Computers & Graphics 134 (2026) 104507

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Real-time haptic-based soft body suturing in virtual open surgery simulations ☆

George Westergaard [a], Mark Ellis [a], Jacob Barker [b], Sofia Garces Palacios [c], Alexis Desir [c], Ganesh Sankaranarayanan [c], Suvranu De [d], Doga Demirel [b,*]

[a] Department of Computer Science, Florida Polytechnic University, Lakeland, FL, USA
[b] School of Computer Science, University of Oklahoma, Norman, OK, USA
[c] Department of Surgery, University of Texas Southwestern Medical Center, Dallas, TX, USA
[d] College of Engineering, Florida A&M University - Florida State University, Tallahassee, FL, USA

ARTICLE INFO

Keywords:
Surgical simulator
Haptics
Virtual reality
Suturing
Extended position-based dynamics
Open surgery

ABSTRACT

In this work, we present a real-time virtual reality-based open surgery simulator that enables realistic soft-tissue suturing with bimanual haptic feedback. Our system uses eXtended Position-Based Dynamics (XPBD) for soft body and suture thread simulation, allowing stable real-time physics for complex interactions like continuous sutures and knot tying. In tests with all four common suturing techniques, purse-string, Connell, stay, and Lembert, the simulator maintained high frame rates (50–80 FPS) with up to 4155 simulated particles, demonstrating consistent real-time performance. As part of our work, we conducted a user study using our suturing simulator, where 24 surgical trainees and experts used the Virtual Colorectal Surgery Trainer – Rectal Prolapse simulator. The user study showed that 71% of participants (n=17) rated the anatomical realism as moderate to very high. Half (n=12) found the force feedback realistic, and 54% (n=13) participants found the force feedback useful, indicating effective immersion while also highlighting the need for improved haptic fidelity. Overall, the simulation provides a low-cost, high-fidelity training platform for open surgical suturing, addressing a critical gap in current virtual reality educational tools.

1. Introduction

Virtual surgical simulators are rapidly becoming a cornerstone of medical professional training [1–3]. Unlike traditional methods, such as using a cadaver, virtual simulations offer a modern approach with limitless scenario potential at minimal cost and resources [4]. Furthermore, with the addition of head-mounted displays (HMDs) and haptic force feedback devices, virtual simulations are advancing their fidelity with each passing year. However, realistic graphics, immersive environments, and haptic feedback are only part of the solution to making these simulations viable to use as learning materials for doctors and medical personnel. Crucially, virtual reality (VR) simulations must offer an accurate and realistic representation of surgical procedures to be valuable training tools. Failure to accurately model the behavior of deformable objects can degrade the training value of a simulator as trainees develop incorrect expectations for the behavior of real-world counterparts to the simulated objects. Additionally, failure to provide adequate realism specifically for a medical simulator can break immersion and further reduce training outcomes.

Unfortunately, existing VR simulations often fail to replicate real-world surgical scenarios, particularly in crucial areas like open surgery. Unlike laparoscopic surgery, open surgery involves a broader range of movements and actions that current simulations may not adequately address. This creates a critical gap in available training resources, hindering the development of crucial surgical skills. Our research addresses this gap by introducing a novel VR surgical simulation environment designed explicitly for suturing soft tissues in open surgery procedures with haptic feedback.

Computers are continuously improving, allowing for more extensive calculations to be made in real-time, enabling the use of more intensive simulation models. From Lenoir et al. [5] in 2004 to Jourdes et al. [6] in 2022, significant strides have been made in replicating surgical suturing. Technological improvements, like the creation of Position Based Dynamics (PBD) [7] and eXtended Position Based Dynamics (XPBD) [8], aid in the pursuit of a lifelike simulation. Building upon these advancements, our simulation utilizes XPBD to model and simulate organs and suture threads with increased realism. XPBD allows

☆ This article is part of a Special issue entitled: 'XR Tech Health' published in Computers & Graphics.

* Correspondence to: University of Oklahoma, School of Computer Science, Devon Energy Hall, 110 W. Boyd St., Norman, OK 73019, USA.
E-mail address: doga@ou.edu (D. Demirel).

https://doi.org/10.1016/j.cag.2025.104507
Received 27 March 2025; Received in revised form 24 November 2025; Accepted 28 November 2025
Available online 2 December 2025

us to create soft body bowel models that behave like their real-world counterparts, deforming and regaining their original shape as expected with minimal compute cost. Taking this approach offers a more realistic representation of open surgery and incorporates force-feedback haptics for an enhanced user experience. In turn, users can practice suturing on soft body models that mimic real-world scenarios which has the potential to revamp surgical training. This VR simulation environment can significantly improve surgical proficiency and, ultimately, patient outcomes by providing a cost-effective and realistic alternative to traditional methods. However, creating such an immersive simulator required addressing several technical challenges, including maintaining real-time performance with complex deformable models, simulating continuous suture threading, such as slack management and knot tying on soft tissue, and integrating dual haptic devices without instability. We overcame these challenges by combining an XPBD-based soft body physics solver with suture loop detection, knot formation, and slack resolution, all integrated with bi-manual haptic feedback.

1.1. Literature review

Surgical simulations are constantly evolving. They are growing more useful as preferred training methods because of the many benefits. With the advancement of technology, more virtual simulators are becoming affordable and applicable as teaching tools for surgeons to practice their knowledge and skills. It has been shown that medical students trained with VR simulations perform better in suturing tasks than those who learned via lectures or readings [9]. In particular, randomized trials and evaluations have confirmed the effectiveness of VR training in enhancing operative performance focusing on robotic surgery skills [10,11]. Their findings show the value of simulation-based education and motivate the development of more advanced simulators.

1.1.1. Surgical simulators in training

Surgical simulations are commonly used to train participants in fundamental skills or procedural knowledge. As shown in [12,13] suturing is a task that requires extensive practice and hands-on experience to master. Many laparoscopic suturing simulators have been developed and validated. Studies in [14–16] have established construct validity and examined the role of haptic feedback in these system and studies in [17–19] have investigated their effectiveness for skill acquisition and integration into structured training programs. In [20–22] correlations between simulator performance metrics and live surgical outcomes were studied. In vitro and clinical studies have demonstrated the validity of such simulators [17,22]. Duffy et al. [14] noted that LapSim provided construct validity and that the suturing module could effectively differentiate the level of laparoscopic skill based on the year of resident training. Zhou et al. [15] showed that haptic feedback is a key factor in the initial learning of a complex surgical task like suturing and knot-tying, but the benefits taper off exponentially with a learning curve plateau. Hagelsteen et al. [16] followed this study by comparing more simulators with varying degrees of haptics. They concluded that haptics help surgeons significantly reduce stretch damage but are noted as low in fidelity by testing surgeons.

1.1.2. Endoscopy and laparoscopy simulators

Endoscopic suturing simulators have also been developed for bariatric procedures and endoscopic sleeve gastroplasty [23,24]. In [23], the endoscopic suturing was performed using a basic keyboard-and-mouse interface to validate the software functionality of the endoscopic simulator without a custom haptic device. In contrast, Erden et al. [24] utilized a dedicated hardware platform that replicated the instrument mechanics and force feedback of endoscopic sleeve gastroplasty suturing. Chheang et al. [25] also incorporated dedicated hardware to emulate the feel of laparoscopic instruments for simulated liver surgery.

While these laparoscopic and endoscopic simulators have demonstrated measurable training benefits, they are primarily designed for minimally invasive procedures that constrain instrument motion to narrow access ports. This focus limits their applicability to open surgical techniques, where surgeons interact directly with tissues and have unrestricted tool movement.

1.1.3. Haptics in surgical simulators

Besides laparoscopic and endoscopic surgery, robotic surgery is a growing field with increasing benefits to patients who undergo this type of surgery. One flaw that some robotic surgery devices have is that they do not impart haptic feedback to the surgeon operating. Jourdes et al. [6] introduced a visual feedback method to compensate for the lack of force feedback in robotic suturing and Pasini et al. [26] developed an autonomous camera control system to assist robotic surgeons, reducing operation time and cognitive load. Sanford et al. [27] examined suturing sub-steps in robot-assisted surgery and found that technical skill in each sub-step correlates with overall suturing performance. This implies that training surgeons should focus on mastering each step involved in the overall suturing process. These efforts improve specific aspects of robotic surgery training but do not address the tactile feedback missing in teleoperative systems. Neither laparoscopic nor robotic simulators provide the hands-on, tactile experience of open surgery, which involves unrestricted tool movement and direct tissue handling capabilities our system aims to provide.

Open surgery simulation remains comparatively underexplored. Differentiating from laparoscopic surgery, in open surgery, the operation often involves cutting open the skin or tissue to the outside world and is more invasive. Berkley et al. [28] created an open surgery sim where surgeons had to suture a patient's hand closed. They made the sim using a "Reach-In" environment where the user would see a near 3D rendering of the simulation in a pane of glass reflecting a screen above them. They could see this in 3D because of a head-mounted display or specially crafted 3D glasses. Surgeons can use a phantom haptic device to move their tools around and feel force-feedback. A recent practice among researchers trying to create and use open surgical simulators is to utilize VR controllers for user input. Gan et al. [29], Mok et al. [30], and Cardona Rivera et al. [31] use their headset's default controllers for their user input. While these approaches are accessible, they are deficient in fidelity. The controllers only provide vibration feedback and do not mimic the shape or mechanics of real surgical tools, limiting realism and skill transfer. In summary, existing open-surgery simulators either require expensive custom hardware [28] or sacrifice fidelity by relying on game controllers [29–31] or untethered magnetic haptic feedback system with low bandwidth for a real-time usage [32]. This shortcoming motivates our use of actual haptic devices and physics-based modeling to achieve a more realistic open surgery simulation.

1.1.4. Deformable objects in surgical simulators

Physically-based deformable modeling is key to high-fidelity surgical simulators. Deformable objects are key to representing non-rigid objects in simulations. A common method to calculate and show deformable objects, referred to as soft bodies, is to use XPBD. Position-based dynamics saw a breakthrough in Müller et al. [7] and was further expanded upon by Macklin et al. [8] to add more parameters, solving for a disconnect between the correlation of the number of physics calculation iterations and the rigidity of objects.

The use of soft bodies is integral to creating a high-fidelity surgical simulation. XPBD models body parts and suture threads in nearly all modern surgical simulators. Researchers like Liu et al. [33], Jourdes et al. [6], Korzeniowski et al. [20], and Qi et al. [19] make use of XPBD either directly or indirectly. Liu et al. [33] use XPBD to model a rope that they took a video of in real life to match its properties. Their motivation was to create a rope that could model a suture thread for robotic surgical simulators. Like Liu et al. [33], Qi et al. [19] modeled

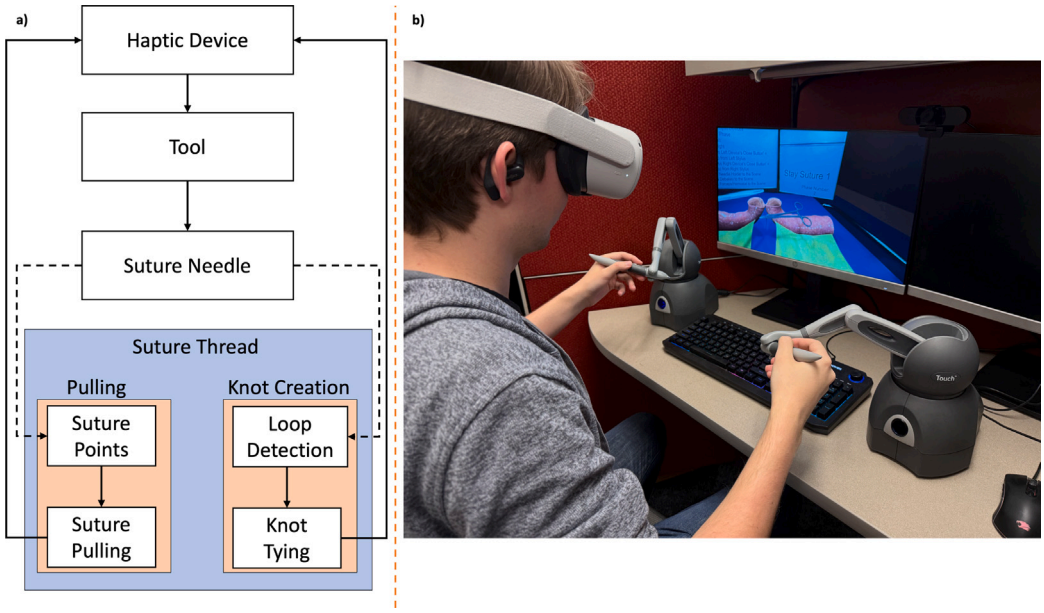

Fig. 1. (a) Framework architecture and (b) Simulation setup.

suture threads for surgical operations. However, their goal included the use of tying knots to complete sutures, which solved an integral step in surgical simulation. Their solution involved loop detection by examining projected line segments on a flat plane to identify a closed loop and thus initiate the following actions, like creating a knot. On the other hand, Korzeniowski et al. [20] used XPBD to model a section of flesh for excision. This method used shape-matching constraints to maintain the given mesh and provide a soft body representation that could deform but return to its original shape when no outside force is acting. They were able to implement cutting and sectioning portions of the soft body to replicate their intended procedure, leading to a higher fidelity simulator. However, none of the prior systems combine all the elements we target real-time soft-tissue deformation, continuous suturing with loop/knot dynamics, and bi-manual force-feedback interaction in a single open-surgery simulation platform. Our work builds directly on this literature to fill that critical gap.

2. Methodology

The real-time soft body suturing process has four main components: (a) Haptic Devices, (b) Surgical Tool, (c) Suture Needle, and (d) Suture Thread. We use two 3D Systems Touch haptic devices for bimanual tool control: One controlling a needle driver, and the other controlling forceps. Each virtual surgical tool in the 3D environment can manipulate a suture needle attached to a virtual suture thread. The framework architecture and simulation setup are illustrated in Fig. 1.

The overall simulation architecture is modular, comprising an input subsystem (two haptic devices and a VR headset capturing position and rotation), a physics subsystem (an XPBD-based soft body and suture solver running in parallel with a rigid body physics engine), and an output subsystem (visual rendering to the HMD and force feedback to the haptic devices). The simulator is implemented in Unity 2022.3, using the XR Interaction Toolkit for VR integration and Haptics Direct to interface with the 3D Systems Touch devices. Simulation space calibration of the haptic devices was performed with assistance from our expert surgeons, and the workspaces of the devices were scaled accordingly for spatial accuracy and consistency. The simulation state is orchestrated by the Procedure Manager, which coordinates updates across all modules. The module diagram illustrating the architecture can be seen in Fig. 2a. Fig. 2b presents the simulation loop from input capture through physics updates, collision handling, and rendering.

2.1. Soft body simulation

In our simulator, XPBD was used to accurately simulate the deformable bodies and suture thread. XPBD offers the strengths of position-based dynamics (PBD) with the pseudo-decoupling of iteration count from object stiffness, allowing for fine-tuning of object behavior and more performant simulation. By representing the shapes of the colon and suture thread as particles, individual behaviors could be created to represent the desired objects.

The suture thread is modeled as a series of particles connected by distance constraints to maintain a fixed segment lengths and bending constraints to control the thread's flexibility by penalizing changes in the angle between consecutive segments. The distance constraint between two adjacent particles p_i and p_j is given by $C_{dist}(p_i, p_j) = \|p_i - p_j\| - L_0 = 0$, where L_0 is the rest length. The bending constraint for three consecutive particles (p_{i-1}, p_i, p_{i+1}) uses the cosine form $C_{bend} = u \cdot v - \cos\theta_0 = 0$, where $u = \frac{p_{i-1}-p_i}{\|p_{i-1}-p_i\|}$, $v = \frac{p_{i+1}-p_i}{\|p_{i+1}-p_i\|}$, and θ_0 is the rest angle.

Meanwhile, the colon is modeled as a soft body composed of an oriented particle network with a shape-matching constraint to preserve its volume and original mesh shape. For each particle cluster, the current centroid c and rest centroid c_o are computed, and the cross-covariance matrix $A = \sum_k m_k(x_k - c)(q_k - c_0)^T$ is constructed from current positions x_k and rest positions q_k. Polar decomposition of A yields the optimal rotation R, which is used to compute the goal positions $g_k = c + R(q_k - c_0)$. The shape-matching constraint is then $Cshape$, $k = x_k - g_k = 0$ applied per coordinate as seen in Fig. 3. Where explicit volume preservation was needed, each tetrahedral element enforced the constraint $C_{vol} = V - V_0 = 0$, with $V(x_0, x_1, x_2, x_3) = \frac{1}{6}[(x_1 - x_0) \times (x_2 - x_0)] \cdot (x_3 - x_0)$.

The custom XPBD solver was implemented to run in parallel with a rigid body physics solver responsible for the surgical instruments and needle. It is fully multi-threaded and optimized to maximize CPU utilization during constraint solving and collision detection. Particles are positioned at key vertices of each object's mesh, and the visible mesh follows these particles via linear skinning with adjustable influence radii and falloff, allowing the colon and suture thread to deform naturally while maintaining their intended anatomical and mechanical properties.

In our implementation, solver parameters were tuned to balance stability and responsiveness. Shape-matching constraints were applied

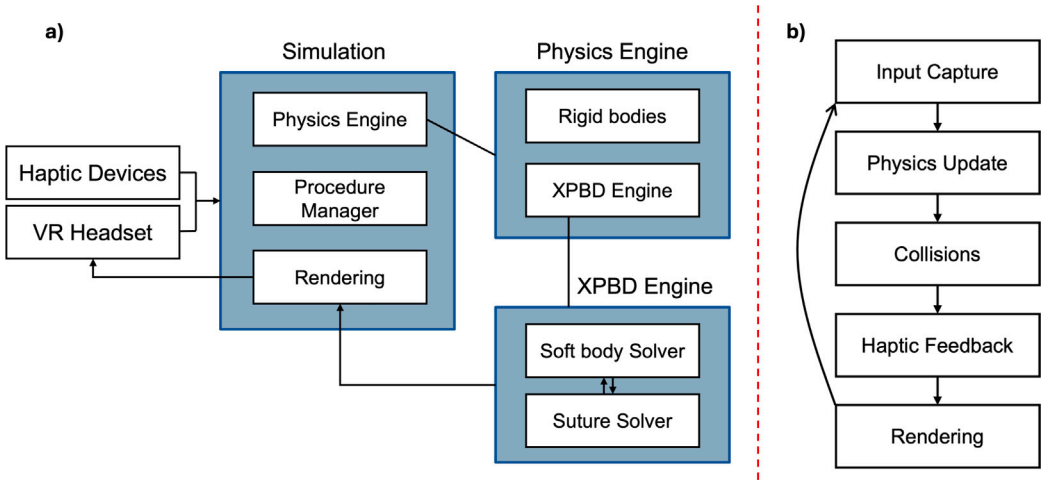

Fig. 2. (a) Module diagram and (b) Simulation loop diagram.

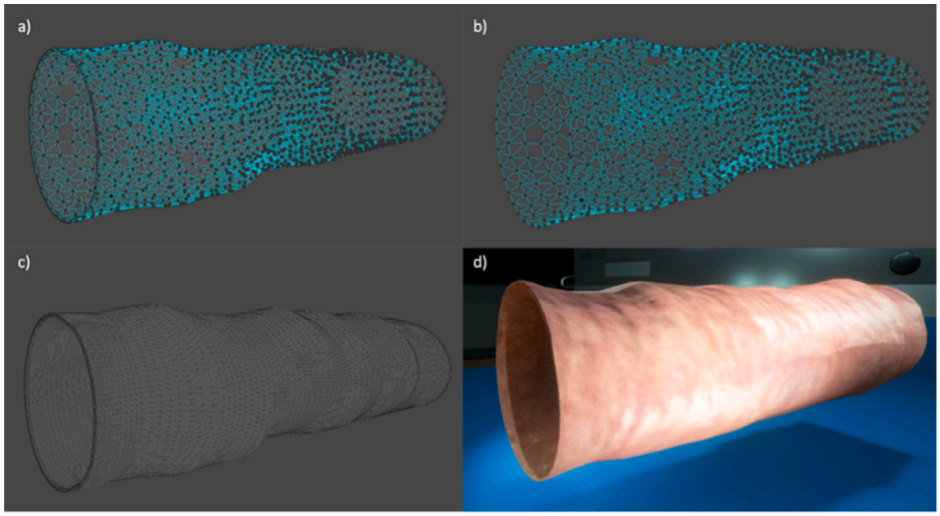

Fig. 3. (a) Generated particles with shape-matching constraints on the generated deformable colon mesh, (b) Generated wireframe model with shape-matching constraints, (c) Generated deformable colon mesh, and (d) Rendering of the surface soft body in the virtual scene.

with a deformation resistance of 0.65 over two iterations per solver step. Bending constraints used a compliance of 0.02 with three iterations, and distance (chain) constraints were solved in three iterations. Particle-based self-collision constraints were iterated 10 times per step, followed by five iterations of general collision handling. The solver computed the simulation using eight sub steps per frame to improve stability during tool interactions.

2.2. Needle simulation

The needle is simulated as a rigid body that the user manipulates via the haptic device. Colliders on the needle's tip (N) and back-end act as trigger colliders that registers a puncture event when they collide with a soft body. As soon as the needle's tip penetrates the soft tissue, the closest soft-body particle within a specified threshold distance X of the tip is identified and attached to the needle as illustrated in Fig. 4. We implement a simple nearest-neighbor search to find this particle, which is sufficient given the moderate number of particles in our organ models. Once the closest particle is found, it is constrained to move with the needle tip.

After a particle is attached to N, the soft body constraints cause the surrounding tissue to deform in response. As the needle pulls the attached particle, that particle in turn pulls on its neighboring tissue particles via the colon's shape-matching constraints (as seen in the four adjacent particles shown in Fig. 5a), producing a localized stretching of the soft tissue. When the needle has dragged the attached particle a certain distance beyond its original position, exceeding a preset distance threshold, the particle's attachment to the needle is automatically released, allowing the previously stretched region of tissue to relax back toward its original shape under the shape-matching constraints. This mechanism simulates the tissue resistance during puncture and ensures that after the needle passes, the local tissue returns to its anatomic shape (Fig. 5b). Once the needle's back-end trigger also passes through the tissue, the system marks the completion of that "bite" of the suture.

When a bite is completed, the needle creates a suture point (SP) and attaches the visual guide to the closest particle found earlier. Fig. 1 shows the process of creating a suture point. In the figure, TP stands for tip point of the suture needle. These suture points are the main component of the suture-pulling system.

The system provides force feedback to the user through the haptic devices in real time. Whenever the needle tip is attached to a soft-body particle during a puncture, a reaction force is computed based on the constraint between the needle and that particle. This force is modeled as a spring-like resistance proportional to the penetration depth and is fed back to the user via the haptic device, allowing the user to feel the tissue's stiffness. We tuned the stiffness coefficient (1100 N/m)

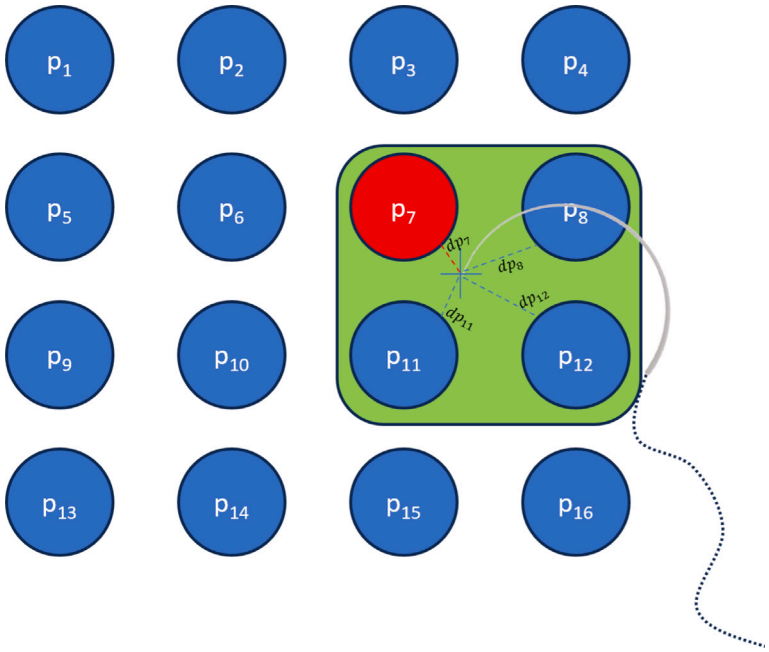

Fig. 4. The needle tip touches a soft body, triggering a contact event. The green square represents the 'threshold range' for the closest particle. The blue circles represent the particles in the soft body $p_1, p_2, \ldots, p_{16}$. The red circle is the closest particle to the needle contact point. The dotted lines show the distances to the particles in the region of interest. The particle p_7 will be attached to the needle as it is identified as the closest particle to the needle tip.

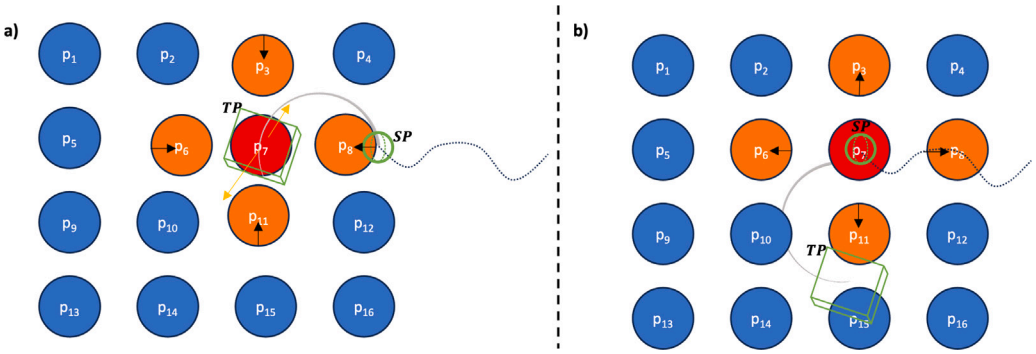

Fig. 5. (a) The tip of the needle N collides with the soft body. This spawns the visual guide attached to the back of the needle. The particle p_7 is found as the closest particle and is attached to the needle. This affects the forces exerted on the orange particles (p_3, p_6, p_8, p_{11}) as p_7 follows the needle. The arrows denote this stretching in the soft body. (b) As the back-end trigger of the needle collides with the soft body, the visual guide is attached to p_7, and p_7 is added to the suture points. Simultaneously, as N moves away from its original collision point, the orange particles are released and are no longer influenced by p_7's stretched movement.

for this virtual spring to produce realistic resistance without causing instabilities in the haptic loop. As the user pulls the needle and the suture thread through tissue, tension in the thread similarly generates force feedback proportional to the thread stretch.

2.3. Suture thread simulation

The suture thread is modeled as a series of ordered particles connected with distance and bend constraints. If the thread contains particles $p_1, p_2, p_3, \ldots, p_n$, each adjacent pair (p_i, p_{i+1}) is linked by a distance constraint to maintain a specified separation, while bend constraints act on triplets (p_i, p_{i+1}, p_{i+2}) to control flexibility. A line is then rendered between each particle to visualize the thread. Fig. 6 visualizes the particles that make up the rope and its final rendered version.

Suture threading and pulling behavior are driven by the suture points that the needle places on the soft body. When a new suture point is created, the simulator temporarily anchors the next free particle at the needle's end (p_n) to the suture point. This locks the thread at that location, allowing tension T to build between the anchored particle and the particle still attached to the needle as it moves away. Once T exceeds a predefined threshold, the simulator advances the thread through the suture point by repositioning the trailing particle to a midpoint between p_n and the anchored particle p_{n-1}. This effectively passes one thread segment through the tissue, moving p_{n-2} in front of the suture point as illustrated in Fig. 7.

Moving a particle to a different position in the suture thread requires shifting which particles are paired with each other so that the thread continues to simulate correctly. This is done in a re-calculation process to place pairs back in order.

The pair of particles with the particle to move (pp_m) is identified to start recalculating the rope. The pair in which the particle moves to target pair (pp_t) must also be identified. If pp_m is not the first pair in the rope (pp_0), the second particle in the next available pair (pp_{m-1}) is set to the second particle of pp_m. A new pair is created where the first particle in the pair is the first particle in pp_t, and the second particle in the pair is the particle that moved. This pair's location is set to be before pp_t in the order of the rope. The first particle in pp_t is set to the

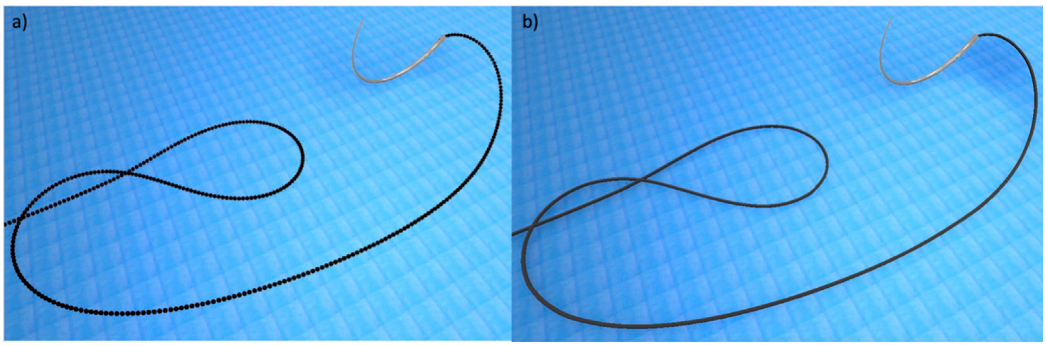

Fig. 6. The suture line attached to the suture needle (a) with and (b) without visualization of the simulated particles.

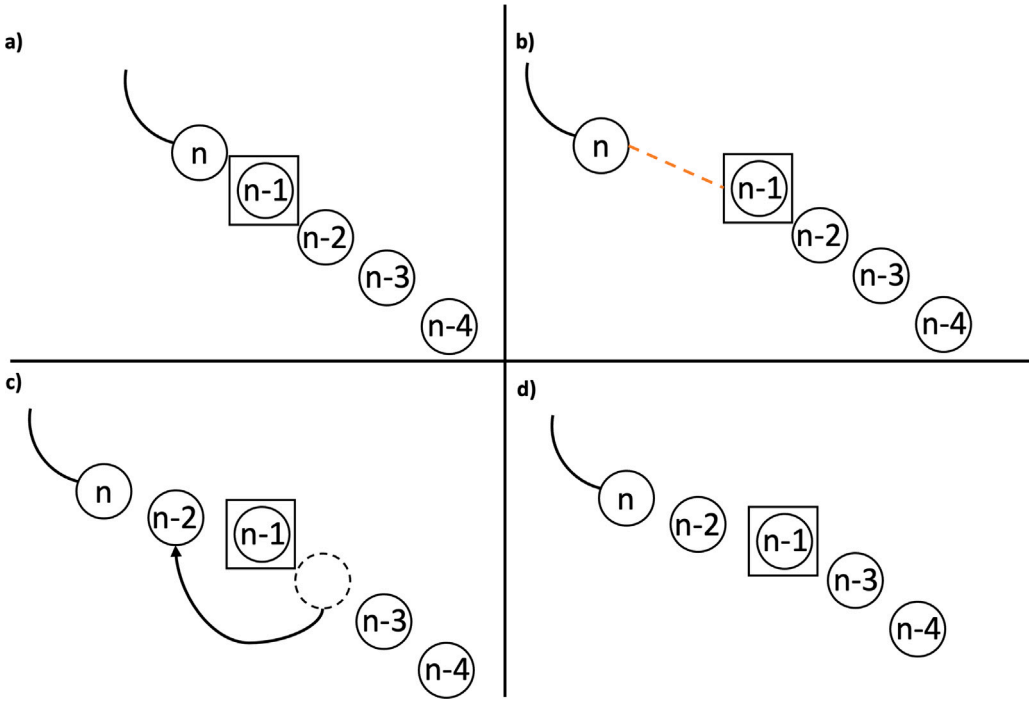

Fig. 7. (a) A suture point is created on the second closest particle to the needle, represented by the square. (b) The needle moves away from the suture point. The dotted line represents tension in the thread. (c) The particle on the free end of the suture point moves to the midpoint of the needle end particle p_n and the suture point p_{n-1}. (d) Tension is resolved after particle movement.

particle that was moved. After this is completed, pp_m is removed from the rope order. The rope is now recalculated with pairs in ascending order and simulates normally. Fig. 8 visualizes this process.

The suture points also manipulate collisions for the particles in the thread. When a suture point is placed, the locked/attached particle is set to no longer collide with anything, along with a set number of particles on either side. For example, if p_{10} is locked to a suture point, p_{11} through p_{13} and p_7 through p_9 will have collision disabled along with p_{10}. Turning off the collision prevents the suture thread from having erroneous physics collisions with the soft body it is sutured to. This gives the appearance of the thread going through the soft body while not permitting excessive physics altercations between the thread and the soft body. A particle's ability to collide is restored as the thread is pulled through the suture point, and the particle pairs are updated.

Pulling the thread becomes more complex as additional suture points are added to the soft body. To ensure realistic pulling, slack in the thread between any two suture points must be resolved before pulling the thread through. Our simulation checks for slack sequentially from the suture point closest to the needle to the free end of the thread. If the length of the thread between two suture points is greater than the length between the suture points themselves, sp_1 and sp_2, the particle on the free end of the thread adjacent to sp_2 is moved, and the suture pulling system goes into effect. If no slack is identified, the particle on the free end of sp_1 is transferred to the needle end of sp_2 as seen in Fig. 9.

2.4. Loop detection and knot tying

An integral step of suturing is the ability to tie knots. This is accomplished by detecting valid loops and creating a knot from that loop. We adopted the loop detection approach introduced by Qi et al. in [19] to identify when the suture thread has encircled a tool. Every particle in the thread colliding with the desired tool is projected to a 2D plane perpendicular to the tool. Each particle pair on the 2D plane is tested with every consecutive pair to see if the two lines between the pairs have an intersection. For instance, if $p_1, p_2, p_3, \ldots, p_n$, are colliding with the tool, particle pair one, pp_1, would contain p_1 and p_2, and pp_2 would contain p_2 and p_3. The particle pair pp_1 is compared to pp_2 through pp_n. If there is an intersection between pp_1 and any following pair, on the 2D plane, there is a loop within the rope [34] and the

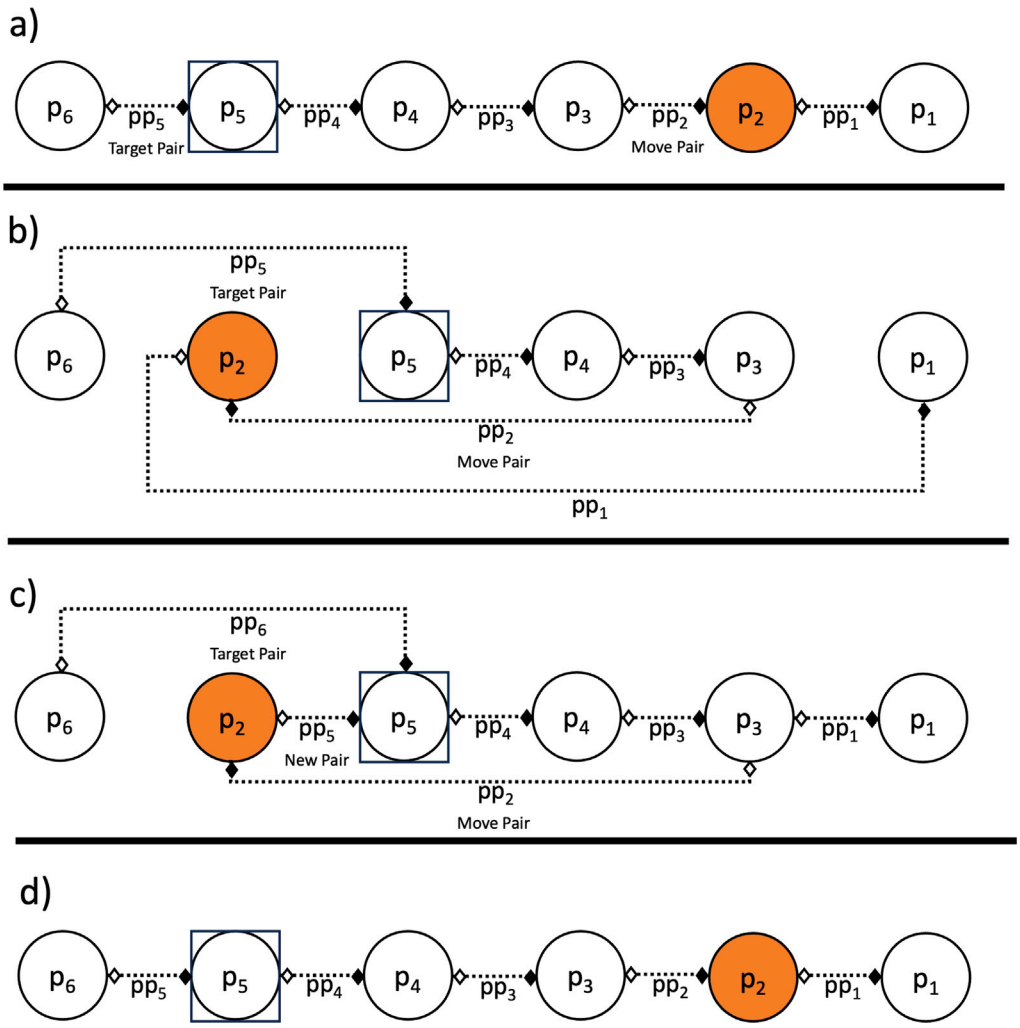

Fig. 8. (a) $p2$ is identified as a particle to move (shown by red color) and will be moved after p_5 (shown by the square.) pp_2 is identified as the moving pair, and pp_5 is the target pair. Each pair (shown by dashed lines) contains two ordered particles: an empty diamond shows particle 2, and a filled diamond shows particle 1. (b) p_2 is moved between p_5 and p_6 in 3D space. (c) A new pair is created at the index of pp_5 with particle 1 from both the target and move pair in that order. (d) The move pair is removed from the list, shifting the number of all subsequent pair indices down by one, and particle 1 of the target pair is set to the moved particle.

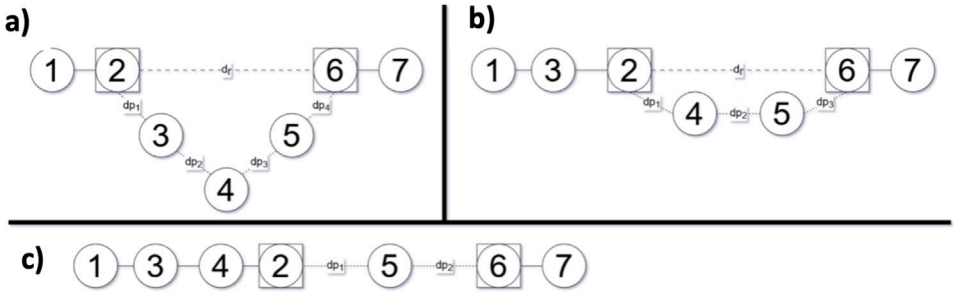

Fig. 9. (a) p_2 and p_6 are both attached to suture points, denoted by squares. Since the summed distance of dp_1 through dp_4 is greater than the distance between the two suture points, d_r, p_3 will be chosen for pulling. (b) Since the summed distance of dp_1 through dp_3 is greater than the distance between the two suture points, p_4 will be chosen for pulling. (c) Now, dp_1 and dp_2 summed equal the distance between the two suture points. Therefore, there is no slack left between the two suture points, and pulling will happen in the thread after p_6.

loop's particles are frozen on the tool. This process is iterated through the entire rope and all its particle pairs.

Two checks are integrated to ensure the loop detection is valid. The first check determines if the loop encircles the tool tip. All particles in the loop are verified to collide with the tool but may not necessarily encircle it. The tool is projected onto the 2D plane from the tip to determine that the loop encircles the tool tip. The even–odd rule, an algorithm used with vector-based graphics, determines if the tool is within the loop [34]. The second check is to verify the depth distance (dp_{1-n}) between the first (p_1) and last particle (p_n) in the loop within 3D space. If the distance is more than one particle pair away, the loop is not desired for tying a knot. When both checks are identified as true, all the

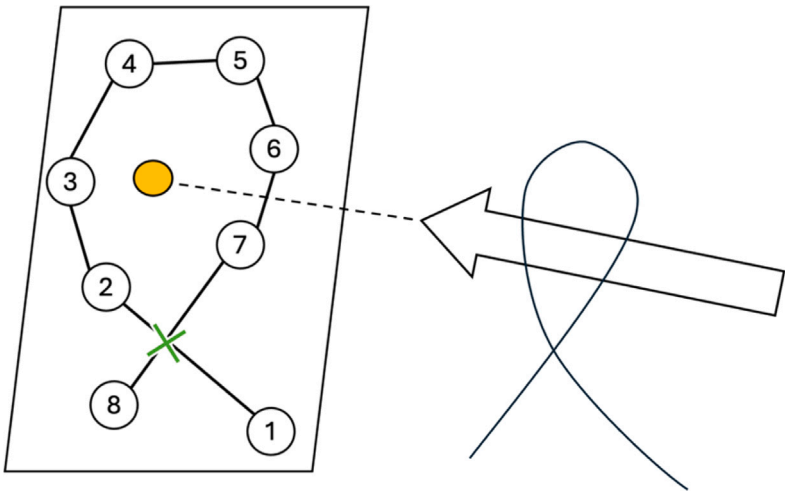

Fig. 10. The thread is projected onto a 2D plane, perpendicular to the tool. A loop is detected since the line from particle pair pp_{12} and pp_{78} intersect at the green X. The tool is projected to the 2D plane from the tip and checked to see if it is within the loop.

particles in the loop are attached to the tool using particle attachments. This prevents the user from undoing the loop. Fig. 10 demonstrates loop detection.

Knots can be created when at least one valid loop and suture point are detected. If the tool that has the loop around it grabs the free end of the rope passed the loop, a 3D object that represents the knot is created. The loop's ends are attached to the object, and the whole loop detaches from the tool. As the tool moves away from its original position, the knot object moves towards the suture point, simulating the tightening of the knot. If more than one suture point has been created, the knot object will move toward the midpoint of the last two created suture points. When the knot object reaches the suture point or midpoint between the two previous suture points, it is locked in place, and the attached particles are released. If the user decides they do not want to proceed with finalizing or tightening the knot, they can release the free end of the thread. After letting go, the loop will be released, and the knot object will be destroyed.

With a knot created and tightened, the suture-pulling system is affected. When the suture pulling checks where to move a particle, it will stop at knots. If there are no knots on the thread, the suture pulling will check from the needle end to the free end for slack between suture points, and if none is found, it will pull from the free end. With a knot created, the suture pulling will perform the same slack check from the needle end but will stop checking when the knot is found. This ensures slack is resolved up to the knot, respecting that the thread cannot be pulled past the knot.

2.5. Utilization of suture pulling and knot tying

The suture pulling and knot tying components combined can be utilized in various ways to simulate different surgical suturing techniques. In this work, we implemented Purse-string suture, Connell (running) suture, Lembert suture, and stay suture. The supplementary video demonstrates each of these in our simulation. Each technique uses the same solver, needle puncture, suture point placement, thread pulling, loop detection, and knot tightening with slight variations in logic to achieve the desired outcome. In our implementation, the purse-string suture involves approximately 1403 simulated particles (one soft body with 1228 particles and a thread with 175 particles). The Connell and stay sutures each involve about 2471 particles in total (two soft bodies of 1185 particles each, plus a 101-particle thread). The Lembert suture is the most complex, with about 4155 particles (a single soft body with 4054 particles representing the bowel layers, and a 101-particle thread).

2.5.1. Purse-string suture

A purse-string suture is created by running the needle in and out of the bowel, like a sinusoidal wave pattern, making a continuous loop around the entire opening. In our simulation, this is implemented by placing a series of suture points evenly around the circumference of the colon opening, without pulling the tissue inward until the very end. To ensure the thread forms a continuous loop around the opening without cutting across the center, the simulator checks that each new suture point lies adjacent to the previous one along the bowel edge. The thread does not cross the opening when sp_{i+1} is the closest suture point to sp_i in 3D space from sp_{i+1}, sp_{i+2}, and sp_{i+3}. If there are not three consecutive points after sp_i, then sp_0 through sp_2 are used. Once suture points are placed all around the opening, the set of edge particles are all pulled inward simultaneously toward a common focal point, the anvil, when the thread is tightened. Fig. 11 demonstrates this purse-string tightening, all edge points (blue circles) are drawn toward a central anvil point (red circle) over time, closing the opening like a purse string.

2.5.2. Connell (running) suture

Connell suturing is a continuous running suture technique to join two tissue edges. In our simulation, we suture two segments of the bowel (soft bodies) together in an alternating pattern. The needle is passed through one soft body (creating a suture point), then through the same soft body again for a second bite, and then the needle crosses over and goes through the other soft body, and so on in an alternating sequence (Figs. 12a–d). This means multiple suture points can occur on the same soft body before the thread crosses to the other. Our system keeps track of which soft body the last two suture points were placed on. If the two most recent suture points are on the same soft body, the thread is simply pulled through without closing the gap between tissues yet. If the two most recent suture points are on different soft bodies, the system automatically pulls those two tissue points toward each other. In practice, when the user makes a suture point on the second bowel after one on the first bowel, the simulator calculates the midpoint between the last suture point on the first soft body (SB_1) and the new suture point on the second soft body (SB_2), and then pulls both of those points toward that midpoint (Figs. 12c–d). This simulates the action of tightening the two tissues together after crossing over.

Midpoints are saved to a list (mp), and if mp contains a midpoint when a new one is added, pulling is done towards the midpoint of these two midpoints. A line is created connecting the two points and is segmented into segments (s) where ($s = (mp_i - mp_{i-1})/dist_{threshold}$). One particle on each SB closest to a segment is identified and paired

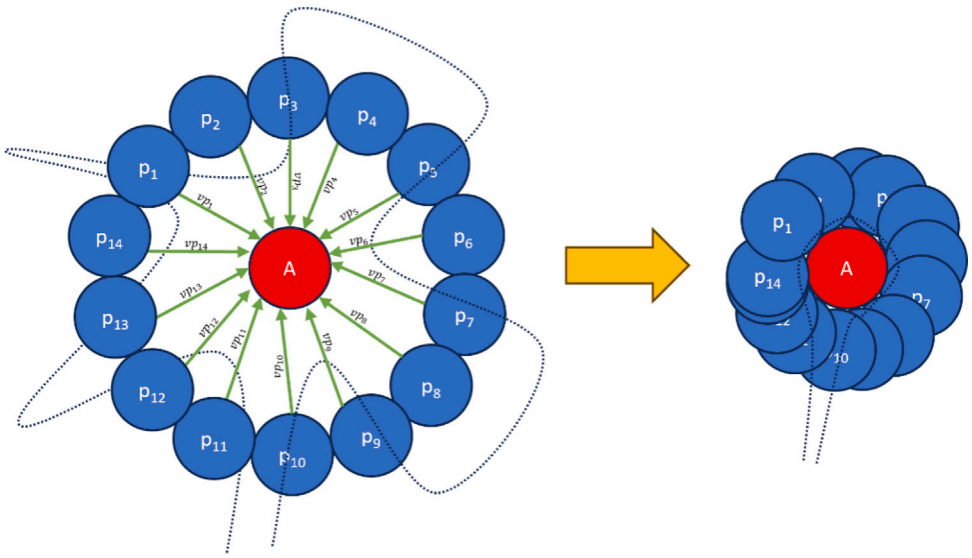

Fig. 11. The particles around the opening, denoted by the blue circles, are identified. Once identified, they are pulled towards the anvil, denoted by the red circle. Each particle is pulled over time, ensuring all particles arrive at the location simultaneously.

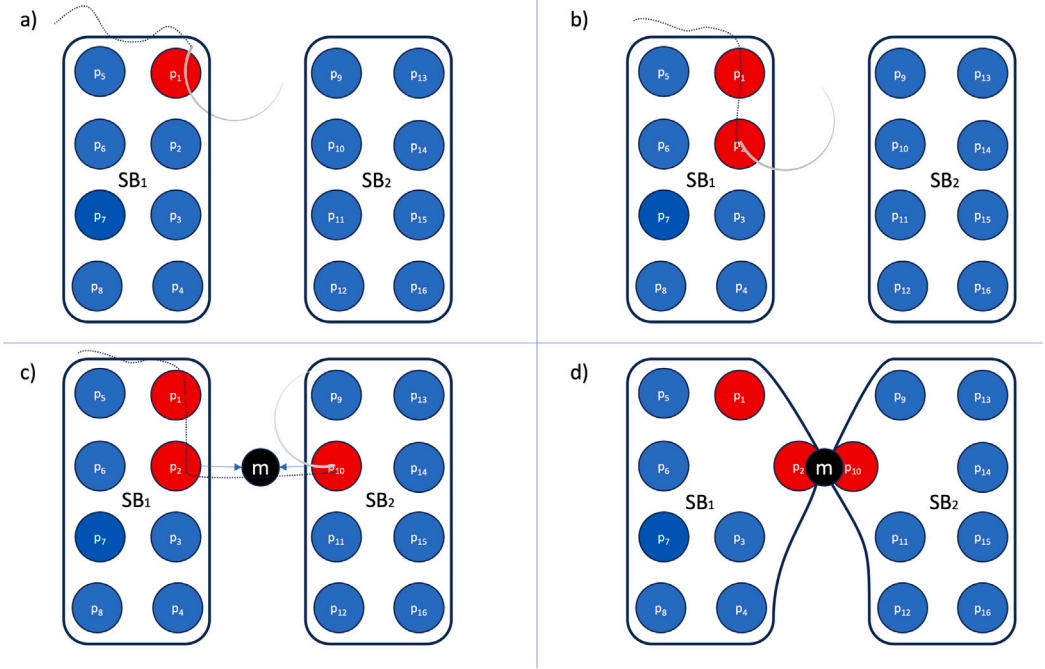

Fig. 12. (a) SB_1 and SB_2 denote two soft bodies. The needle places a suture point on $p1$ in SB_1. (b) The needle places a suture point on p_2 in SB_1. (c) The needle places a suture point on p_{10} in SB_2. Since the last two suture points were placed on different soft bodies, the midpoint of the two suture points is calculated. (d) p_2 and p_{10} are pulled to their midpoint.

at each segment point. Each pair is pulled to the midpoint of the pair as the needle pulls the thread through. Pulling these extra particles replicates how human tissue deforms during the suturing process. Fig. 13 demonstrates an example where two midpoints (mp_1 and mp_2) have been formed, and intermediate tissue particles are pulled together between them. After placing the desired number of running sutures, the user ties a knot to secure the continuous suture completing the Connell suture.

2.5.3. Stay sutures

Stay sutures are created with a single bite through two different bowels, and a knot is tied to complete it. Like the Connell suture, two bowels are pulled together to a midpoint from where the suture points are created. The same method of pulling as the Connell suturing is used. Once the two suture points have been pulled together, the stay suture must be tied off. Once the knot is tied, the thread is disconnected from the needle. A tool can grab the two ends of the thread and hold it as a stay suture. Stay sutures are depicted in Fig. 14.

2.5.4. Lembert sutures

Lembert suturing was utilized to simulate the suturing of a cut piece of bowel after a prolapse was removed. This required a single bite through one soft body. After one suture point is created, it is tied with multiple throws. When a knot is pulled tight, the inner and outer layers must be pulled together around the created suture point. If multiple

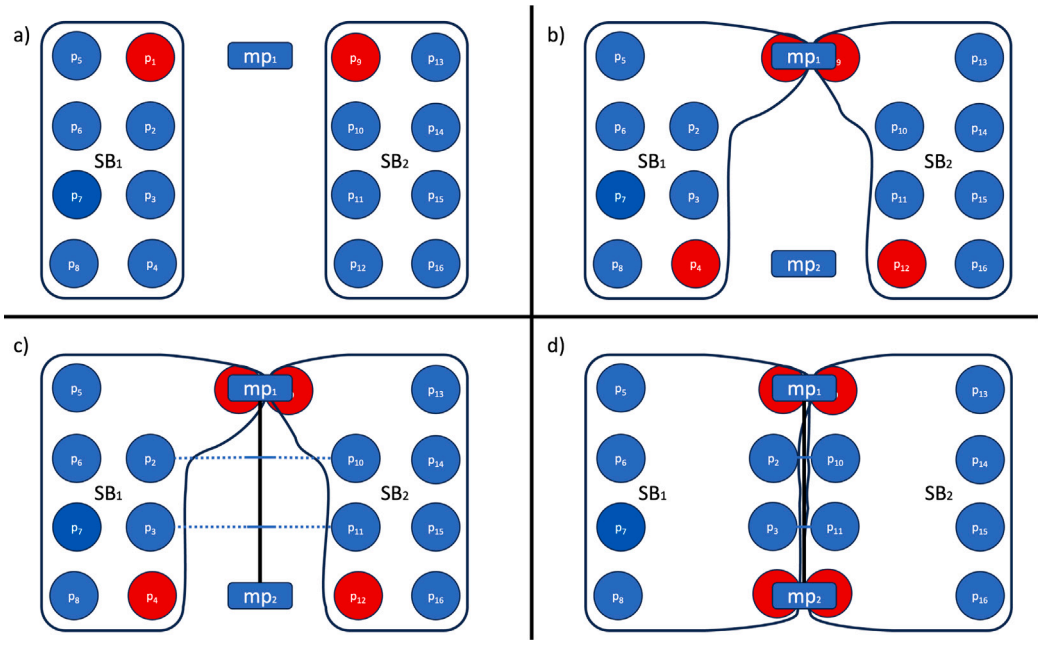

Fig. 13. (a) A suture point is created on p_1 and p_9, denoted by the red particles. A midpoint, mp_1, is found between the two particles since they are on different soft bodies. (b) p_1 and p_9 are pulled toward mp_1 as suture points are created on p_4 and p_{12}. The midpoint, mp_2, is found from these two new particles. (c) Since two mp have been found, a line between mp_1 and mp_2 is found and then segmented. The closest particle to each segment from both soft bodies is found. (d) The suture points p_4 and p_{12} are pulled toward mp_2. At the same time, the particles for each segment are pulled together.

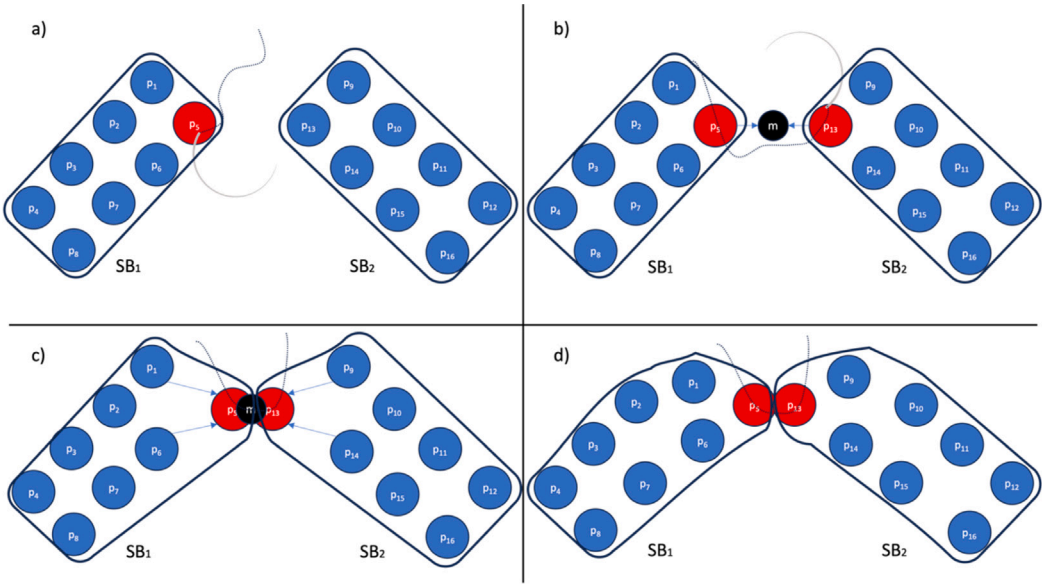

Fig. 14. (a) SB_1 and SB_2 denote two soft bodies. The needle places a suture point on p_5 in SB_1. (b) The needle places a suture point on p_{13} in SB_2. Since the last two suture points were placed on different soft bodies, the midpoint of the two suture points is calculated, and they are pulled to it by the user moving the needle. (c) After pulling, the surrounding particles in each of the soft bodies experience forces to follow the pulled particles to keep the shape of the soft body. (d) The shape is resolved, terminating external forces and allowing the thread to go through both soft bodies, readying it to be tied into a knot.

Lembert sutures are made close together, the bowel's flesh between the two sutures should also be pulled closed.

Two particle groups (pg) are created for the inner and outer bowel layers: pg_i and pg_o. When a suture point (sp_1) is placed, the closest particle in pg_i and pg_o is found and they are pulled to their midpoint when a knot is tied. When a subsequent suture point (sp_2) is placed, particles in pg_i and pg_o between the two suture points are pulled together. This is done by finding the center point (cp) of all particles in pg_i and pg_o and calculating the angle between the vector (v_1) from cp to sp_1 and (v_2) cp to sp_2. The vector (v_p) is calculated as cp to a particle in either pg_i or pg_o. If the angle between v_1 and v_p and the angle between v_2 and v_p equals the angle between v_1 and v_2, then the particle is between sp_1 and sp_2. A particle between the two suture points is pulled toward its midpoint, and the closest particle in the opposite particle group. Fig. 15 shows the angle calculation for determining particles to pull.

3. Results

The suturing processes built with our method of suture pulling and knot tying were implemented into several procedures. The effectiveness

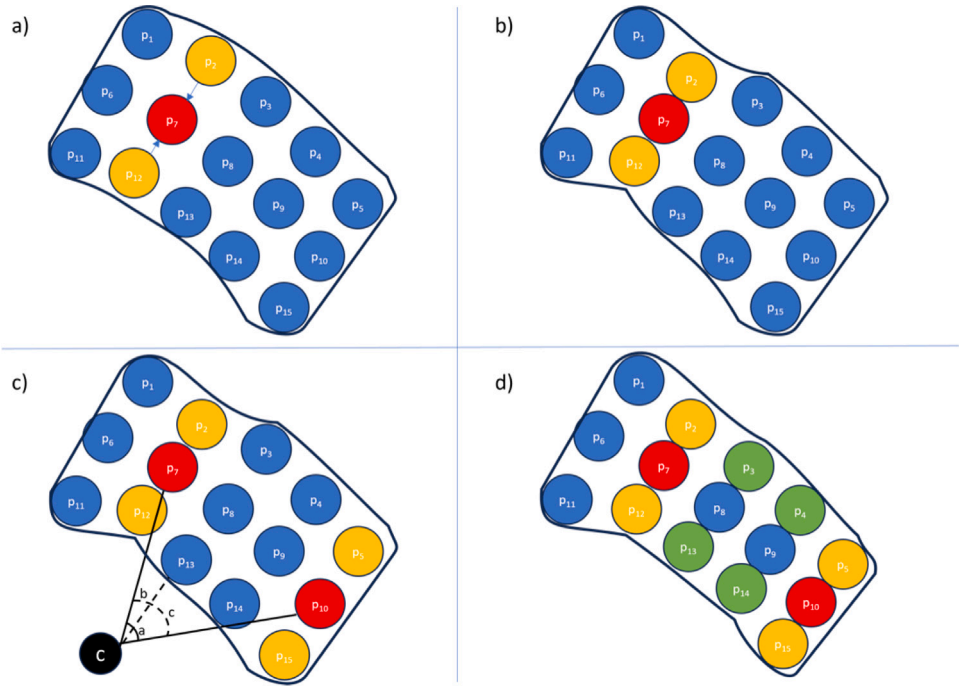

Fig. 15. (a) p_7 has a suture point placed on it, as shown in the red particle. p_2 and p_{12} are identified as the closest particles to p_7, as denoted by the gold particles. (b) p_2 and p_{12} are pulled toward p_7. (c) p_{10} has a suture point placed on it, with p_{10} and p_{15} being determined as the closest particles. p_{13} is checked to see if it is in between p_7 and p_{10}. The center of all inner and outer particles is denoted by the black circle, c. Since angle b and angle c, represented by dotted lines, are added together and equal to angle a, represented by solid lines p_{13}, it is determined to be between p_7 and p_10. (d) The process is repeated for all inner and outer particles. All particles between p_7 and p_{10} are pulled toward the midpoint of the closest opposite layer particle.

of the outcome was determined based on whether the suturing result was visually correct and whether the suturing and knot tying had any adverse effects on the overall performance of the simulation. The visual correctness of the suture was evaluated by an expert. Performance was measured with frames per second (FPS), render and solver computation times in milliseconds. All performance measure calculations were carried out with two haptic devices and a head-mounted display connected to the virtual scene. Solver computation time is the time for the XPBD solver to compute the physics, constraints, and collisions for all simulated particles.

These values were recorded at: (1) the start of the simulation when everything was at rest, (2) during suturing, (3) during loop detection, (4) during knot tying, and (5) at the completion of the procedure when everything returned to a resting state. For consistency, each scene was executed ten times, and the average results were reported. For each execution, the performance averages for each state are calculated. All tests were performed on an Intel(R) Core (TM) i7-13700KF 3.4 GHz CPU, 32 GB RAM, and NVIDIA GeForce RTX 4070 Ti. For interaction with the scene, we used two 3D Systems Touch haptic devices for tool control and Oculus Quest 2 for immersive visual feedback, enhancing the overall user experience and immersion within the virtual environment.

Along with performance, fidelity is also considered for success. A high frame rate should be maintained while visually replicating the real-life suturing technique of each suturing procedure within a virtual reality environment.

3.1. Purse suture

The first suture method to be tested was the purse suture. This simulation has 1403 particles (Soft Body: 1228, Suture Thread: 175) from one soft body for simulating the colon and one rope for simulating the suture thread. Suturing is performed by moving the needle in and out of the soft body. For this procedure, suturing was performed ten times. Table 1 shows the output performance for this simulation.

Table 1
Average performance data for a purse suture over ten iterations.

Purse suture			
	Start	During suturing	After completion
FPS	82.1	81.38	82.2
Render time (ms)	12.1	12.21	12.1
Solver time (ms)	5.42	5.61	5.08

From the performance data, it was found that the performance of the simulation was stable throughout the whole procedure, and the simulation was able to replicate the process of creating a purse suture. During active suturing, the system averaged 81.38 FPS (σ : 0.29), while the render time was 12.21 ms (σ : 0.07 ms) and solver time was 5.61 ms (σ : 0.03 ms). Fig. 16 shows the process of creating a purse suture within the simulation environment.

3.2. Connell (running) suture

Connell suturing was tested with two soft bodies and a suture thread. These objects had 2471 particles in total (Each Soft body: 1185, Suture Thread: 101). Suturing was performed by moving the needle into one soft body, SB_1, and then out of that same soft body, SB_1. The needle was then pushed through the opposite soft body, SB_2. This was repeated ten times, alternating between the two soft bodies. Unlike the purse suture, this simulation also requires a knot to be made. After the suturing, the performance of creating a loop and making a knot was recorded. Table 2 contains all recorded performance data. Similar to the purse suture, this data maintained consistent performance throughout the procedure. There was a slight decrease in performance during the looping phase, however simulation performance stayed at 65.7 FPS (σ : 0.32) in the VR scene. The slight FPS drop was caused by both the render (15.7 ms with σ: 0.04 ms) and the solver time (7.99 ms with σ: 0.04 ms). Fig. 17 shows the creation of a Connell suture. In the figures,

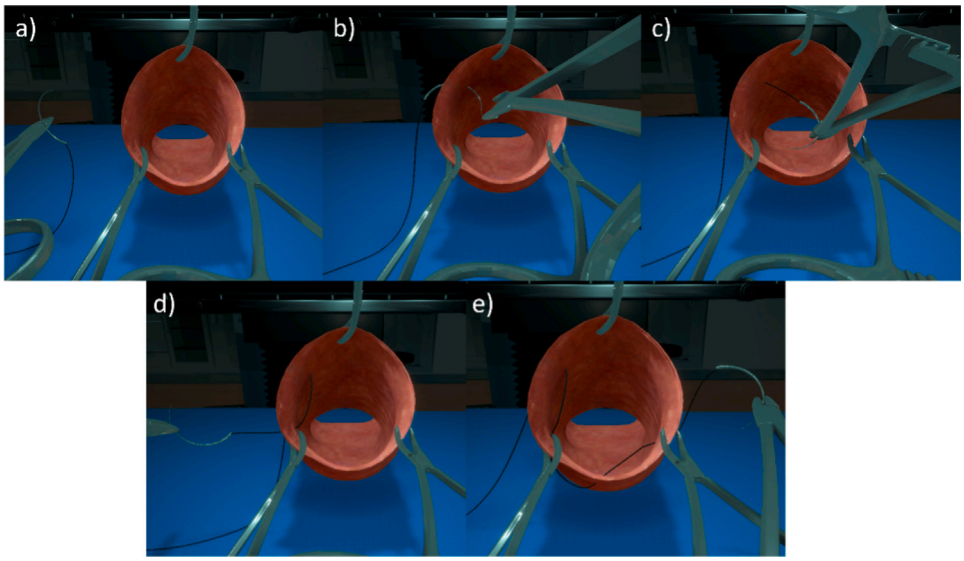

Fig. 16. (a) Preparation for suturing. (b) First puncture of soft body. (c) Pull the suture line through. (d) Performing the second suture. (e) Continued until a continuous suture is formed around the entire opening.

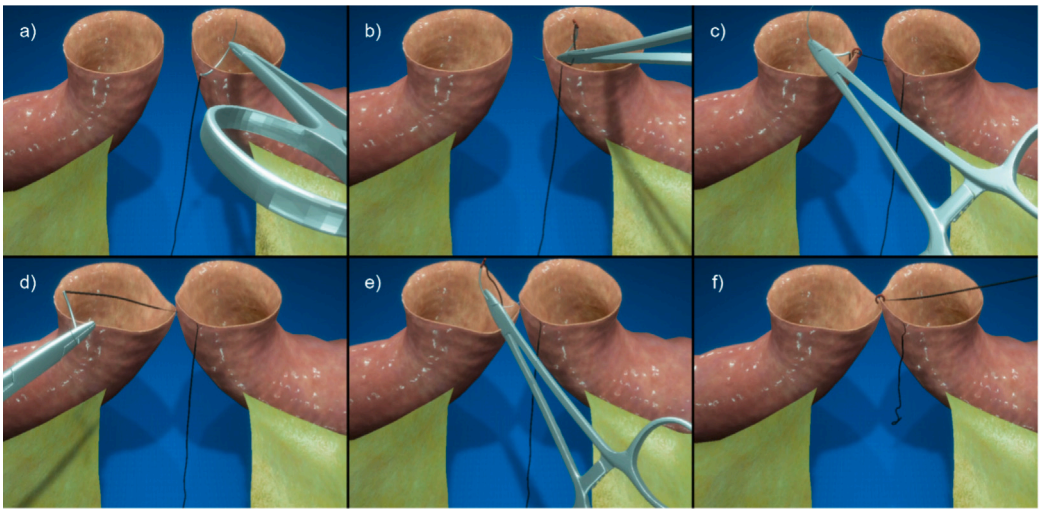

Fig. 17. (a) The needle is pulled inward through the right bowel, creating a suture point. (b) The needle is pulled back through the right bowel, creating another suture point. (c) The needle is pulled inward through the left bowel, creating a suture point. (d) As the tool moves away, the suture point on the right bowel and the last placed suture point on the left bowel are pulled together. (e) The needle is pulled through both the right and left bowel. A suture point is created on each. (f) The two new suture points are pulled together.

Table 2
Average performance data for Connell suturing over ten iterations.

Connell suture				
	Start	During suturing	During looping	After completion
FPS	72.3	70.1	65.7	70.2
Render time (ms)	13.8	14.2	15.7	14.3
Solver time (ms)	6.70	6.78	7.99	7.35

the visual guide indicates to the user that a suture point has started and that they can continue to make their sutures.

3.3. Stay suture

The stay suture was performed in the same simulation as the Connell suture with two soft bodies and a suture thread totaling 2471 particles (Each Soft body: 1185, Suture Thread: 101). Like the Connell suture, both suturing and knot-tying take place. The suturing consists of bringing the needle through both soft bodies (SB_1 and SB_2) once. The thread is then pulled through until both sides are even. Ten stay sutures were performed, and their performance was recorded. Afterwards, a loop is made, and a knot is created. Table 3 contains all the performance data for the stay suture procedure. Real-time performance and fidelity were maintained during this procedure while running in VR. Performance remains consistent throughout the whole process and is higher than Connell suturing. This is due to fewer particle collisions during the suturing as the stay suture only goes through each soft body once. The average FPS during suturing was 72.1 (σ: 0.30), and during looping, it was 72.6 (σ: 0.34). Fig. 18 demonstrates the procedure for stay sutures from within the simulation environment.

3.4. Lembert suture

Finally, Lembert sutures were tested. This simulation consisted of two soft bodies and one suture thread totaling 4155 particles (Soft

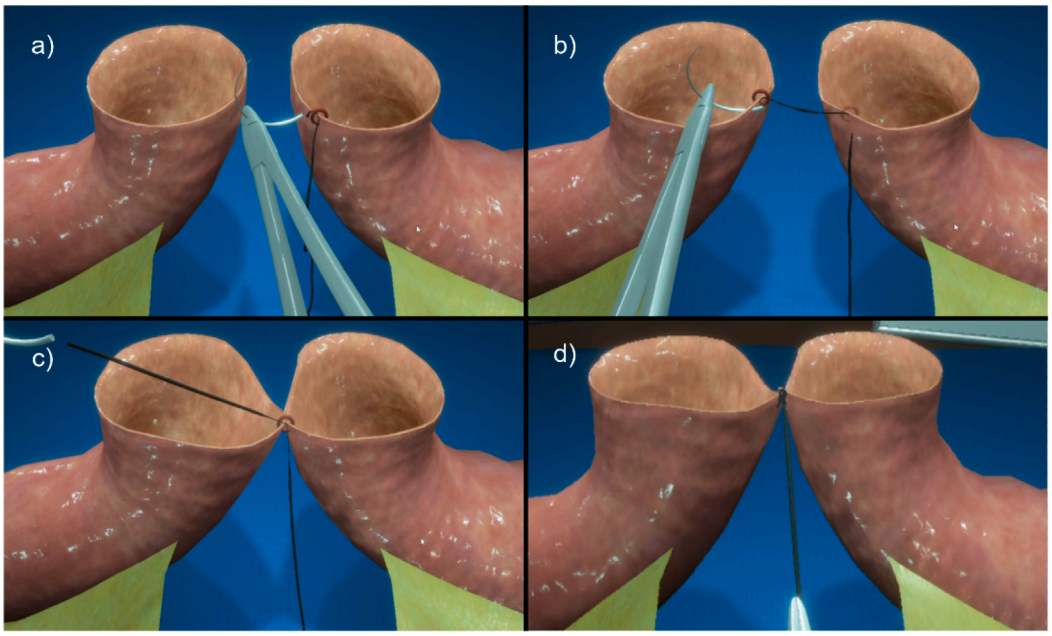

Fig. 18. (a) A suture point is created on the right bowel. (b) A suture point is created on the left bowel. (c) The points are pulled together as the tool is pulled away. (d) A knot is created, and the two free ends are grabbed.

Table 3
Average performance data for stay sutures over ten iterations.

Connell suture				
	Start	During suturing	During looping	After completion
FPS	76.5	72.1	72.6	73.1
Render time (ms)	12.9	13.8	13.82	13.6
Solver time (ms)	5.9	6.33	6.78	6.41

Body: 4054, Suture Thread: 101.) Each suture had a suturing step and a looping step. The performance is recorded once the needle is pulled through the soft body. After, a loop is made around the tool, and a knot is created. The looping and knot-tying performance is then recorded. These steps are repeated a total of ten times. Table 4 shows the output of the performance data that was recorded during each step. Performance is lower within this scene, likely due to the higher particle count of the soft body. However, the procedure was consistently performed throughout the whole process. This process replaces the thread with a 3D mesh when a knot is created. This is done so that physics is no longer performed on previous sutures. This is likely the cause of the consistent and even slight increase in FPS seen within this simulation. The average suturing FPS varied between 50.62 to 52.5 and average looping FPS varied between 52.0 to 53.2. The average render times (Suturing σ^2: 0.53 ms and σ: 0.73 ms, Looping σ^2: 0.7 ms and $\sigma = 0.84$ ms) and solver times (Suturing σ^2: 0.35 ms and σ: 0.59 ms, Looping σ^2: 0.28 ms and σ: 0.53 ms) showed low variability. Fig. 19 demonstrates the use of Lembert suturing within the simulation.

Overall, every suturing procedure maintained consistent performance through suturing and tying knots without significantly impacting performance after completion. Every suturing procedure ran entirely in real-time within a VR environment while maintaining high-fidelity visuals. Alongside the real-time performance, the suture-pulling and knot-tying procedures replicated the visual process of each outlined suturing procedure.

3.5. Overall soft body suturing performance results

In this section, we tested the limits of our suturing system. Performance with two soft bodies was determined by creating sutures until the simulation no longer ran in real-time. Real-time was considered when the simulation executes 24 per second or higher [35,36]. Figs. 20 and 21 show the suturing performance as each suture was placed. As seen in Fig. 20, the FPS in the scene is consistent for sutures 1–21. Between suture #21 and #22 there is a drop of 15 FPS. Between sutures 22 and 26 the decrease is linear, however at suture #26, the FPS of the scene two soft body scene goes below 24 FPS. The FPS drop is consistent with the solver and render time which is shown in Fig. 21. Solver time stays under 10 ms until suture #13 and at #26 the solver time $\approx$ 20 ms. The solver time increases during the simulation, which was expected as the suturing is being carried out. This is due to more collisions between particles are happening as the two soft bodies are pulled together. For render time, the biggest increase is between sutures 21 and 22, where the render time increases by $\approx$ 10 ms. This increase is consistent with the drop of FPS noted in Fig. 20.

Suturing with one soft body was also tested. Since there would be no collisions between two soft bodies, the limits for one soft body were tested for 50 sutures. The method used for testing was similar to purse suture, where the thread was moved in and out of the soft body. Figs. 22 and 23 show the performance of the suturing system with one soft body. Overall, the performance remained consistent throughout the whole process. The solver time increases as more sutures were added. At suture #50, the FPS was 76.2. The FPS change was linear and consistent with solver time. The render time stayed in the range of 12–12.2 ms during the suturing process, this shows that for one body suturing, render time stays consistent throughout unlike two soft body suturing.

3.6. Case study: Rectal prolapse simulator

We conducted a study (IRB information is anonymized) at a tertiary center in which surgery residents and expert colorectal surgeons were recruited to perform the Altemeier procedure [37] on the VR-based Virtual Colorectal Surgery Trainer (VCoST) [38–40] Rectal Prolapse (RP) simulator (as seen in Fig. 24), utilizing our real-time suturing solution. The anastomosis phase of the RP procedure demands a high level of suturing dexterity, as it is critical for restoring intestinal continuity by reattaching the resected prolapsed rectum to the anus.

The procedure was divided into four major steps. First, during the preparation phase as seen in Fig. 25, the rectal prolapse was secured

Table 4
Average performance data for Lembert suturing over ten iterations.

Lembert suture												
	Start	Suture #1	Loop #1	Suture #2	Loop #2	Suture #3	Loop #3	Suture #4	Loop #4	Suture #5	Loop #5	After completion
FPS	52.2	51.3	52.0	52.5	53.1	50.8	53.2	50.65	52.8	50.62	52.81	53.2
Render Time (ms)	19.0	19.2	19.3	19.2	18.8	19.54	18.6	19.73	18.95	19.48	18.62	18.8
Solver Time (ms)	11.47	11.80	11.85	11.60	11.68	11.69	11.65	11.95	11.93	11.84	11.74	11.45

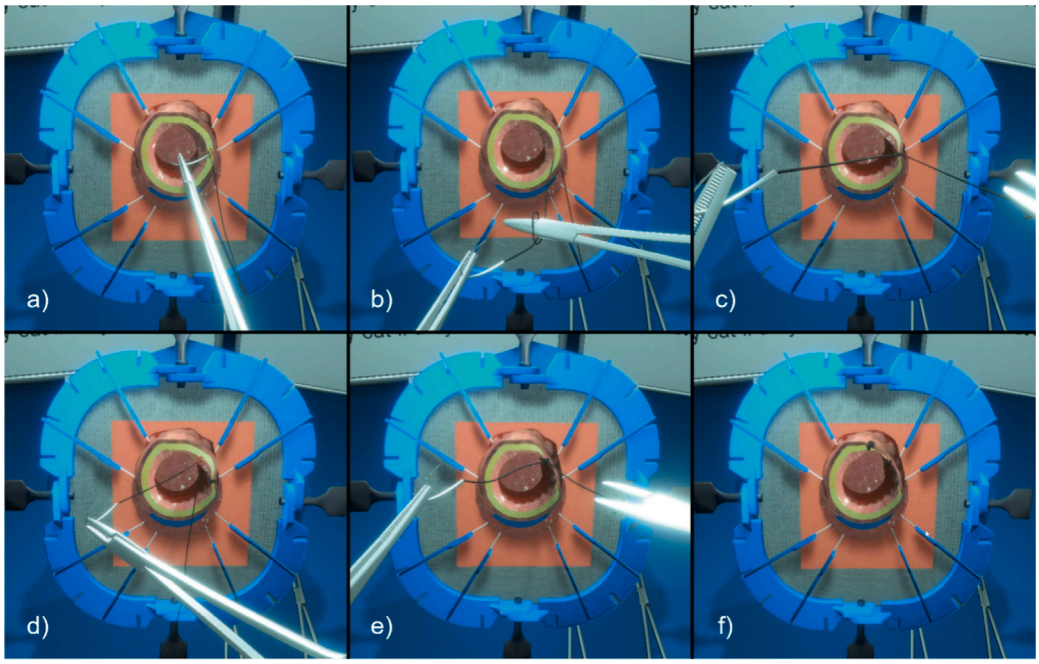

Fig. 19. (a) The needle is pulled through the inner and outer layers, and a suture point is placed. (b) A loop is created around the tool. (c) The free end of the thread is grabbed and pulled through the loop to create a knot. The inner and outer layers are pulled together by the knot. (d) The needle is pulled through the bowel to the left of the first suture. (e) A knot is created for the second suture point. (f) A third suture is created by repeating the whole process.

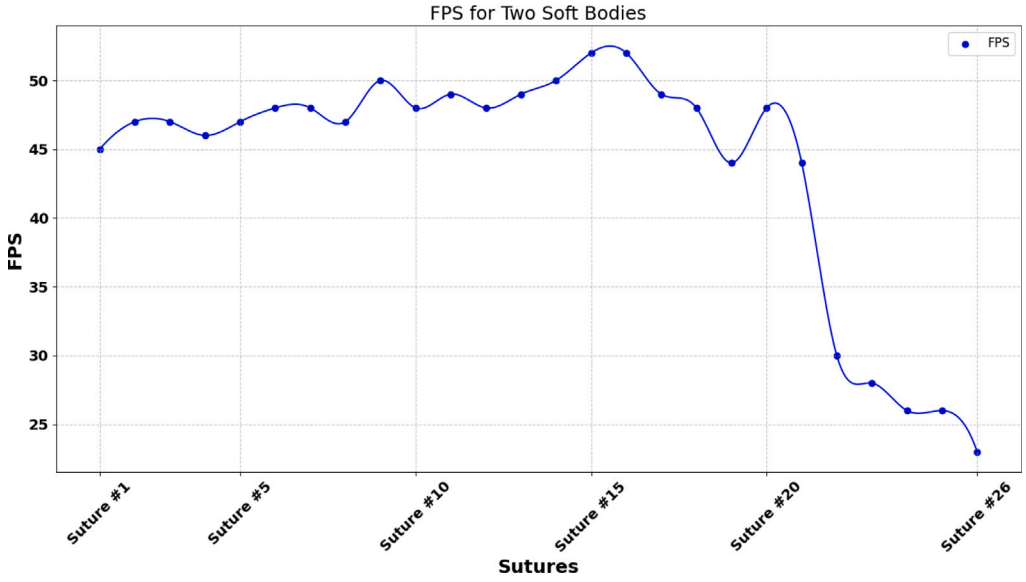

Fig. 20. Performance measured in frames per second of a simulation with two soft bodies sutured together.

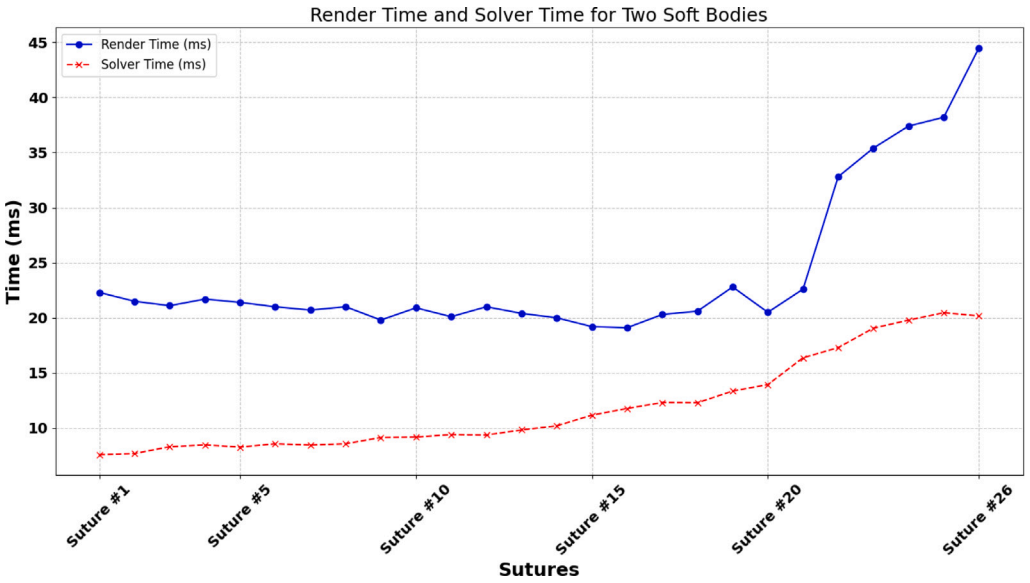

Fig. 21. Render time and simulation time measured in milliseconds of a simulation with two soft bodies sutured together.

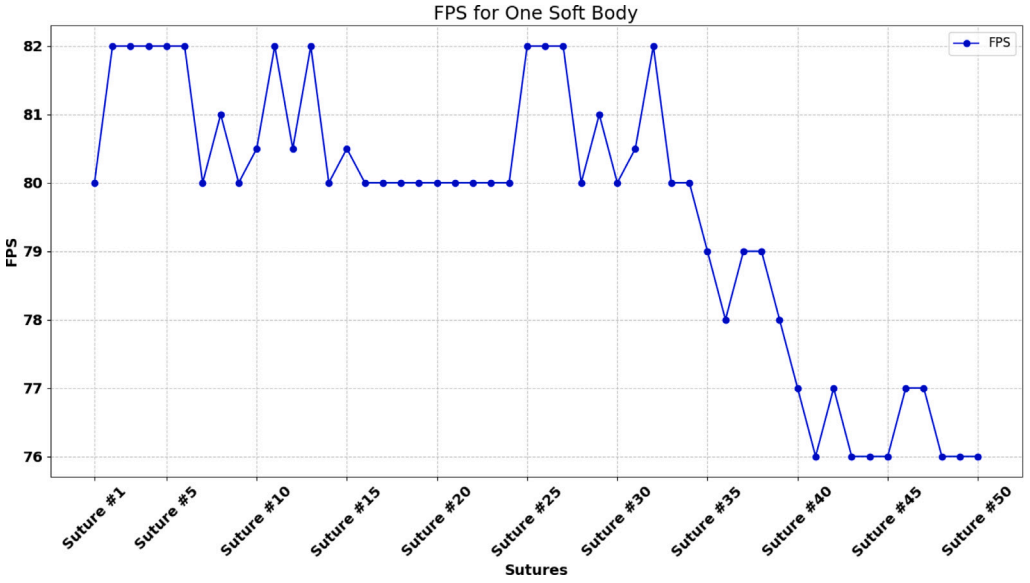

Fig. 22. Performance measured in frames per second of a simulation with one soft body being sutured.

using Allis clamps. Next, the circumferential excision was performed with electrocautery (Bovie). After the prolapsed rectum was excised, the anastomosis phase involved placing sutures around the anus, approximately 4–6, with full-thickness bites and knot tying, ensuring the proper approximation of the two tissue walls which is shown in Fig. 26.

The participants ($n = 24$) were divided into two groups: Novice (PGY 1–2) ($n = 12$) and Experts (PGY 3–5 and Faculty) ($n = 12$) groups. At the end of the task, participants were asked to fill in an exit survey designed to evaluate the effectiveness of the simulator. The survey included Likert scale (5-point system) questions related to the realism of the anatomy, the quality of the model and textures, the interface design and the force feedback and its usefulness. The survey questions and the associated box-plots can be seen in Table 5 and Fig. 27, respectively.

Table 5
Likert scale questions from the Rectal Prolapse Simulator case study.

Q#	Question
Q1	Rate the degree of realism of the anatomy (how realistic it looks)
Q2	Rate the degree of realism of the anatomy (models and textures)
Q3	Rate the degree of overall realism of the simulator interface (instrument, model, and display)
Q4	Rate the realism of the force feedback in the simulator
Q5	Rate the usefulness of the force feedback in the simulator

Approximately 71% of both the participants ($n = 17$) rated the anatomy (Q1) as being moderate (3 points) to very realistic (5 points). Furthermore, the model, textures (Q2) and the interface (Q3) realism were also rated as moderate to very realistic by 71% of the total number

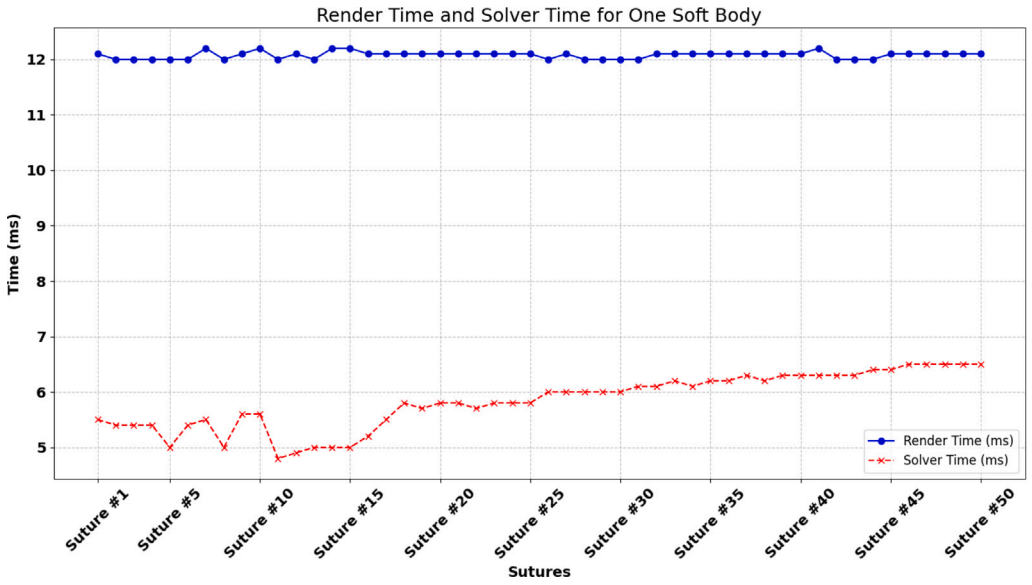

Fig. 23. Render time and simulation time measured in milliseconds of a simulation with one soft body being sutured.

Fig. 24. A surgery resident using the VCOST-RP simulator during the study at UT Southwestern.

of participants ($n = 17$). Notably, half of the participants ($n = 12$) rated the realism of the force feedback (Q4) as moderate to very realistic, while more than half ($n = 13$) rated the force feedback's usefulness (Q5) similarly, highlighting its potential value in surgical training.

4. Discussion

The user study feedback indicated that our simulator's force feedback remains an area for improvement, with only about half of participants rating the haptics as realistic or highly useful. This reception is likely influenced by simplifications in our haptic model and the inherent constraints of the hardware. In its current form, the haptic model does not incorporate frictional forces; instead, feedback is limited to normal forces generated by tissue puncture or stretch, which simplifies implementation but may reduce realism [41].

The force limitations of our device (maximum 3.3 N) also shaped our design choices. For tissue simulation, we selected a spring constant of 1100 N/m, substantially lower than the values derived from published Young's modulus data for colon tissue under typical suturing conditions approximately 30,000 N/m for the circumferential direction (E = 0.6 MPa) and 85,000 N/m for the longitudinal direction (E = 1.7 MPa) [42]. Using these physical values would result in perceptible deformations of less than 0.1 mm before reaching the device's maximum force, which is impractical for interactive training. By reducing stiffness to 1100 N/m, the simulator allows for about 3 mm of tissue deformation during suturing, enhancing both perceived realism and

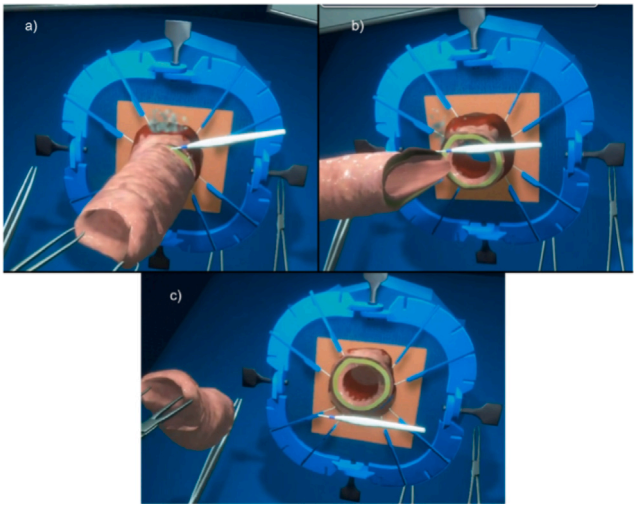

Fig. 25. (a) Cutting with the Bovie pen is started. (b) Cutting around the circumference of the bowel is nearly complete. (c) Cutting is completed and the prolapse is removed.

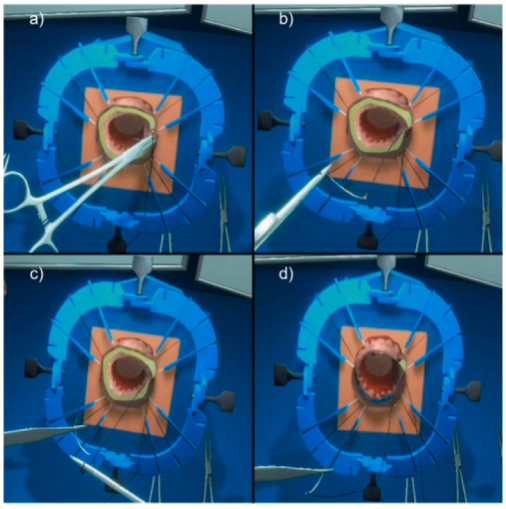

Fig. 26. (a) A full-thickness bite is performed with the needle. (b) The suture thread is pulled through. (c) The suture thread is wrapped around the needle holder to create a knot. (d) Knots are created around the circumference of the bowel.

training value. The selected stiffness value was approved by expert surgeons.

There are several technical limitations to our current approach. First, our handling of thread–tissue collisions near suture points involves disabling collisions for a few thread particles in these regions to prevent unstable reactions during pull-through. While this improves stability and performance, it can cause the thread to visibly interpenetrate the tissue, slightly reducing physical realism. Second, we were unable to compare performance against existing simulators because no open-surgery platform currently exists for direct benchmarking. Our evaluation was therefore restricted to internal performance metrics and feedback from surgeons. The evaluation metrics for this work have been purposefully kept limited in scope to performance and realism metrics. We have left evaluation of this simulator as a valid training tool for future work. Finally, the absence of modeled friction between the thread and tissue, or between the tools and thread, may result in subtle mismatches between visual and haptic cues, such as, a thread appearing taut or sliding without corresponding force feedback.

In terms of robustness, the simulator performed reliably during both internal testing and the user study, with no major failures such as thread breakage, knot slippage, or simulation instability. However, we acknowledge that our system assumes the user follows reasonable suturing procedures.

5. Conclusion

In conclusion, we have developed a real-time haptic-based soft-body suturing system within a VR environment tailored for open surgery. This system serves to both train and assess doctors' proficiency in suturing soft tissues. To represent the soft bodies and suture threads, we utilized an XPBD particle system. We employed methods to have each part of the suturing process interact with one another, whether it was the soft body organ, rope-like suture thread, or the rigid body needle. We implemented a user input system that uses two 3D Systems Touch haptic devices to control the tools. Users can perform suturing techniques, such as a Purse suture, a Connell suture, a Lambert suture, looping suture thread around a tool, tying a knot with tools, and pulling a knot taut. It was deployed with realistic graphics and textures to increase fidelity and immerse the user in the surgical environment. As a part of this work, we carried out a case study using the VCoST-RP simulator, where we utilized our suturing system. The results of this study provided valuable insights into how the VCoST-RP simulator offered opportunities for further surgical simulation refinement, enhancing the trainee's experience, allowing them to practice in a highly realistic and safe environment, ensuring better preparation for real-world procedures. As for future work, we want to further improve the performance of our suturing techniques by identifying bottlenecks and utilizing parallel processing for GPU acceleration.

CRediT authorship contribution statement

George Westergaard: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Mark Ellis:** Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Jacob Barker:** Writing – original draft, Software, Methodology, Conceptualization. **Sofia Garces Palacios:** Writing – original draft, Validation, Formal analysis, Data curation. **Alexis Desir:** Writing – original draft, Validation, Investigation. **Ganesh Sankaranarayanan:** Writing – review & editing, Writing – original draft, Validation, Supervision, Investigation, Funding acquisition, Formal analysis. **Suvranu De:** Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition. **Doga Demirel:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project was supported by grants from the National Institutes of Health (NIH)/NIBIB R01EB025241, R01EB033674, R01EB032820, and R01EB005807.

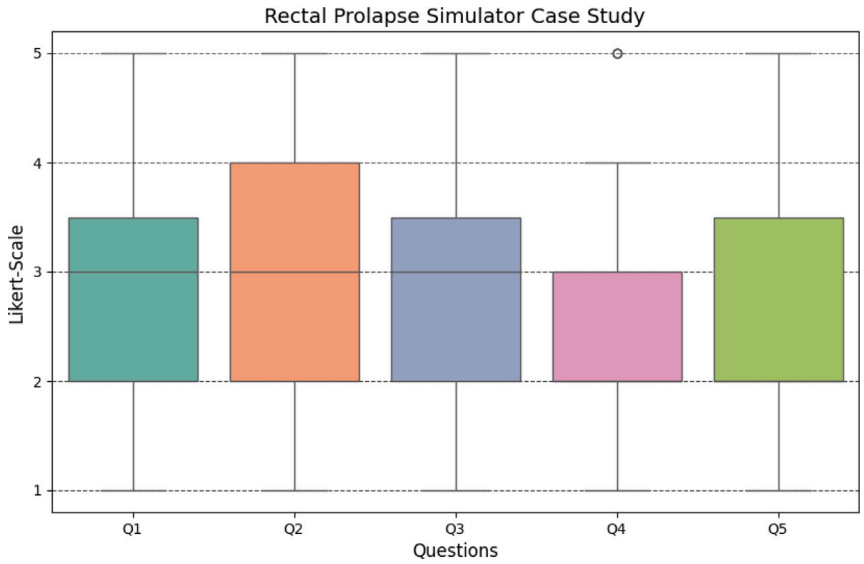

Fig. 27. Boxplots for the Likert scale questions from the Rectal Prolapse Simulator case study.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2025.104507.

Data availability

Data will be made available on request.

References

[1] Humm G, Mohan H, Fleming C, Harries R, Wood C, Dawas K, et al. The impact of virtual reality simulation training on operative performance in laparoscopic cholecystectomy: Meta-analysis of randomized clinical trials. BJS Open 2022;6:zrac086. http://dx.doi.org/10.1093/bjsopen/zrac086.

[2] Behmadi S, Asadi F, Okhovati M, Ershad Sarabi R. Virtual reality-based medical education versus lecture-based method in teaching start triage lessons in emergency medical students: Virtual reality in medical education. J Adv Med Educ Prof 2022;10:48–53.

[3] Wu Q, Wang Y, Lu L, Chen Y, Long H, Wang J. Virtual simulation in undergraduate medical education: A scoping review of recent practice. Front Med 2022;9:855403.

[4] Brown K, Swoboda S, Gilbert G, Horvath C, Sullivan N. Integrating virtual simulation into nursing education: A roadmap. Clin Simul Nurs 2022;72:21–9.

[5] Lenoir J, Meseure P, Grisoni L, Chaillou C. A suture model for surgical simulation. Med Simul 2004;105–13.

[6] Jourdes F, Valentin B, Allard J, Duriez C, Seeliger B. Visual haptic feedback for training of robotic suturing. Front Robot AI 2022;9:800232.

[7] Müller M, Heidelberger B, Hennix M, Ratcliff J. Position based dynamics. J Vis Commun Image Represent 2007;18:109–18.

[8] 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. 2016, p. 49–54.

[9] Peters P, Lemos M, Bönsch A, Ooms M, Ulbrich M, Rashad A, et al. Effect of head-mounted displays on students' acquisition of surgical suturing techniques compared to an e-learning and tutor-led course: A randomized controlled trial. Int J Surg 2023;109:2228–40.

[10] Perrenot C, Perez M, Tran N, Jehl J, Felblinger J, Bresler L, et al. The virtual reality simulator dV-Trainer® is a valid assessment tool for robotic surgical skills. Surg Endosc 2012;26:2587–93.

[11] Cho J, Hahn K, Kwak J, Kim J, Baek S, Shin J, et al. Virtual reality training improves da Vinci performance: A prospective trial. J Laparoendosc Adv Surg Tech 2013;23:992–8.

[12] Gabrysz-Forget F, Bonds M, Lovett M, Alseidi A, Ghaderi I, Nepomnayshy D. Practicing on the advanced training in laparoscopic suturing curriculum (ATLAS): Is mastery learning in residency feasible to achieve expert-level performance in laparoscopic suturing? J Surg Educ 2020;77:1138–45.

[13] Rodrigues S, Horeman T, Blomjous M, Hiemstra E, Dobbelsteen J, Jansen F. Laparoscopic suturing learning curve in an open versus closed box trainer. Surg Endosc 2016;30:315–22.

[14] Duffy A, Hogle N, McCarthy H, Lew J, Egan A, Christos P, et al. Construct validity for the LAPSIM laparoscopic surgical simulator. Surg Endosc Other Interv Tech 2005;19:401–5.

[15] Zhou M, Tse S, Derevianko A, Jones D, Schwaitzberg S, Cao C. Effect of haptic feedback in laparoscopic surgery skill acquisition. Surg Endosc 2012;26:1128–34.

[16] Hagelsteen K, Johansson R, Ekelund M, Bergenfelz A, Anderberg M. Performance and perception of haptic feedback in a laparoscopic 3D virtual reality simulator. Minim Invasive Ther Allied Technol 2019;28:309–16.

[17] McDougall E, Kolla S, Santos R, Gan J, Box G, Louie M, et al. Preliminary study of virtual reality and model simulation for learning laparoscopic suturing skills. J Urol 2009;182:1018–25.

[18] Dehabadi M, Fernando B, Berlingieri P. The use of simulation in the acquisition of laparoscopic suturing skills. Int J Surg 2014;12:258–68.

[19] Qi D, Panneerselvam K, Ahn W, Arikatla V, Enquobahrie A, De S. Virtual interactive suturing for the fundamentals of laparoscopic surgery (FLS). J Biomed Informatics 2017;75:48–62.

[20] Korzeniowski P, Płotka S, Brawura-Biskupski-Samaha R, Sitek A. A virtual reality simulator for fetoscopic spina bifida repair surgery. In: 2022 IEEE/RSJ international conference on intelligent robots and systems. IROS, 2022, p. 401–6.

[21] Sanford D, Ma R, Ghoreifi A, Haque T, Nguyen J, Hung A. Association of suturing technical skill assessment scores between virtual reality simulation and live surgery. J Endourol 2022;36:1388–94.

[22] Beyer-Berjot L, Aggarwal R. Toward technology-supported surgical training: The potential of virtual simulators in laparoscopic surgery. Scand J Surg 2013;102:221–6.

[23] Erden U, Gromski M, De S, Demirel D. Preliminary validation of the virtual bariatric endoscopic simulator. IGIE 2024;3:453–62.

[24] Erden U, Toker O, Gromski M, De S, Demirel D. Design and development of a real-time virtual bariatric endoscopic simulator with haptic feedback. ACM Trans Sens Networks 2025.

[25] Chheang V, Saalfeld P, Huber T, Huettl F, Kneist W, Preim B, et al. Collaborative virtual reality for laparoscopic liver surgery training. In: 2019 IEEE international conference on artificial intelligence and virtual reality. AIVR, 2019, p. 1–17.

[26] Pasini N, Mariani A, Munawar A, De Momi E, Kazanzides P. A virtual suturing task: Proof of concept for awareness in autonomous camera motion. In: 2022 sixth IEEE international conference on robotic computing. IRC, 2022, p. 376–82.

[27] Sanford D, Der B, Haque T, Ma R, Hakim R, Nguyen J, et al. Technical skill impacts the success of sequential robotic suturing substeps. J Endourol 2022;36:273–8.

[28] Berkley J, Turkiyyah G, Berg D, Ganter M, Weghorst S. Real-time finite element modeling for surgery simulation: An application to virtual suturing. IEEE Trans Vis Comput Graphics 2004;10:314–25.

[29] Gan W, Mok T, Chen J, She G, Zha Z, Wang H, et al. Researching the application of virtual reality in medical education: One-year follow-up of a randomized trial. BMC Med Educ 2023;23:3.

[30] Mok T, Chen J, Pan J, Ming W, He Q, Sin T, et al. Use of a virtual reality simulator for tendon repair training: Randomized controlled trial. JMIR Serious Games 2021;9:e27544.

[31] Cardona-Rivera J, Cardona-Reyes H, Alvarez-Rodriguez F, Muñoz-Arteaga J. Virtual reality environment for surgical skills practice for medical students. In: Proceedings of the XI latin American conference on human computer interaction. 2023, p. 1–6.

[32] Hu J, Chang C, Tardella N, Pratt J, English J. Effectiveness of haptic feedback in open surgery simulation and training systems. Stud Health Technol Inform 2006;119:213–8.
[33] Liu F, Su E, Lu J, Li M, Yip M. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. IEEE Robot Autom Lett 2023;8:3964–71.
[34] Shimrat M. Algorithm 112: Position of point relative to polygon. Commun ACM 1962;5:434.
[35] Parico A, Ahamed T. Real time pear fruit detection and counting using YOLOv4 models and deep SORT. Sensors 2021;21:4803.
[36] Lucas ACarmo, Heithecker S, Ruffer P, Ernst R, Ruckert H, Wischermann G, et al. A reconfigurable HW/SW platform for computation intensive high-resolution real-time digital film applications. In: Proceedings of the design automation & test in europe conference, vol. 1, 2006, p. 1–6.
[37] Cirocco W. The altemeier procedure for rectal prolapse: An operation for all ages. Dis the Colon Rectum 2010;53:1618–23.
[38] Desir A, Pourghaderi P, Hegde S, Demirel D, Pogacnik J, De S, et al. Validity of task-specific metrics for assessment in perineal proctectomy. Surg Endosc 2024;38:5319–30.
[39] Sankaranarayanan G, Parker L, Jacinto K, Demirel D, Halic T, De S, et al. Development and validation of task-specific metrics for the assessment of linear stapler-based small bowel anastomosis. J Am Coll Surg 2022;235:881–93.
[40] Sankaranarayanan G, Parker L, Khan A, Dials J, Demirel D, Halic T, et al. Objective metrics for hand-sewn bowel anastomoses can differentiate novice from expert surgeons. Surg Endosc 2023;37:1282–92.
[41] Duriez C, Dubois F, Kheddar A, Andriot C. Realistic haptic rendering of interacting deformable objects in virtual environments. IEEE Trans Vis Comput Graphics 2006;12:36–47.
[42] Massalou D, Masson C, Afquir S, Baqué P, Arnoux P, Bège T. Mechanical effects of load speed on the human colon. J Biomech 2019;91:102–8.