $\mathbf{p}_1$ and $\mathbf{p}'_1$ are zero. See Fig. 4.

$$\mathbf{D}_{\text{rest}} = [\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4] \tag{29}$$

and the deformed shape matrix is defined as:

$$\mathbf{D}_{\text{def}} = [\mathbf{p}'_2, \mathbf{p}'_3, \mathbf{p}'_4], \tag{30}$$

where $\mathbf{p}'_1, \mathbf{p}'_2, \mathbf{p}'_3, \mathbf{p}'_4$ are the particles' positions after deformation.

We calculate the current deformation gradient $\mathbf{F}$ of the tetrahedron in terms of $\mathbf{D}_{\text{def}}$ and $\mathbf{D}_{\text{rest}}$ as follows:

$$\mathbf{F} = \mathbf{D}_{\text{def}}\mathbf{D}_{\text{rest}}^{-1}, \tag{31}$$

while the Green–Lagrange strain tensor can be expressed as:

$$\mathbf{G} = \mathbf{F}^T\mathbf{F} - \mathbf{I}, \tag{32}$$

where $\mathbf{I}$ is the identity matrix. An energy based potential is derived from $\mathbf{G}$ and measures the nonlinear strain, capturing both stretching and shearing constraints. Therefore, we define individual constraint functions based on the components of the strain tensor $\mathbf{S} = \mathbf{F}^T\mathbf{F}$. The diagonal entries $\mathbf{G}_{ii}$ represent stretch, while the off-diagonal entries $\mathbf{G}_{ij} = \mathbf{G}_{ji}$ correspond to shear. Then, the stretch constraints are defined as follows:

$$C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \mathbf{S}_{ii} - s_i^2, \quad i \in \{1, 2, 3\}, \tag{33}$$

where s_i is the rest stretch. The shear constraints correspond to the off-diagonal components, which can be defined as:

$$C_{\text{shear}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \mathbf{S}_{ij}, \quad i, j \in \{1, 2, 3\}, i < j, \tag{34}$$

where j is the corresponding axis index. The shear constraints can be decoupled from the stretch constraints. For more details on how strain constraint is adjusted to the XPBD formulation, see [3]. Both shape matching and tetrahedral strain constraints can be utilised for soft-body modelling. Shape matching constraints aim to preserve the overall form by approximating a rigid body transformation. In contrast, tetrahedral strain constraints prioritise the physical plausibility of deformations at the element level.

5 Force haptic feedback

In this section, we discuss how we formulate force-based haptic feedback using XPBD within our framework.

5.1 Haptic device

Haptic technology enables users to interact with virtual environments through tactile sensations, typically delivered through vibrations or force feedback [29, 30]. While standard VR controllers often rely on vibrotactile cues triggered by collisions or discrete events, these lack the capacity to convey continuous or directional force information [9, 14]. In contrast, wearable force haptic systems such as haptic gloves provide multimodal feedback, including force, vibrotactile, and thermal cues, allowing users to perceive pressure, texture, and temperature during interaction. In our framework, we employ the *WEART TouchDIVER G1* haptic glove [35] (which we will refer to as the *WEART Glove*) as an input device to deliver force feedback during deformable body manipulation (see Fig. 2). The glove transmits force cues via modular fingertip actuators, which can be configured according to task requirements. Its lightweight three-finger design provides a balance between functionality and comfort. Unlike bulkier systems such as the HaptX glove or stationary devices like the Phantom Premium, the WEART Glove offers a compact, wireless, and wearable solution that supports natural interaction in VR environments.

The WEART Glove captures the movements of the user's individual three fingers through integrated motion sensors. It also tracks hand and wrist positions in space and transmits real-time joint position and orientation data to Unity via the WEART SDK. This data drives a virtual hand model in real-time, ensuring spatial alignment between the physical hand and its virtual counterpart. A common approach to calculate the haptic force for driving haptic actuators in contact interactions is the penalty-based formulation, in which the system computes a contact point $\mathbf{c}$ on the object's surface, the local outward normal $\mathbf{n}$, and a penetration depth $\text{pd} = \max(0, -(\mathbf{P}_{\text{finger}} - \mathbf{c}) \cdot \mathbf{n})$. The restoring force is then approximated to $\mathbf{F} = k \cdot \text{pd} \cdot \mathbf{n}$, where k is a stiffness coefficient. In mechanotactile rendering, where the actuator delivers a scalar pressure rather than a directional force, the magnitude $k \cdot \text{pd}$ is mapped to the actuator input range.

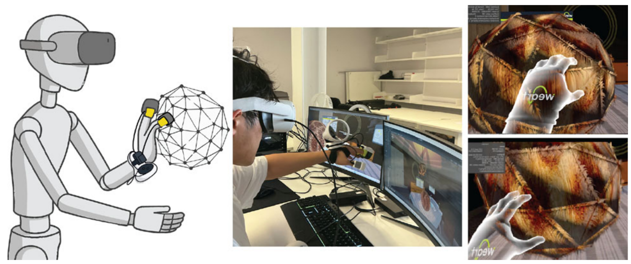

Fig. 5 The WEART Glove drives the virtual hand, while contact interactions between the fingers and deformable bodies are handled within the XPBD solver. The resulting interaction forces are computed and transmitted back to the glove to provide haptic feedback

5.2 Haptic feedback force integration

To integrate haptic force computation into the XPBD framework, we derive the haptic signal directly from the Lagrange multipliers λ_j accumulated by the XPBD solver during constraint projection. The λ_j encodes the magnitude of the constraint force that maintains the geometric invariant of constraint j under the current deformation. Rather than explicitly computing contact geometry or tuning a stiffness coefficient k, we read λ_j directly from the XPBD solver, which naturally encodes the material's nonlinear, position-dependent resistance to deformation. We aggregate the volume and tetrahedral strain multipliers $\hat{\lambda}_j^V, \hat{\lambda}_j^S$ over all tetrahedra via inverse-distance weighting relative to the grasp centre. Contact constraint multipliers $\hat{\lambda}^C$ are additionally incorporated, producing a normalised scalar in [0, 1] that is dispatched to the WEART TouchDIVER's actuator. This renders force feedback that mirrors the magnitude of resistance imposed by the virtual material (see Fig. 5).

The proposed system supports multiple interaction modalities, including grasping and poking. When a pinch gesture is detected, the reference positions $\{\mathbf{x}_j^{\text{ref}}\}$ of all particles are cached, and particle positions are prescribed to follow the hand displacement each frame:

$$\mathbf{x}_j \leftarrow \mathbf{x}_j^{\text{ref}} + \Delta \mathbf{p}_{\text{hand}} \tag{35}$$

After external forces are applied and predicted positions $\tilde{\mathbf{x}}_i$ are estimated, the contact set is defined as:

$$\mathcal{N}(r) = \begin{cases} \emptyset & \text{if grasping} \\ \{i : \|\tilde{\mathbf{x}}_i - \mathbf{p}_{\text{finger}}\| \leq r\} & \text{otherwise} \end{cases} \tag{36}$$

For each particle $i \in \mathcal{N}(r)$, a contact constraint is generated:

$$C_i(\tilde{\mathbf{x}}) = \|\tilde{\mathbf{x}}_i - \mathbf{p}_{\text{finger}}\| - r \geq 0 \tag{37}$$

The XPBD solver resolves each contact constraint:

$$\Delta\lambda_i = \frac{-C_i - \tilde{\alpha}\lambda_i}{\|\nabla C_i\|^2/m + \tilde{\alpha}} \tag{38}$$

$$\Delta\tilde{\mathbf{x}}_i = \frac{\Delta\lambda_i}{m}\nabla C_i, \qquad \nabla C_i = \frac{\tilde{\mathbf{x}}_i - \mathbf{p}_{\text{finger}}}{\|\tilde{\mathbf{x}}_i - \mathbf{p}_{\text{finger}}\|} \tag{39}$$

where $\tilde{\alpha} = \alpha/\Delta t^2$ is the effective compliance. The accumulated multiplier λ_i^C represents the normal contact force between the finger and particle i. All constraints $\{C_j^V, C_j^S, C^{SM}, C_i^C\}$ are projected simultaneously, accumulating $\lambda_j^V, \lambda_j^S, \lambda_i^C$. Particle positions are then finalised.

The normalised contact multiplier aggregates over all particles in $\mathcal{N}(r)$:

$$\hat{\lambda}^C = \frac{\sum_{i \in \mathcal{N}(r)} |\lambda_i^C|}{|\mathcal{N}(r)| \cdot \tau^C} \tag{40}$$

For each tetrahedron j, the volume and strain multipliers are normalised:

$$\hat{\lambda}_j^V = \text{clip}\left(\frac{|\lambda_j^V|}{\tau^V},\ 0,\ 1\right), \qquad \hat{\lambda}_j^S = \text{clip}\left(\frac{\|\boldsymbol{\lambda}_j^S\|}{\tau^S},\ 0,\ 1\right) \tag{41}$$

$$f_j = \beta_v \hat{\lambda}_j^V + \beta_s \hat{\lambda}_j^S \tag{42}$$

with inverse-distance weights $w_j = 1/(d_j + \varepsilon)$, where d_j is the distance from tetrahedron j's centroid to the grasp centre $\mathbf{p}_g$. The haptic intensity is computed as:

$$F_{\text{raw}} = \begin{cases} F_{\text{VS}} & \text{if grasping} \\ \text{clip}(\beta_c \hat{\lambda}^C + (1 - \beta_c) F_{\text{VS}},\ 0,\ 1) & \text{if } |\mathcal{N}(r)| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

where $F_{\text{VS}} = \sum_{j=1}^{M} w_j f_j / \sum_{j=1}^{M} w_j$ and $\beta_c, \beta_v, \beta_s$ are weighting coefficients with $\beta_v + \beta_s = 1$. The output is temporally smoothed:

$$F_{\text{out}} \leftarrow (1 - \alpha_t) F_{\text{out}} + \alpha_t F_{\text{raw}} \tag{44}$$

The smoothed scalar $F_{\text{out}} \in [0, 1]$ is dispatched to the WEART TouchDIVER, rendering the material's deformation resistance as a continuous pressure signal. This ensures that contact-induced deformation remains visually plausible while respecting the underlying material model. As a result, users receive tactile feedback from the glove that aligns closely with the observed deformation of the virtual soft body (see Algorithm 2).

Algorithm 2 XPBD-driven haptic feedback

Require: Tetrahedral mesh with particles $\{\mathbf{x}_i\}$, volume constraint, tetrahedral strain constraint, shapematching constraints $\{C_j^V, C_j^S, C^{SM}\}$, contact radius r

Build constraints from rest configuration;
while simulation running **do**
  if collision detection meets grasping requirements **then**
    if not currently grasping **then**
      Cache $\{\mathbf{x}_j^{\text{ref}}\}$ for all particles
    end if
  end if
  if ISGRASP **then**
    Prescribe $\mathbf{x}_j \leftarrow \mathbf{x}_j^{\text{ref}} + \Delta \mathbf{p}_{\text{hand}}$ for all particles
  end if
  Apply external forces; predict $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
  if ISGRASP **then**
    $\mathcal{N}(r) \leftarrow \emptyset$
  else
    $\mathcal{N}(r) \leftarrow \{i : \|\tilde{\mathbf{x}}_i - \mathbf{p}_{\text{finger}}\| \le r\}$
  end if
  for iter $= 1$ to N_{iter} **do**
    Reset $\lambda_j \leftarrow 0$ for all constraints
    Generate contact constraints C_i^C for $i \in \mathcal{N}(r)$
    Project $\{C_j^V, C_j^S, C^{SM}, C_i^C\}$, accumulating $\lambda_j^V, \lambda_j^S, \lambda_i^C$
    Discard C_i^C
  end for
  Update velocities; commit $\mathbf{x}_i \leftarrow \tilde{\mathbf{x}}_i$
  $\mathbf{p}_g \leftarrow$ mean fingertip position
  for each tetrahedron $j = 1$ to M **do**
    $d_j \leftarrow \|\mathbf{c}_j - \mathbf{p}_g\|$
    $\hat{\lambda}_j^V \leftarrow \text{clip}(|\lambda_j^V|/\tau^V,\ 0,\ 1)$
    $\hat{\lambda}_j^S \leftarrow \text{clip}(\|\boldsymbol{\lambda}_j^S\|/\tau^S,\ 0,\ 1)$
    $w_j \leftarrow 1/(d_j + \varepsilon)$
    $f_j \leftarrow \beta_v \hat{\lambda}_j^V + \beta_s \hat{\lambda}_j^S$
  end for
  $F_{\text{VS}} \leftarrow \dfrac{\sum_{j=1}^{M} w_j f_j}{\sum_{j=1}^{M} w_j}$
  if ISGRASP **then**
    $F_{\text{raw}} \leftarrow F_{\text{VS}}$
  else if $|\mathcal{N}(r)| > 0$ **then**
    $\hat{\lambda}^C \leftarrow \text{clip}\left(\dfrac{\sum_{i \in \mathcal{N}(r)} |\lambda_i^C|}{|\mathcal{N}(r)| \cdot \tau^C},\ 0,\ 1\right)$
    $F_{\text{raw}} \leftarrow \beta_c \cdot \hat{\lambda}^C + (1 - \beta_c) \cdot F_{\text{VS}}$
  else
    $F_{\text{raw}} \leftarrow 0$ {no interaction}
  end if
  $F_{\text{out}} \leftarrow (1 - \alpha_t) F_{\text{out}} + \alpha_t F_{\text{raw}}$
  WEARTSDK.SETFORCE(F_{out})
end while

6 Results

The implementation of our framework was developed in *Unity3D*, using *C#* on a desktop computer equipped with a *13th Gen Intel(R) Core(TM) i7-13700K* processor running at 3.40 GHz. The input devices used are the Head Mounted Device (*Meta Quest 2* headset) and the *WEART TouchDIVER G1* haptic glove. Please refer to the enclosed video for the animated results. Our method is evaluated on diverse sets of models that vary in geometric complexity and mesh resolution. Each model was tetrahedralized using *TetGen* to produce the volumetric mesh for simulation. The surface mesh for rendering is constructed from the boundary faces of the tetrahedral mesh.

Environmental assets for the scenes were obtained from the *Unity Asset Store*, and rendering was performed using Unity's built-in render pipeline. At the beginning of each simulation session, the deformable bodies are initialised under the influence of gravity until they reach a stable resting configuration. Following this initialisation, the WEART Glove is calibrated with the virtual hand model to ensure alignment between physical and virtual interaction. Once calibration is completed, the user can begin interacting with the deformable bodies.

6.1 Performance analysis

We measured the performance of our method using six different virtual scenes, featuring virtual deformable objects with a varying range of material characteristics. Our results for all scenes are summarised in Tables 1 and 2, where Table 1 addresses the first five scenes, while Table 2 summarises the sixth, which is flag. In the tables, we report the number of volume constraints and the number of tetrahedral stretch constraints, we remark that each tetrahedron consists of one tetrahedral stretch constraint, and one volume constraint, which explains why the columns N_v, N_s, and *Tetra.* are equal. Additionally, the ranges used across all tested scenes: compliance ranging from 0.2 to 0.4, damping from 0.02 to 0.05, and uniform particle mass set to 1.0 for all deformable bodies.

A simulation timestep of $\Delta t = 1/60$ s was used across all scenes, consistent with common practice in real-time interactive simulation and sufficient to capture the dynamic response of the simulated objects at interactive rates. The solver was run for 5 iterations per timestep, providing a practical balance between simulation stability and real-time computational constraints. Among three types of constraints, the tetrahedral strain constraints accounted for 86.6% of the total time, whereas the volume constraints accounted for 11.1%, and the shape matching constraints for only 2.3%. The tetrahedral strain constraints introduce significant computational overhead due to their high algorithmic complexity. It oper-

Table 1 The table lists the properties and performance statistics of five virtual 3D deformable objects in a 120 Hz VR HMD scene. N_v denotes the number of volume constraints; N_s denotes the number of tetrahedral strain constraints, N_m denotes the number of shape matching constraints; T_v denotes the average volume constraint computation time per frame; T_s denotes the average tetrahedral strain constraint computation time per frame; T_m denotes the average shape matching constraint computation time per frame; $T_{\text{total}} = T_v + T_s + T_m$; *Tetra.*, the number of tetrahedra of the model. The number of solver iterations is increased to preserve the relatively complex polygonal mesh structure of the soft bear model

Object	N_v	N_s	N_m	T_v (ms)	T_s (ms)	T_m (ms)	T_{total} (ms)	Tetra.	Scene vertices	Scene triangles	Time step	Solver iterations	Game view FPS	Volume loss (%)
Fur ball	1280	1280	1	1.471	11.972	0.219	12.790	1280	486.6k	496.6k	1/60	5	70.2	0.4
Liver	770	770	1	0.848	7.836	0.120	8.804	770	201k	221k	1/60	5	96.3	0.6
Balloon	1713	1713	1	1.931	17.157	0.233	19.321	1713	346.6k	464.4k	1/60	5	48	0.8
Jelly	1530	1530	1	1.693	14.814	0.226	16.733	1530	191.8k	254.6k	1/60	5	59.9	0.6
Bear toy	1041	1041	1	1.155	10.467	0.146	11.768	1041	156.6k	200.4k	1/60	5	76.3	0.5

Table 2 Properties and performance statistics the flag object, where N_c, denotes the number of stretch constraints; N_b denotes the number of bending constraints; T_c denotes the average stretch constraint computation time per frame; and T_b denotes the average bend constraint computation time per frame

	N_c	N_b	T_c (ms)	T_b (ms)	T_{total} (ms)	Scene vertices	Scene triangles	Time step	Solver iterations	Game view FPS
Flag	3136	3071	0.573	0.693	1.266	1.3M	1.2M	1/60	2	197

ates on each tetrahedral element and is applied 5 times per frame during the iterative solver loop. In contrast, although the volume constraint is also applied to each tetrahedral element, it involves relatively simple geometric computations. The shape matching constraint, on the other hand, is applied globally to the entire soft body and contributes minimally to the computational load due to its low complexity. These factors make tetrahedral strain constraints the primary performance bottleneck in the simulation.

Figure 6 illustrates the interaction scenarios for the jelly deformable body. To quantitatively assess simulation accuracy, we compute the volume loss rate as $e = (v_0 - v)/v_0$, where v_0 denotes the rest-state volume and v the volume at the current frame. In the absence of external interaction, volume remains at 100% throughout simulation, confirming the absence of numerical drift. The reported losses of 0.4–0.8% are observed during active deformation and remain stable throughout sustained contact, attributable to the finite number of solver iterations, which prevents full convergence of the volume constraints within each timestep. Upon release, volume recovers to the rest state. As reported in Table 1, across all tested scenes, our method effectively preserves the volume of 3D deformable bodies, maintaining a loss rate below 1%.

In Fig. 7, the user is interacting with a virtual deformable cloth object. Since it is a surface-based representation, volume constraints and tetrahedral elements are not present. Instead, we incorporate stretch and bending constraints to simulate cloth behaviour. In this scene, we apply a constant force of 20 N to each particle in the direction of -1, which produces natural wrinkles that suggest dynamic airflow. We refer to the accompanying video for visual results. Performance metrics are reported in Table 2.

6.2 System latency analysis

All measurements (in Table 3) are taken in standalone build mode with vertical synchronisation disabled, to ensure that the timings reflect the actual pipeline latency. Hardware-side latency is not included in the reported measurements. These hardware-determined delays are additive and independent of the proposed simulation pipeline. The 99th-percentile software latency during contact interactions is 11.38 ms. The XPBD constraint solver dominates per-frame cost, accounting for 99.0% of total latency in the idle state and 95.3% in the interaction. The slight reduction in relative share during contact reflects the activation of the haptic feedback pipeline rather than any increase in solver cost. Activation of the haptic computation thus introduces a marginal increase of 0.45 ms in mean total latency, confirming that the proposed integration scheme does not compromise the real-time guarantees of the underlying simulation. The solver itself increases only marginally from 9.17 ms to 9.26 ms between states, confirming that contact interactions do not impose additional constraint-solving overhead. Other simulation stages, including position prediction, contact detection, finalisation, and rendering update, together account for less than 0.1 ms and are effectively free at the analysed scale.

6.3 User study

We evaluated the performance of our proposed system with human subject participants ($N = 16$). The study questions were structured around four key dimensions: (1) visual feedback, (2) haptic Feedback, (3) visual-haptic congruence, and (4) interaction stability. We conducted our experiments at Brunel University London, after obtaining ethical approval (reference: 50550-LR-Mar/2025-54032-2) prior to data collection. Participants gave written informed consent before participating, and were informed about privacy and safety considerations before participating in the experiment. Among the 16 participants, 13 reported having used VR on only one to five occasions, and the majority had no prior experience with haptic devices, suggesting that the sample is representative of a general, non-expert population. All participants were asked to evaluate two experimental conditions. Condition A represented the proposed system, while Condition B employed a baseline implementation using the standard WEART SDK for haptic feedback. The order of conditions was counterbalanced across participants to mitigate potential order effects. Each participant began with virtual hand calibration, after which they were instructed to interact with the soft-body simulation through grasping and poking actions. Following a 10 minute rest period, participants performed the identical set of interactions under the other condition. Upon completing both conditions, participants were asked to complete the questionnaire to evaluate their experience and indicate their preference between the two systems.

Visual feedback: A Wilcoxon signed-rank test was conducted to compare the visual feedback ratings between Condition A and Condition B. The results revealed no statistically significant difference between the two conditions ($W = 5.0$, $p = 0.128$), with Condition A achieving a mean rating of 5.52 ($SD = 1.87$) and Condition B achieving a mean rating of 5.17 ($SD = 2.16$). The absence of a significant difference, combined with the comparably high ratings observed in both conditions, confirms that the XPBD simulation pipeline delivers high visual realism, demonstrating that the proposed system achieves good visual fidelity.

Haptic feedback: Regarding haptic feedback ratings, a significant difference was found between Condition A and Condition B ($W = 4.0$, $p = 0.006$, $r = 0.827$), with Condition A achieving a mean rating of 5.27 ($SD = 1.84$) and Condition B achieving a mean rating of 4.17 (SD = 1.97). The large effect size further corroborates the robustness of this finding,

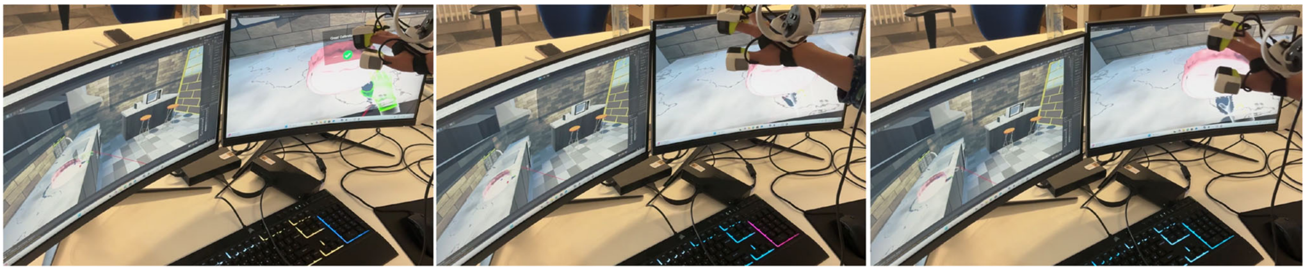

Fig. 6 User touching a jelly-like virtual soft body object in a virtual kitchen scene. Upon contact, the jelly exhibits elastic jitter behaviour, demonstrating high physical plausibility

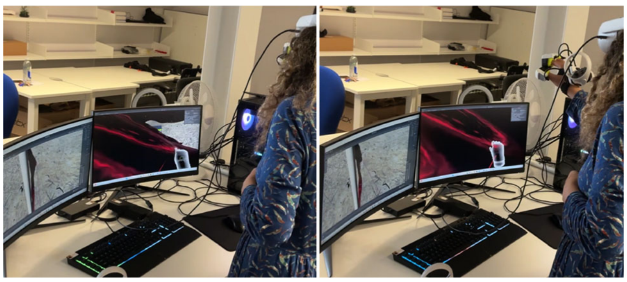

Fig. 7 The system delivers real-time visual rendering and force feedback, enabling the user to perceive cloth deformation and haptic force magnitudes during manipulation

Table 3 Per-stage simulation latency broken down by interaction state (idle and contact), measured over $N = 2837$ frames collected in standalone build mode with vertical synchronisation disabled. All values in milliseconds

Stage	Mean	Med	Std	P95	P99
Idle					
Predict	0.06	0.06	0.01	0.07	0.10
Contact detect	0.01	0.01	0.00	0.01	0.03
XPBD solve	9.17	9.20	0.52	9.91	10.25
Finalise	0.01	0.01	0.00	0.01	0.02
Render	<0.01	<0.01	<0.01	<0.01	<0.01
Force calc	<0.01	<0.01	<0.01	<0.01	<0.01
SDK dispatch	0.01	0.00	0.02	0.02	0.15
Total	9.27	9.29	0.52	10.00	10.34
Contact					
Predict	0.06	0.06	0.01	0.07	0.13
Contact detect	0.01	0.01	0.00	0.01	0.01
XPBD solve	9.26	9.36	0.69	10.12	10.56
Finalise	0.01	0.01	0.00	0.01	0.02
Render	<0.01	<0.01	<0.01	<0.01	<0.01
Force calc	0.20	0.16	0.29	0.19	0.92
SDK dispatch	0.17	0.18	0.09	0.24	0.36
Total	9.72	9.79	0.75	10.68	11.38

confirming that the proposed system delivers superior haptic feedback than the baseline implementation.

Visual-haptic congruence: Condition A (our proposed system) received a mean rating of 5.35 ($SD = 1.84$), compared to 4.33 (SD = 2.26) for Condition B (the baseline). A significant difference was found between the two conditions ($W = 2.5$, $p = 0.003$), indicating that participants perceived the timing and synchronisation between visual and haptic feedback to be significantly better in the proposed system than in the baseline implementation.

Interaction stability: In interaction stability, the result shows a statistically significant difference between the two system ($p = 0.005$). The proposed system achieved a mean stability rating of 5.50 ($SD = 1.72$), substantially higher than that of the baseline, which recorded a mean of 4.10 ($SD = 2.28$). This significant difference demonstrates that the proposed system delivers markedly superior interaction stability, highlighting the effectiveness of the integrated architecture in sustaining reliable and consistent real-time performance during soft-body interaction.

Preference: After completing all trials (3 per condition), participants compared the two conditions. The results of Fig. 8 indicated that the majority of participants favoured our proposed system over the baseline. In particular, a significant preference was observed for haptic naturalness ($p = 0.022$), with 56.2% of participants rating the proposed system more favourably. Although the overall preference score also leaned toward Condition A (D4: Median = −2.50, Mean = −1.19), this difference did not reach statistical significance ($p = 0.054$). We attribute this marginal result to individual differences in participant background; the majority of participants had limited prior experience with both VR and haptic devices, which may have affected their ability to critically differentiate between the two conditions and articulate a definitive preference.

Overall, participants demonstrated a clear preference for our proposed system (Fig. 10), which consistently achieved higher ratings across the majority of evaluation dimensions, particularly in haptic feedback, visual-haptic congruence, and interaction stability (Fig. 9). The visual feedback scores confirm that the XPBD-based deformable body simulation produces plausible, perceptually convincing deformation behaviour, thereby validating the visual fidelity of the pro-

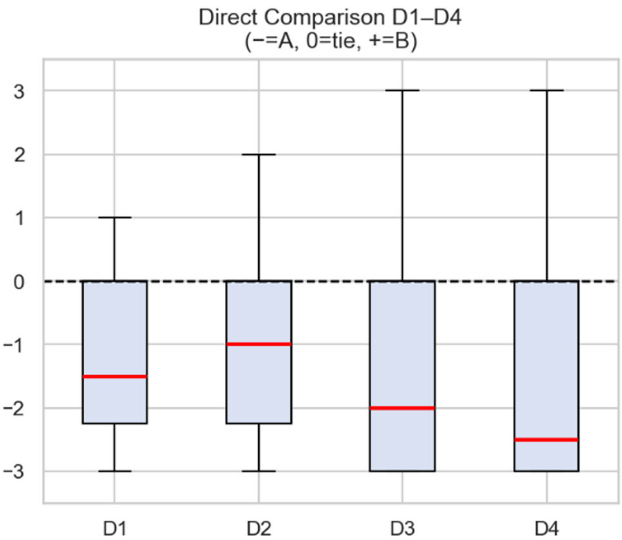

Fig. 8 Box plots of direct comparison ratings (D1–D4) between the proposed system and the baseline, where negative values indicate a preference for the proposed system (**A**), zero indicates no perceived difference, and positive values indicate a preference for the baseline (**B**). The median scores for all four items fall below zero, reflecting a consistent preference for the proposed system

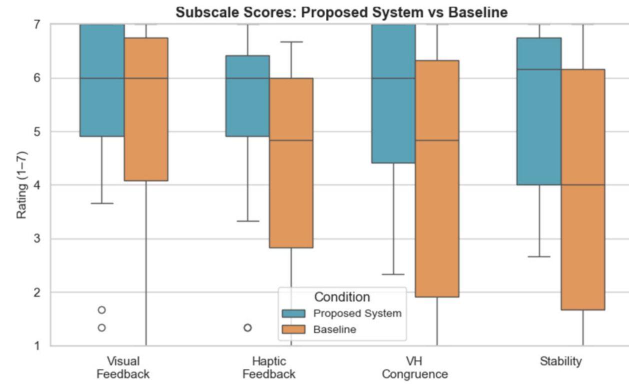

Fig. 9 Box plots illustrating the distribution of subscale ratings across four dimensions. The proposed system (blue) consistently achieved higher median scores than the baseline (orange) across all evaluated dimensions

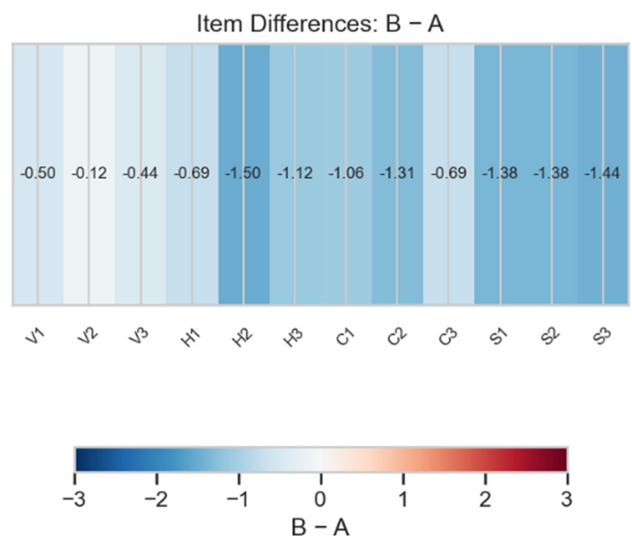

Fig. 10 Heatmap of item-level mean score differences across 12 rating items, where A refers to our proposed system and B refers to the baseline implementation. Positive values indicate higher ratings for the proposed system

posed rendering pipeline. Furthermore, the haptic feedback results demonstrate that the proposed approach successfully enables realistic real-time interaction with deformable body objects. The inferior performance of the baseline in interaction stability is attributed to its binary force feedback mechanism. The baseline, which uses the WEART Unity SDK, computes haptic force based on the distance between the real and virtual hand; however, in the context of soft-body interaction, this distance remains zero during contact, causing the force to be generated solely based on a predefined stiffness value rather than the actual deformation state. This results in a discretised force output that produces abrupt, discontinuous force sensations, most notably during the transition from compression to recovery, when no gradual force variation is conveyed to the user. This observation was further corroborated by post-experiment interviews, in which participants explicitly reported that the proposed system enabled them to perceive continuous, progressive force changes corresponding to soft-body deformation, in contrast to the discrete force response characteristic of the baseline. These results demonstrate that the proposed approach effectively addresses this limitation, providing meaningful, graded haptic feedback during soft-body interactions with the WEART G1 glove (Fig. 10).

Additionally, it is worth noting that Overall Plausibility (D1) was consistently skewed towards the proposed system. This finding suggests that well-designed, physically grounded haptic feedback does not operate in isolation but meaningfully enhances the overall user experience, reinforcing the value of tight visual-haptic integration in immersive VR environments.

7 Conclusion

We have presented an efficient and light-weight system that integrates XPBD with a wearable haptic glove to enable real-time interaction with deformable bodies. During the simulation, the soft model preserves volume and structure, allowing physically plausible deformations in response to external forces. Lagrange multipliers λ accumulated during constraint projection are used directly to derive the haptic signal, enabling users to feel soft tissue deformation in real-time. In the first step of our approach, XPBD computes deformation based on a set of constraints. By incorporating stretch, bend, volume preservation, and shape matching constraints, the system maintains the elastic nature of the simulated model. Furthermore, we implement a custom deformation mechanism that couples contact interactions to plausible deformations, enabling visually consistent and physically reactive interactions. Then, in the second step, the resulting deformations and constraint violations are used to calculate the haptic parameters. Finally, haptic parameters are sent via the SDK to the glove, generating corresponding force feedback.

We evaluated our method across six scenarios, achieving peak visual rendering rates of approximately 96.3 FPS for 3D deformable object simulation and up to 197 FPS for 2D cloth simulation on consumer hardware. Our user study confirmed that participants found the simulated deformable body visually plausible and preferred the proposed system for visual and haptic congruence. Participants reported that the force feedback varied in accordance with the observed deformation behaviour in our system, which led to higher ratings. This also demonstrates that physically grounded haptic rendering enhances overall perceptual quality in VR deformable body interaction.

The current implementation does not fully address self-collision, which can occur in extreme deformation cases. Moreover, end-to-end latency remains dependent on the hardware device and is beyond the scope of software-level optimisation. Furthermore, the user study was conducted with a relatively small sample size of 16 participants. In the future, we aim to conduct larger-scale user evaluations to compare tactile feedback quality across different hardware setups.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00371-026-04570-3.

Acknowledgements M. Z. received funding from the Turing Mobility Fund to support a 3-month research visit to the TAUCHI Research Center at Tampere University. The authors declare no other competing financial or non-financial interests.

Author Contributions M. Z. prepared all the figures and ran the simulations. M. Z. and N. A. wrote the manuscript together. All authors contributed to and reviewed the final version.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Batik, T., Brument, H., Vasylevska, K., Kaufmann, H.: Shiftly: a novel origami shape-shifting haptic device for virtual reality. IEEE Trans. Visual Comput. Graph. **31**(5), 2331–2341 (2025)
2. Biswas, S., Visell, Y.: Haptic perception, mechanics, and material technologies for virtual reality. Adv. Func. Mater. **31**(39), 2008186 (2021)
3. Cetinaslan, O.: Esbd: exponential strain-based dynamics using xpbd algorithm. Comput. Graph. **116**, 500–512 (2023)
4. Chen, G., Qian, L., Ma, J., Zhu, Y.: Smoothed FE-Meshfree method for solid mechanics problems. Acta Mech. **229**(6), 2597–2618 (2018)
5. Comas, O., Taylor, Z.A., Allard, J., Ourselin, S., Cotin, S., Passenger, J.: Efficient nonlinear fem for soft tissue modelling and its gpu implementation within the open source framework sofa. In: Bello, F., Edwards, P.J.E. (Eds.) Biomedical Simulation, pp. 28–39. Springer, Berlin (2008)
6. Duriez, C.: Control of elastic soft robots based on real-time finite element method. In: 2013 IEEE International Conference on Robotics and Automation, pp. 3982–3987 (2013)
7. Fang, G., Tian, Y., Weightman, A., Wang, C.C.: Collision-aware fast simulation for soft robots by optimization-based geometric computing. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12614–12621. IEEE (2022)
8. Fazioli, F., Ficuciello, F., Fontanelli, G.A, Siciliano, B., Villani, L.: Implementation of a soft-rigid collision detection algorithm in an open-source engine for surgical realistic simulation. In: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2204–2208 (2016)
9. Giri, G.S., Maddahi, Y., Zareinia, K.: An application-based review of haptics technology. Robotics **10**, 29 (2021)
10. Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: Acm siggraph 2016 Courses, pp. 1–52 (2016)
11. Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H., Gao, F., Yang, Y., Jiang, C.: Vr-gs: a physical dynamics-aware interactive gaussian splatting system in virtual reality. In: ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH '24. Association for Computing Machinery, New York (2024)
12. Johnsen, S.F., Taylor, Z.A., Han, L., Hu, Y., Clarkson, M.J., Hawkes, D.J., Ourselin, S.: Detection and modelling of contacts in explicit finite-element simulation of soft tissue biomechanics. Int. J. Comput. Assist. Radiol. Surg. **10**(11), 1873–1891 (2015)
13. Kim, H., Kim, M., Lee, W.: Hapthimble: A wearable haptic device towards usable virtual touch screen. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 3694–3705. Association for Computing Machinery, New York
14. Kourtesis, P., Argelaguet, F., Vizcay, S., Marchal, M., Pacchierotti, C.: Electrotactile feedback applications for hand and arm interactions: a systematic review, meta-analysis, and future directions. IEEE Trans. Haptics **15**(3), 479–496 (2022)

15. Kudry, P., Cohen, M.: Development of a wearable force-feedback mechanism for free-range haptic immersive experience. Front. Virtual Reality **3**, 824886 (2022)
16. Li, X., Li, M., Han, X., Wang, H., Yang, Y., Jiang, C.: A dynamic duo of finite elements and material points. In: ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH '24. Association for Computing Machinery, New York (2024)
17. Lim, J., Choi, Y.: Force-feedback haptic device for representation of tugs in virtual reality. Electronics **11**, 1730 (2022)
18. Lin, M.C., Otaduy, M.: Haptic Rendering: Foundations, Algorithms, and Applications. CRC Press, Boca Raton (2008)
19. Liu, Y., Kerdok, A.E., Howe, R.D.: A nonlinear finite element model of soft tissue indentation. In: Cotin, S., Metaxas, D. (Eds.) Medical Simulation, pp. 67–76. Springer, Berlin (2004)
20. Lupinetti, K., Ranieri, A., Giannini, F., Monti, M.: 3d dynamic hand gestures recognition using the leap motion sensor and convolutional neural networks (2020)
21. Macklin, M., Müller, M., Chentanez, N.: Xpbd: position-based simulation of compliant constrained dynamics. In: Proceedings of the 9th International Conference on Motion in Games, MIG '16, pp. 49–54. Association for Computing Machinery, New York (2016)
22. Macklin, M., Storey, K., Lu, M., Terdiman, P., Chentanez, N., Jeschke, S., Müller, M.: Small steps in physics simulation. In: Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '19. Association for Computing Machinery, New York (2019)
23. Mahmud, M., Sayama, T., Moriyama, S., Hashimoto, T., Kato, H.: Standing balance improvement using vibrotactile feedback in virtual reality. In: Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology (VRST '22), pp. 1–10. Association for Computing Machinery, New York (2022)
24. Martins, M., Morais, L., Torchelsen, R., Nedel, L., Maciel, A.: Efficient position-based deformable colon modeling for endoscopic procedures simulation. In: ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH '24, Association for Computing Machinery, New York (2024)
25. Mesit, J., Hastings, E.: Muti-level sb collide: collision and self-collision in soft bodies, vol. 01, (2006)
26. Müller, M., Chentanez, N., Kim, T.-Y., Macklin, M.: Strain based dynamics. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '14, pp. 149–157. Eurographics Association, Goslar (2015)
27. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. ACM Trans. Graph. **24**(3), 471–478 (2005)
28. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. J. Vis. Commun. Image Represent. **18**(2), 109–118 (2007)
29. Pacchierotti, C., Sinclair, S., Solazzi, M., Frisoli, A., Hayward, V., Prattichizzo, D.: Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives. IEEE Trans. Haptics **10**(4), 580–600 (2017)
30. See, A.R., Choco, J.A.G., Chandramohan, K.: Touch, texture and haptic feedback: a review on how we feel the world around us. Appl. Sci. **12**(9), 4686 (2022)
31. Si, H.: Tetgen, a Delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw. **41**(2), 11 (2015)
32. Tong, Q., Wei, W., Zhang, Y., Xiao, J., Wang, D.: Survey on hand-based haptic interaction for virtual reality. IEEE Trans. Haptics **16**(2), 154–170 (2023)
33. Va, H., Choi, M.-H., Hong, M.: Efficient simulation of volumetric deformable objects in unity3d: Gpu-accelerated position-based dynamics. Electronics **12**(10), 2229 (2023)
34. Wang, Y., Feng, L., Andersson, K.: Haptic force rendering of rigid-body interactions: a systematic review. Adv. Mech. Eng. **13**(9), 16878140211041538 (2021)
35. Weart. La Sapienza University leveraging Touchdiver's full haptic feedback in medical robots (2024). Accessed 20 July 2025
36. Xia, P.: New advances for haptic rendering: state of the art. Vis. Comput. **34**(2), 271–287 (2018)
37. Xu, W., Wang, Y., Huang, W., Duan, Y.: An efficient nonlinear mass-spring model for anatomical virtual reality. IEEE Trans. Instrum. Meas. **71**, 1–10 (2022)
38. Yan, X., Li, C.-F., Chen, X.-S., Hu, S.-M.: Mpm simulation of interacting fluids and solids. In: Computer Graphics Forum, vol. 37, pp. 183–193. Wiley Online Library, Hoboken (2018)
39. Yang, Z., Long, Y., Chen, K., Wei, W., Dou, Q.: Efficient physically-based simulation of soft bodies in embodied environment for surgical robot (2024)
40. Zeng, W., Liu, G.R.: Smoothed finite element methods (S-FEM): an overview and recent developments. Arch. Comput. Methods Eng. **25**(2), 397–435 (2018)
41. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., Grundmann, M.: Mediapipe hands: on-device real-time hand tracking (2020)
42. Zhang, J., Zhong, Y., Gu, C.: Deformable models for surgical simulation: a survey. IEEE Rev. Biomed. Eng. **11**, 143–164 (2017)
43. Zilles, C., Salisbury, J.: 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, pp. 146–151 (1995)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mingzhao Zhou is a PhD student in the Department of Computer Science at Brunel University London. His research interests lie in Virtual Reality and Haptics, with a focus on developing algorithms for physics simulation in real-time. His broader research goal is to bridge computational efficiency and perceptual realism in interactive virtual environments.

Nadine Aburumman received the Ph.D. degree in computer science from Sapienza University of Rome. She is currently an assistant professor in Computer Science at Brunel University London and leads the Graphics and Extended Reality Team (GERT). Her research interests include real-time computer graphics, virtual and extended reality, and physics-based simulation.

Zhenxing Li received his MSc degree in Radio Frequency Electronics from Tampere University of Technology and PhD degree in User Interface Software Development from University of Tampere, in 2009 and 2013 respectively. He is a member of the Tampere Unit for Computer–Human Interaction (TAUCHI) Research Center at Tampere University. His research interests include haptics, VR, gaze, multimodal interaction, mathematical modelling and algorithms.

Roope Raisamo is a professor of computer science in Tampere University, Faculty of Information Technology and Communication Sciences. He received his PhD degree in computer science from the University of Tampere in 1999. He is the head of TAUCHI Research Center (Tampere Unit for Computer–Human Interaction), leading Multimodal Interaction Research Group. His 25-year research experience in the field of human-technology interaction is focused on multimodal interaction, XR, haptics, gaze, gestures, interaction techniques, and software architectures.