Design and Development of a Real-Time Virtual Bariatric Endoscopic Simulator with Haptic Feedback

UTKU ERDEN, Florida Polytechnic University, Lakeland, United States
ONUR TOKER, Florida Polytechnic University, Lakeland, United States
MARK A. GROMSKI, Indiana University School of Medicine, Indianapolis, United States
SUVRANU DE, FAMU-FSU College of Engineering, Tallahassee, United States
DOGA DEMIREL*, School of Computer Science, University of Oklahoma, Norman, United States

Virtual Bariatric Endoscopy (ViBE) simulator is designed for Endoscopic Sleeve Gastroplasty (ESG), a minimally invasive bariatric procedure. While virtual simulators exist for bariatric surgeries, there has been a lack of ESG-specific tools. The ViBE simulator fills this gap by providing a cost-effective alternative to physical models and enhancing ESG training. The simulator consists of three main components: software simulation, hardware, and a hardware interface linking the two. The software focuses on ESG techniques like marking, suturing, and tissue-pulling, with an algorithm for suturing and soft-body physics simulations. The simulator features two human-computer interfaces: one using USB-HID protocol with optical encoders and ARM Cortex M7 devices, and another using computer vision for delicate instruments. The computer vision interface simplifies mechanical design. Performance tests showed an average of 55 FPS, with render times between 2ms and 4ms and solver times between 16ms and 18ms. The end-to-end delay was under 75ms, and haptic feedback forces updated every 1ms. The ViBE simulator aims to improve ESG training and suturing techniques, demonstrating its potential as an effective learning tool with efficient software performance and minimal hardware latency.

CCS Concepts: • **Human-centered computing** → **HCI design and evaluation methods**; • **Computing methodologies** → **Modeling and simulation**; • **Hardware** → **Emerging technologies**.

Additional Key Words and Phrases: Surgical simulation, Endoscopy, Gastroplasty

1 Introduction

The widespread prevalence of global obesity [9, 31] poses a significant challenge, often rendering a majority of suitable candidates prefer bariatric surgery. Although this surgical approach remains the most effective method for weight loss in severe obesity cases [20], limitations concerning accessibility, cost, and associated risks have spurred the development of innovative alternatives [2, 26]. Endoscopic Sleeve Gastroplasty (ESG) is a minimally invasive procedure that utilizes endoscopic suturing to reduce stomach size and promote weight loss [5]. Multiple case studies have demonstrated this method's safety and technical feasibility, significantly reducing average weight and body mass index [35]. Nevertheless, mastering this technique presents inherent challenges, necessitating specialized skills, extensive training, and a gradual learning process [16]. In response

*Corresponding Author

Authors' Contact Information: Utku Erden, Florida Polytechnic University, Lakeland, Florida, United States; e-mail: uerden4601@floridapoly.edu; Onur Toker, Florida Polytechnic University, Lakeland, Florida, United States; e-mail: otoker@floridapoly.edu; Mark A. Gromski, Indiana University School of Medicine, Indianapolis, Indiana, United States; e-mail: mgromski@iu.edu; Suvranu De, FAMU-FSU College of Engineering, Tallahassee, Florida, United States; e-mail: sde@eng.famu.fsu.edu; Doga Demirel, School of Computer Science, University of Oklahoma, Norman, Oklahoma, United States; e-mail: doga@ou.edu.

ACM 1550-4867/2025/7-ART
https://doi.org/10.1145/3747184

to the essential need for comprehensive training, virtual simulations have become a well-established tool for enhancing medical education and practice [21]. This paper presents the Virtual Bariatric Endoscopy (ViBE) simulator [11, 12] for the ESG procedure. The ViBE simulator's software underwent preliminary validation in [11], where its performance and simulation metrics were assessed through the correct differentiation between expert and novice participants. However, this work presents for the first time the complete ViBE system. This includes a comprehensive description of the simulator's architecture, integration of custom-designed haptic hardware, hardware-software interfacing, and real-time control strategies. Collectively, these contributions allow ViBE to function as a fully integrated, immersive, and high-performance simulator for ESG. The system encompasses all required software tasks, such as argon plasma coagulation (APC) marking, suturing, and tissue pulling, all integrated within virtual reality (VR). Additionally, it incorporates hardware components to enhance the realism of the surgical experience, along with a hardware interface facilitating data transfer between hardware and software.

Numerous models and simulators have been created for minimally invasive bariatric surgery training. Laparoscopic box trainers like the EndoSuture Trainer Box Simulator and Laparoscopic VR (Lapsim) are proficient in teaching fundamental skills [34]. The study in [40] has investigated the effectiveness of virtual reality simulators such as MIST-VR, Simsurgery, LapMentor, and Sinergia for basic and advanced laparoscopic and minimally invasive surgical skills. Nevertheless, there is insufficient data regarding applying acquired skills to real patients and the influence of virtual reality simulation training on clinical results. While virtual reality simulators provide objective metrics for laparoscopic task performance, they lack haptic feedback. While affordable box trainers offer realistic haptic feedback, they lack objective performance assessment. Hybrid simulators provide metrics but lack haptic feedback, and augmented reality simulators offer both at a higher cost. Laboratory animals and cadaver models provide realistic training experiences but are expensive and differ anatomically from humans [40]. The EndoSuture Trainer Box Simulator (ESTBS) is an economical laparoscopic skills simulator featuring a fiberglass console, hollow ring, and suture tray matrix [6]. The Lapsim surgical simulator assesses specific technical skills in both basic and advanced laparoscopic tasks within a virtual environment [10].

There is a growing integration of VR technology in endoscopic training as it was shown that using VR simulators for learning endoscopic skills, resulted in better outcomes [41]. Siau et al. [36] evaluated the effectiveness of a simulation-based induction program for gastroscopy training. Their study found that trainees who underwent structured simulation training exhibited significantly improved procedural competency and confidence compared to those following traditional learning pathways. The program accelerated learning curves by offering hands-on practice in a standardized manner before real patient exposure. Most importantly, the findings indicated that early-stage simulation training reduced procedural errors and improved overall knowledge retention. In [33] a protocol for a randomized trial to evaluate the impact of gamification in simulation-based endoscopy training was developed. In this study, EndoVR [37], a VR-based endoscopy simulator with haptic feedback was used for training. They have used gamification to apply game-design elements to educational settings with a goal of enhancing engagement and motivation. The study randomized 36 novice endoscopists into two groups: a conventional simulation training group and a gamified curriculum group. Both groups received six hours of simulation training with four hours of small-group didactic sessions. However, the gamified curriculum included elements such as real-time performance tracking, progressive challenges, and rewards for high performance. Participants' skills were assessed pre-training, post-training, and at a follow-up period of 4–6 weeks, with evaluations based on simulated and real-world clinical colonoscopies.

In [1] VR-based clinical skill training in neurosurgery, including endoscopic procedures was examined. The results showed that VR-based simulations helped neurosurgical trainees become proficient in handling endoscopes before entering the operating room and that standardized VR training protocols could improve patient outcomes by minimizing novice errors. In a different study, Karnam et al. [18] introduced a VR-based inverse kinematics tool designed to enhance endoscopic robotic surgery. This system allowed surgeons to teach robotic systems

specific endoscope maneuvers, significantly reducing operation time and physical strain, while demonstrating that their VR-assisted robotic system improved hand-eye coordination and shortened learning curves for endoscopic procedures.

VR laparoscopic simulators (VRLS) and augmented reality laparoscopic simulators (ProMIS) have been created to address the specific skill requirements, time constraints, and legal considerations associated with laparoscopic training. VRLS function as effective and secure tools for laparoscopic surgery, with studies underscoring the significance of haptic feedback for enhanced skill transfer in laparoscopic training [13]. Augmented reality (AR) simulators, such as ProMIS, offer realistic haptic feedback and objective performance assessment, amalgamating the benefits of box trainers and VRLS [3]. As emphasized by Botden et al. [4], AR simulators provide essential realistic haptic feedback for effective laparoscopic suturing training. They show potential in training tasks related to bariatric and colon surgery, highlighting their significance in procedural training [32]. Conversely, VR remains appropriate for basic skills, establishing itself as a credible training method. Studies utilizing Laparoscopy VR software indicated that haptic-enhanced simulation significantly improves laparoscopic skill performance, demonstrating substantial enhancements in drills compared to non-haptic simulation [32]. Our proposed simulator, ViBE, offers a cost-effective alternative to physical stomach models. Despite the abundance of virtual simulators for various bariatric surgeries, there is a noticeable gap in simulators tailored for ESG. Considering the benefits of ESG over traditional bariatric procedures, there is a compelling need for broader adoption. The introduction of the ViBE simulator represents a significant leap forward in ESG training methodology [7, 14]. To the best of our knowledge, the ViBE simulator detailed in this study is the only VR-based ESG training platform. This study aims to design and develop the software and hardware components of the ViBE simulator, utilizing advanced hardware systems and virtual reality to ensure its effectiveness.

2 Design & Implementation

The simulator has three main components: a) software, b) hardware, and c) hardware interface. Software tasks include Argon Plasma Coagulation (APC) marking, suturing, and tissue pulling. Suturing involves three technical aspects: APC marking is a primary method used to determine suturing locations in the stomach. Typically, two or three different stomach areas are marked to define the working zone. In the simulator, we are rendering suture lines to initiate the first step of suturing, elongating the clamped part of the stomach using soft body calculations, and generating random bleedings to enhance the realism of the surgical experience for surgeons. Tissue pulling involves three steps: detecting the suture technique to create realistic tension between the bite points, dropping the T-Tag to finalize the suture set and prepare the sutures for pulling towards the center, and finally dropping the cinch after completing the tissue pulling to finalize the suture set. The suturing and tissue-pulling process is repeated until the surgeon is satisfied with the procedure. Through the integration of VR, we provide a more lifelike and immersive surgical environment. In the hardware, the endoscope is linked to two separate edge devices. One provides haptic feedback and position sensing, while the other enables button controls and position sensing. Both devices are connected via USB-HID to ensure rapid communication and ease of software development.

Furthermore, a computer vision-based technique has been employed as another human-computer interface solution between the hardware and software. More specifically, a camera detects specific components of the Overstitch (such as the helix and trigger), and a computer vision algorithm tracks the movement of these components. The location data is then transmitted to the simulation via User Datagram Protocol (UDP). The overall architecture of the ViBE simulator can be seen in Figure 1 and the complete simulation can be seen in the supplemental video.

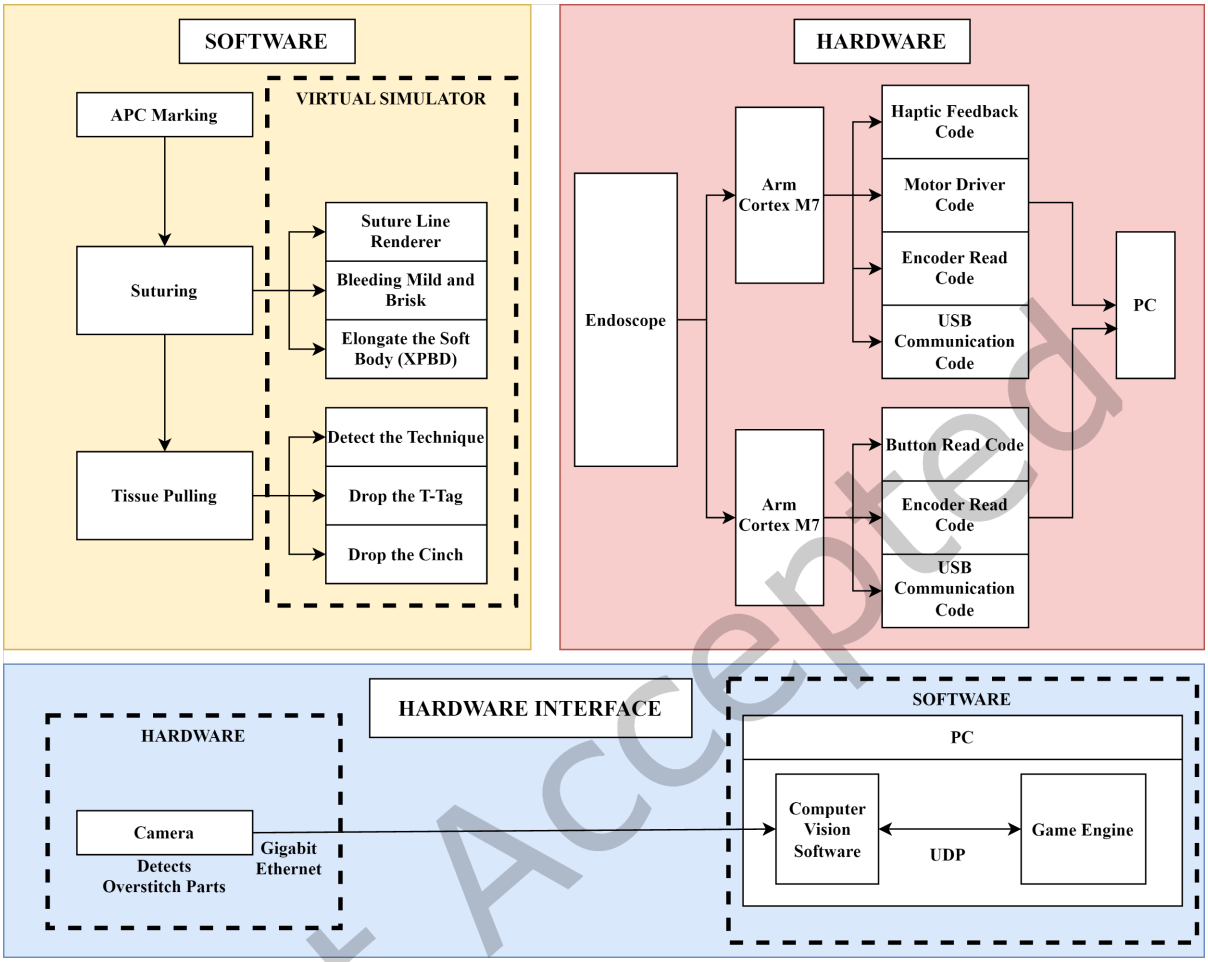

Fig. 1. System Diagram of the Framework.

2.1 Software

2.1.1 APC Marking. Confirming the specific stomach areas is essential before starting the cinching and bite-taking procedures. This involves understanding the distinct anatomical regions of the stomach as depicted in Figure 2a, namely, the fundus, pyloric canal, esophagus, duodenum, and body—holds importance prior to undergoing gastroenterology. As depicted in Figure 2b, knowing the location of the Posterior, Anterior, and Greater Curvature is vital for the suturing and APC marking process. Surgeons need to mark these specific zones to confine the suturing area accurately. This practice enables surgeons to identify the precise section of the stomach suitable for suturing, ensuring accuracy in the procedure.

Another crucial consideration during suturing and marking involves the soft-body behavior of the stomach. Physicians are expected to anticipate the dynamic movement of the stomach before or after marking and suturing to ensure accuracy. To emulate complex soft-body physics, especially in simulating the stomach, we employed Extended Position-Based Dynamics (XPBD) [24]. This method enabled us to separate the simulation frequency

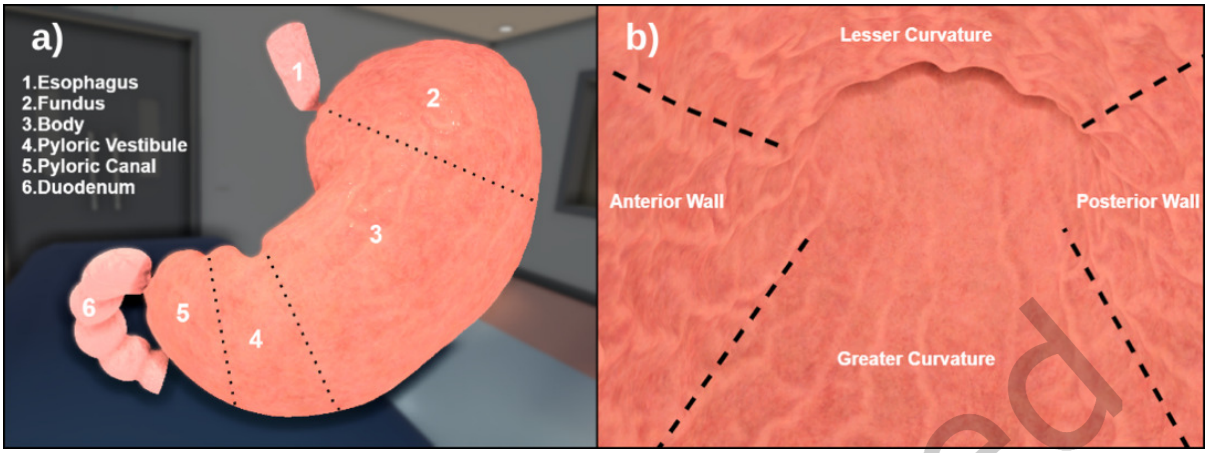

Fig. 2. a) Anatomical Division of the Stomach in Medical Context and b) Anatomical Division of the Stomach Body in Medical Context.

from object stiffness, allowing the use of a larger number of particles [25], specifically, a total of 3,618 to precisely mold the shape of the stomach. These particles were finely adjusted for soft-body deformation, shaping the mesh through linear skinning techniques [19].

For the detailed simulation of the stomach as a soft body, we integrated shape-matching constraints with oriented particles [27–29]. This integration is illustrated in Figure 3, demonstrating the alignment between the particles regulated by shape-matching constraints and the stomach's actual form. This approach ensured realistic deformations while adhering to strict shape regulations.

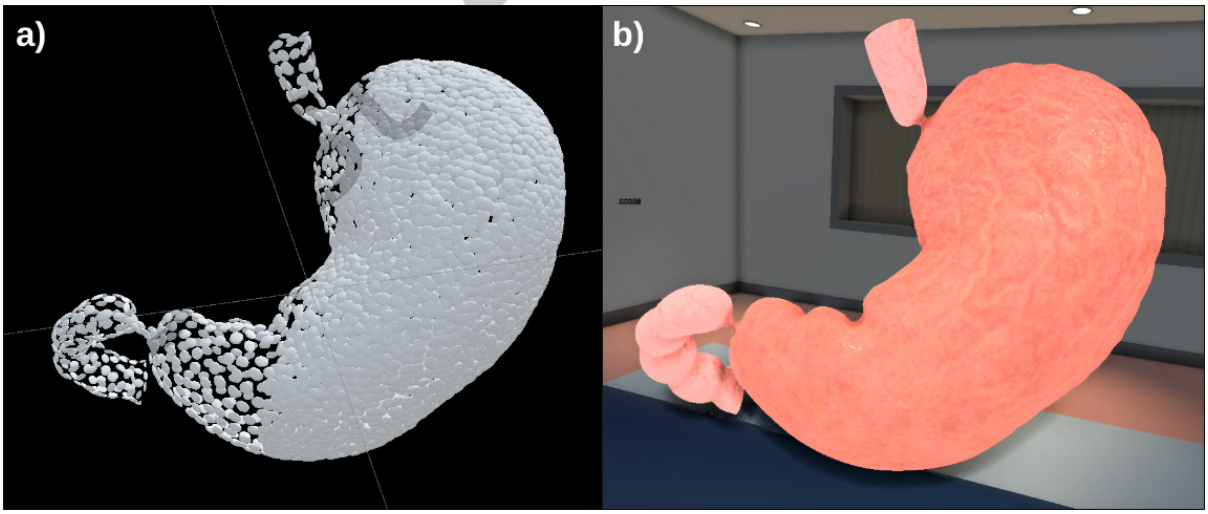

Fig. 3. a) Generated particles with shape-matching constraints and b) Rendering of the surface soft body in the virtual scene.

The preferred tool for navigating and marking the designated stomach zones is Overstitch. Overstitch is specifically tailored to affix to an endoscope. The Overstitch tool should effectively deliver Argon, outlining both the posterior-anterior walls and the greater curvature, as shown in Figure 4a. During the marking process, the stomach exhibits characteristics akin to a soft-body object, leading to elastic movements. It is essential to underscore that this soft-body behavior diverges from XPBD implementation, opting for a gradient mesh deformation [23, 39]. Prolonged marking enforcement on the same tissue may result in burning scars, as observed in Figure 4b. Stability and precision are paramount at this stage to minimize errors for the subsequent stages of the process.

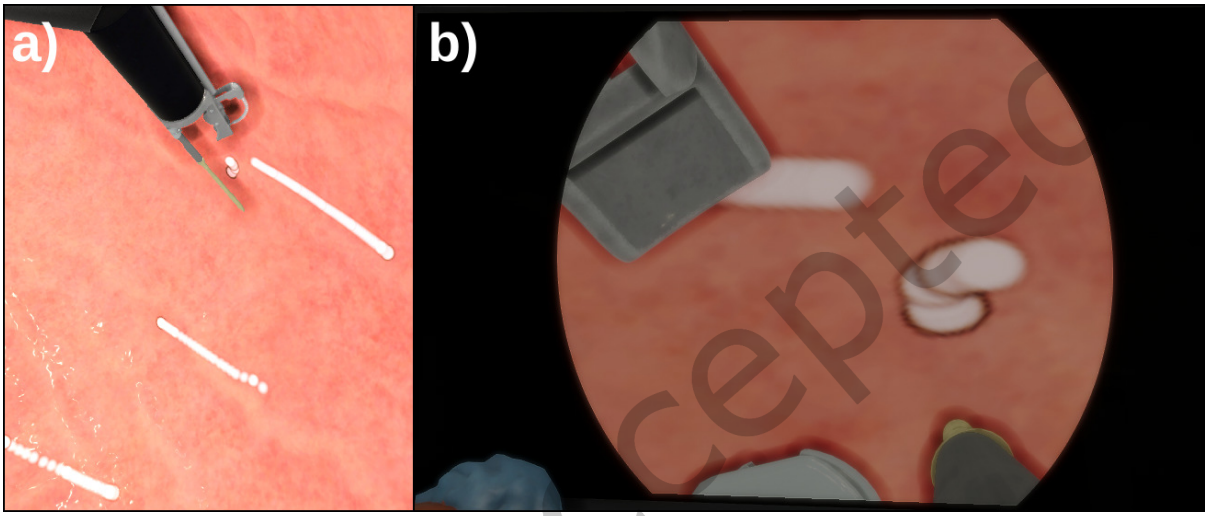

Fig. 4. a) A successful implementation of the APC marking to walls Anterior, Posterior, and Greater Curvature and b) An example of burning occurs when the Overstitch machine is held in the same spot for an extended duration.

2.1.2 Suturing. Following a successful APC Marking step, identifying the Anterior, Posterior, and Greater Curvature locations, the helix can be advanced to make contact with the surface of the stomach. Once the targeted particle is identified, the helix can be pulled back. Stretching the tissue enough with the helix indicates the correct tissue positioning for engagement. Another significant aspect to consider is the soft body's elasticity, noticeable as the displacement of the targeted particle increases compared to other particles during the helix's pulling process, as depicted in Figure 5. This situation results in a tighter elasticity of the soft body.

Upon taking the bite, the needle pierces the tissue and threads the sutures, as illustrated in Figure 6. Occasionally, this action may result in minor bleeding, often overlooked by physicians if it remains mild. However, the procedure necessitates immediate completion in the event of brisk bleeding. This sequential process must be repeated until the designated number of bites is achieved.

During suture creation, a mesh is constructed. This method permits connecting sequential points to create an unbroken line. It grants control over line characteristics like color, width, material, and segment count. Its core function involves rendering lines or shapes in real-time. The rendering of lines in a three-dimensional (3D) environment involves fundamental mathematical principles to define and display continuous paths. In this context, points within the 3D space are represented by their coordinates (x, y, z), employing vector-based representations. A sequence of consecutive points is interconnected to create a line, forming line segments that collectively represent the line's trajectory. The formulation of lines in 3D space often employs parametric

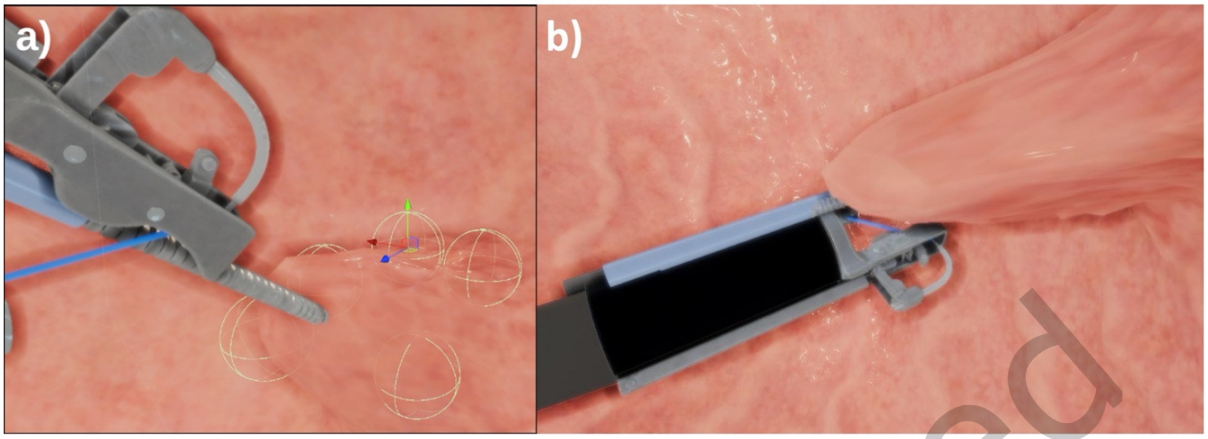

Fig. 5. a) Interaction between Particles' Collider and Helix, and b) Soft body Elongation due to Helix Traction.

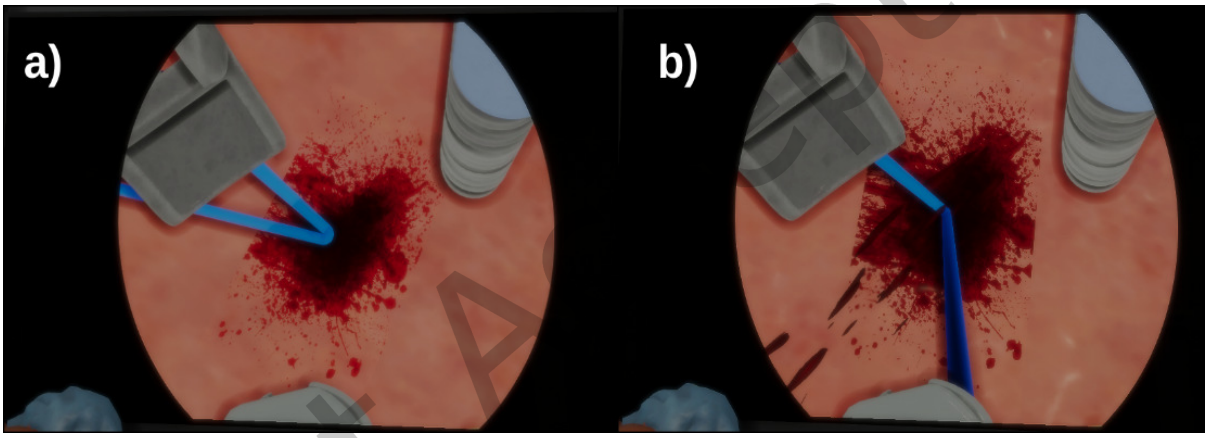

Fig. 6. a) Mild bleeding and b) Brisk bleeding.

equations to describe the line's path. One such equation, for instance, is $P(t) = P0 + t * (P1 - P0)$, where $P(t)$ signifies a point along the line as parameter t varies from 0 to 1, with $P0$ and $P1$ representing the initial and final points respectively. This parametric equation allows for the generation of discrete points along the line. Rendering these lines involves intricate processes within the graphical rendering engine. Matrices and transformations, such as perspective projection and view transformations, are employed to project these 3D points onto a two-dimensional (2D) screen space. This transformation process enables the accurate visualization of the line on the screen, translating the 3D spatial representation into a 2D display.

2.1.3 Tissue Pulling. A critical aspect of this application involves capturing the employed suturing technique. Techniques U and Z represent the most prevalent closure methods utilized for stomach procedures [15, 30]. Therefore, specific tissue-pulling solutions are developed for them.

After completing the suture set, all sutured points require pulling to apply shrinkage in the stomach. To achieve this effect, calculating the suture technique beforehand is essential. The techniques and the resulting tissue pull

are depicted in Figure 7. Upon determining the technique, our algorithm systematically records sign change indices discerned during the dot product, denoted as $\text{transition}_{\text{indices}}$. The U and Z techniques exhibit two distinct transition indices, as depicted in Figure 8.

$$d_{i,i+1} \leftarrow \{\hat{s}_i \cdot \hat{s}_{i+1} \mid i \in \{1, 2, 3, \ldots, n-1\}\} \tag{1}$$

$$\text{if } d_{i,i+1} \leq -0.5, \text{ then } \begin{cases} z_{\text{transition}_{\text{indices}}} \leftarrow z_{\text{transition}_{\text{indices}}} \cup \{d_{i,i+1}\} \\ u_{\text{transition}_{\text{indices}}} \leftarrow u_{\text{transition}_{\text{indices}}} \cup \{d_{i,i+1}\} \end{cases} \tag{2}$$

$$d_{1,n} \leftarrow \{\hat{s}_1 \cdot \hat{s}_n\} \tag{3}$$

$$\text{ST} = \begin{cases} \text{ST.Z} & \text{if } \left|z_{\text{transition}_{\text{indices}}}\right| = 2 \wedge d_{1,n} > 0.7 \\ \text{ST.U} & \text{if } \left|u_{\text{transition}_{\text{indices}}}\right| = 2 \wedge d_{1,n} < -0.7 \\ \text{ST.Other} & \text{otherwise} \end{cases} \tag{4}$$

$$\text{if Ray}(\text{Point}_1, \text{Point}_2) \cap p_i = \emptyset \text{ then } p_{\text{indices}} \leftarrow p_{\text{indices}} \cup \{i\} \tag{5}$$

To determine the technique used, we utilize the current positions of all sutures as parameters. Our algorithm performs its computations using equations (1), (2), (3) and (4). Initially, in equation (1), we compute the dot product, $d_{i,i+1}$, between the respective normalized suture vectors, $\hat{s}_i$ and $\hat{s}_{i+1}$. $\hat{s}_i = \frac{\vec{s}_i}{\|\vec{s}_i\|}$, where $\frac{\vec{s}_i}{\|\vec{s}_i\|}$ represents the normalized suture vector, $\vec{s}_i$ is the vector representation of the suture at index i, and $\|\vec{s}_i\|$ is the magnitude of the suture vector $\vec{s}_i$. If the absolute value of the dot product is less than 0.5, it indicates that the values are potentially part of both techniques U and Z. In equation (2), $d_{i,i+1}$ represents each dot product from equation (1). Following this identification, the algorithm calculates the dot value between the equation's first $\hat{s}_1$ and last normalized suture vectors $\hat{s}_n$ in equation (3). Equation (4) checks if these vectors exhibit nearly similar orientations ($d_{1,n} > 0.7$) and there exist two transition indices, the suturing technique (ST) is identified as Z. Conversely, if the vectors display almost opposite orientations ($d_{1,n} < -0.7$) and there exist two transition indices, the ST is recognized as U. Otherwise, it indicates a different ST.

After determining the technique, the tissues can be pulled towards the middle point. Hence, the equation (5) serves this purpose. p_i represents the ith particle and $\text{Ray}(\text{Point}_1, \text{Point}_2) \cap p_i = \emptyset$ checks if the ray intersects with p_i. Operating primarily with two parameters, Point_1 and Point_2, our algorithm uses this equation to establish a ray between the specified points and retrieve particle indices within that range. Within the Z technique, equation (5) operates twice to facilitate the task. Initially, it designates Point_1 as the first bite location and Point_2 as the second index within the transition indices. This action forms index group 1, consisting of particles located between Point_1 and Point_2. Upon its subsequent execution, the algorithm employs Point_1 as the first index of the transition indices and Point_2 as the final bite location. This results in the formation of index group 2, comprising particles situated between Point_1 and Point_2.

On the contrary, in the U technique, equation (5) utilizes the first bite as Point_1 and the second as Point_2 within its parameters. This action represents index group 1. Meanwhile, index group 2 includes the first transition index as Point_1 and the second as Point_2.

Following the determination of all particle indices, these two distinct index groups initiate a pulling process towards the 'middle' point, computed as the average location derived from all identified bite positions. Index group 1 and index group 2 are drawn toward the 'middle' point until they approach the elasticity limit of the soft body, as visualized in Figure 8.

In cases where neither the U nor Z techniques apply, all bite segments are uniformly drawn toward a central 'midpoint.' However, relying solely on this approach results in an outcome that doesn't meet the intended objective. Targeting specific elements alone isn't comprehensive enough. Adjustments across all relevant elements in the

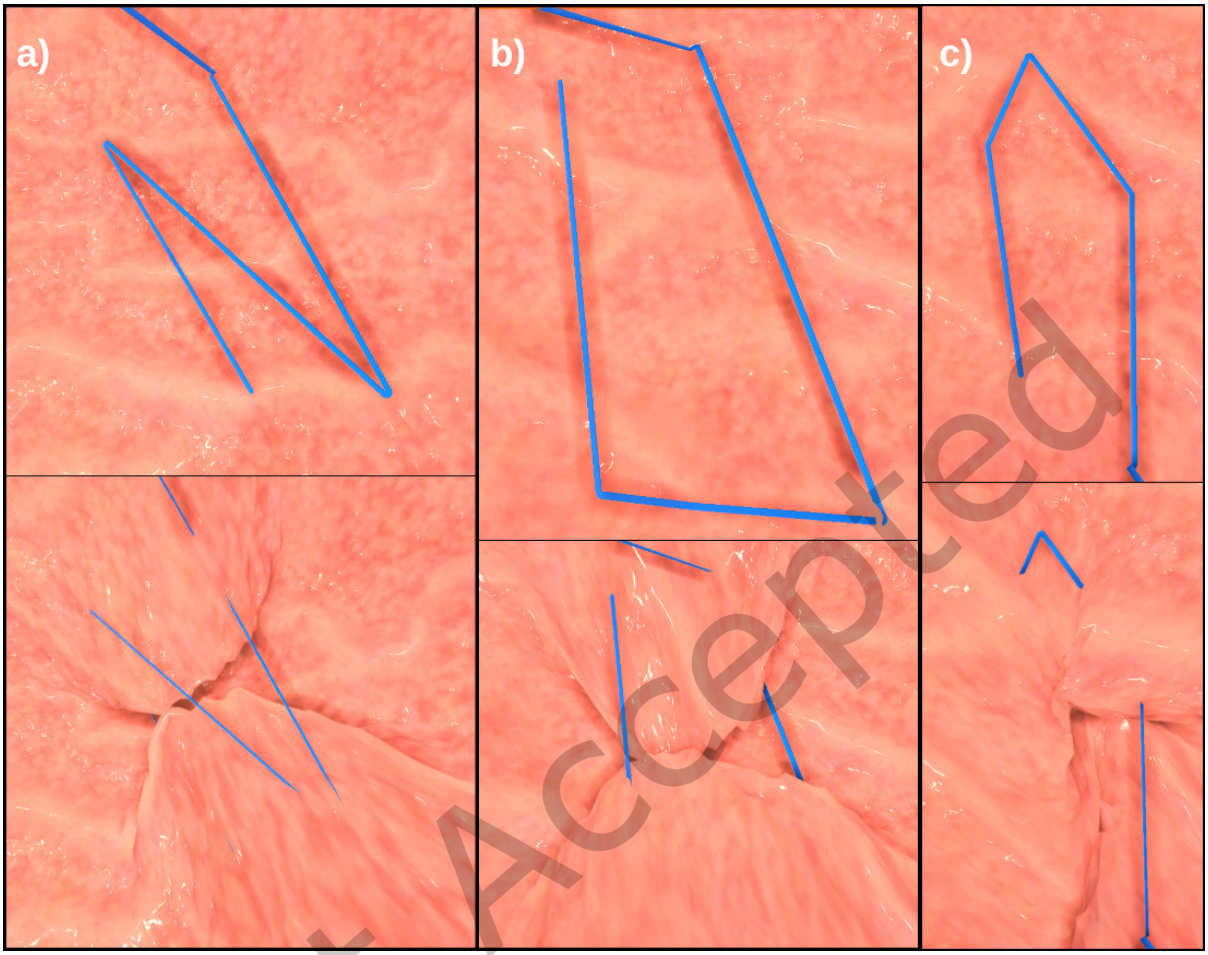

Fig. 7. a) Above: Technique Z Application | Below: Tissue Convergence towards the Midpoint, b) Above: Technique U Application | Below: Tissue Convergence towards the Midpoint, c) Above: Technique Triangular Application | Below: Tissue Convergence towards the Midpoint.

affected domains are essential to achieve a cohesive effect. Hence, determining all particle positions effectively requires the use of equation (5).

This approach begins by setting Point_1 as the first bite and Point_2 as the second, forming the index group 1. Following this, our algorithm calls equation (5) again, with Point_1 being the second bite and Point_2 being the third bite, forming the subsequent index group 2. This sequence repeats until reaching the final bite. To ensure a more structured pulling process that does not resemble a circular motion, only the initial five particles are included when creating these index groups, although this parameter is adjustable and may vary (defaulting to 5 particles). Following this stage, all particle index groups converge towards the middle point, akin to the approach observed in the other two methods and the results can be seen in Figure 7c.

Due to our understanding of the conclusive outcomes produced by techniques U and Z, we deliberately segregated their solutions from those pertaining to other techniques. For the alternative techniques (excluding U

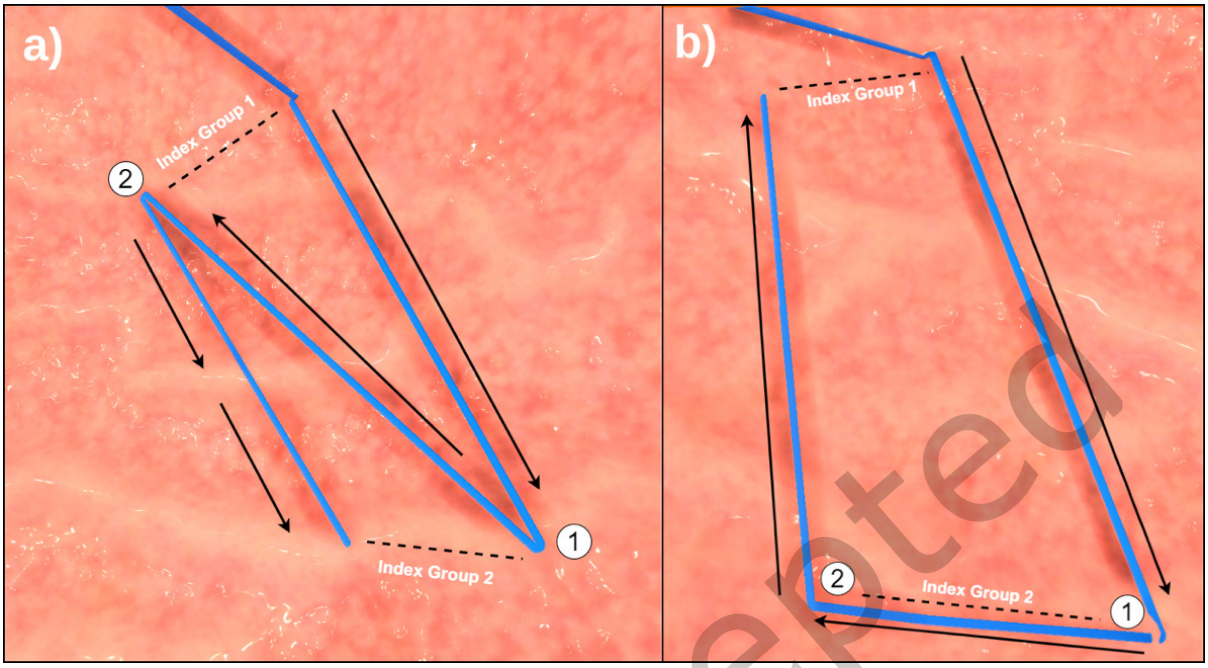

Fig. 8. a) Vector Directions, Transition Indices, and Index Group in Technique Z, b) Vector Directions, Transition Indices, and Index Group in Technique U.

and Z), a fundamental necessity exists to ensure that the generated outcome aligns with realism and complies with established physical principles. This imperative need drove the creation of an algorithm tailored explicitly for this objective.

At the end of the pulling process, the first T-Tag is deployed to establish a stabilizing environment for tissue pulling. Once the pulling is completed, the cinch, as shown in Figure 9, is released, thus completing the suturing set. The entire procedure can be repeated as needed until the physician is content with the outcome.

2.2 Hardware and Hardware Interface

The hardware interface of the surgical simulator system is crucial as it allows surgeons to interact with the simulation. The interface impacts the realism, effectiveness, and usability of the simulator. In our simulator, the hardware has three separate components:

a. An ARM Cortex-M7 haptic system connected to two encoder/motor pairs. This bare metal system running a USB stack communicates with the host PC using the USB HID protocol.
b. Another ARM Cortex-M7 input device is connected to two encoders and a set of buttons. This is also a bare metal system running a USB stack and communicates with the host PC over the USB HID protocol.
c. Computer vision software running on the host PC. This is the third input device and communicates with the simulation running on the host PC via UDP-based interprocess communication.

The input device described in part (a) is shown in Figure 10. The host PC sees this input device as a USB HID device with two 16-bit axes. Each encoder is monitored by a dedicated peripheral on the Cortex-M7, and specific peripheral registers are continuously updated automatically independent of the Cortex-M7 software. There is no

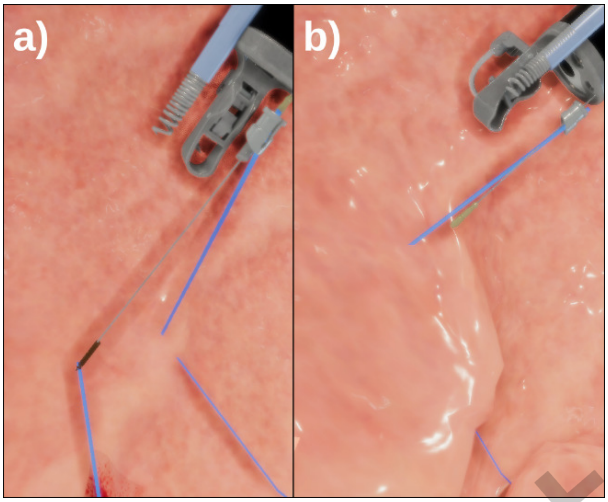

Fig. 9. Figure 9: a) Release of the T-Tag and b) Release of the Cinch.

need to monitor voltage changes at these interface pins by using the polling technique. There is also no need to use voltage-level change interrupts. This specific HW/SW interface greatly simplifies the software development on the Cortex-M7. This bare metal system has a haptic feedback loop running at 1000 Hz and utilizes a timer interrupt service. The haptic feedback code implements a simple spring-damper sensation, and all parameters are configurable but currently have fixed values. The haptic feedback calculations guarantee a completely passive system and oscillation-free behavior. Special coreless DC motors control both haptic feedback sensations. These DC motors have lower torque values than iron core alternatives, but they are much more responsive because of very low inertia. Between the motors and the Cortex-M7, we have a dual H-bridge DC motor driver driven by two pairs of PWM outputs. Each PWM signal is generated by hardware, and duty cycles are updated inside the control loop.

The input device described in part (b) is shown in Figure 11. The host PC also sees this input device as a USB HID device with two 16-bit axes and a set of buttons. This device has no haptic functionality. The Cortex-M7 is a 3.3V device, but the optical encoders used in this work are 5V. We have a passive and bi-directional level shifter for each encoder used in both input devices. These input devices report encoder and button positions (if any) every 10 ms or at a 100 Hz update rate. The update rate is configurable, but the effects of shorter/longer update periods have not been studied. Figure 12 shows an orthogonal image of the two subsystems (a and b).

2.2.1 Computer Vision Subsystem. The third component described in part (c) is shown in Figure 13. This input device has a Gigabit Ethernet Mako G-319 global shutter camera running at 147 FPS. The camera hardware supports the region of interest functionality and is interfaced with the host PC over PCI-Express. Instead of mechanically coupling an encoder to a surgical instrument, we use a single camera to monitor the position of two surgical instruments. A computer vision code is used to track the position of these surgical instruments visually. The code is designed to be computationally lightweight to minimize the latency of the input device. The game engine and the computer vision software communicate with each other over a local UDP connection. In more advanced computationally demanding setups, it is possible to run the computer vision software on a physically separate PC and use a dedicated Gigabit Ethernet cable for UDP communication between two computers. This

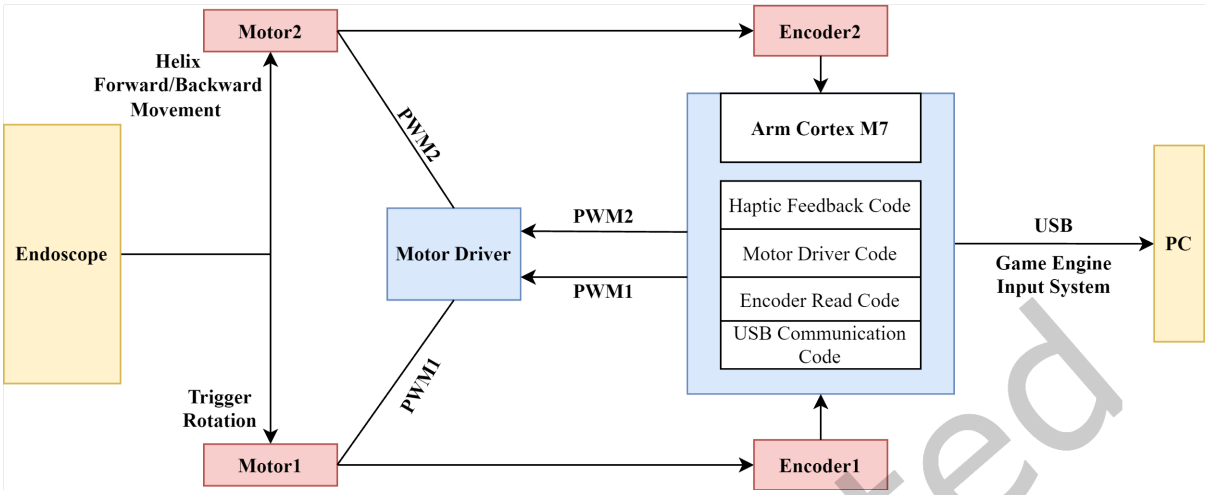

Fig. 10. ARM Cortex-M7 haptic system connected to two encoder/motor pairs.

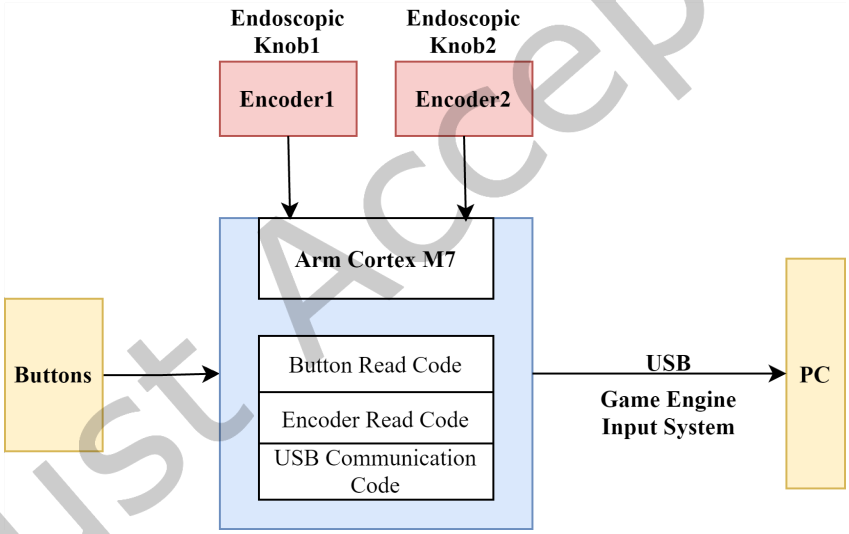

Fig. 11. ARM Cortex-M7 input device connected to two encoders and a set of buttons.

may increase the communication delay, but because of reducing the load on the host PC, it will reduce the computational delays. Figure 14 shows the computer vision subsystem running in the simulator.

3 Results

3.1 Performance Results

The tissue-pulling technique represents a key bottleneck in the simulator. To assess its impact, we evaluated whether tissue pulling adversely affected overall simulation performance. Performance was measured in frames

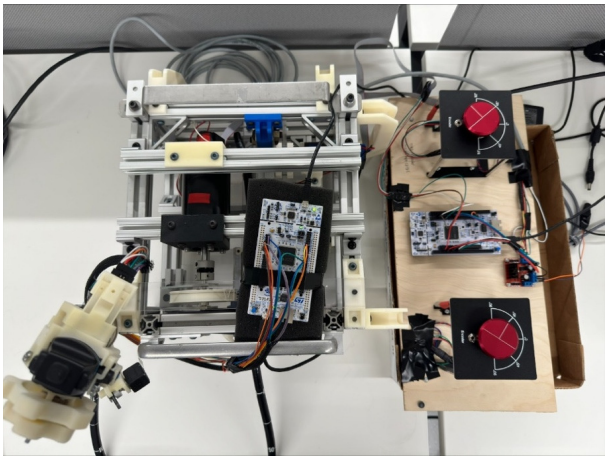

Fig. 12. Image of the two ARM Cortex-M7 systems taken from above.

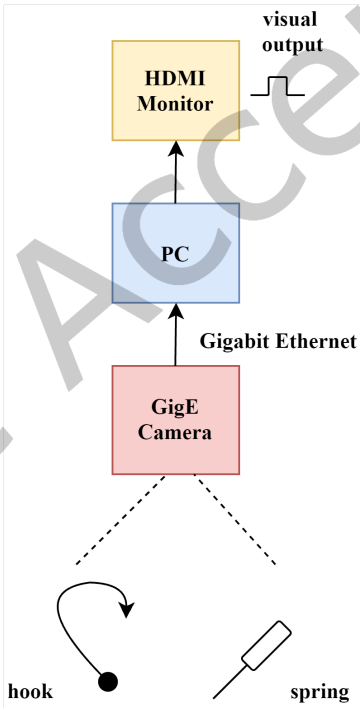

Fig. 13. Computer vision subsystem tracks the position of two surgical instruments.

per second (FPS), render time in milliseconds, and solver computation time in milliseconds, all of which play a critical role in determining real-time simulation efficiency. In addition to these performance metrics, fidelity is

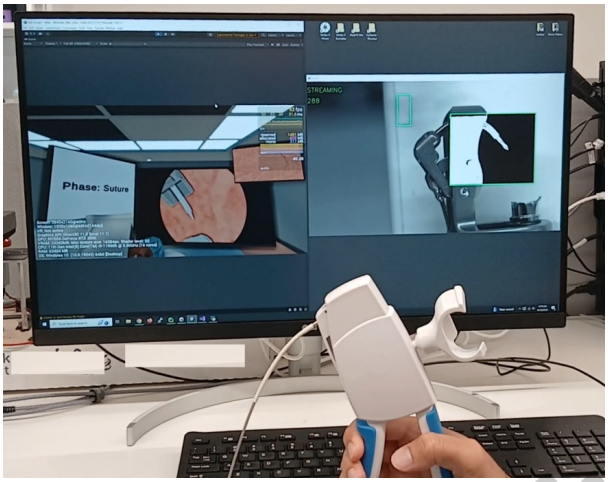

Fig. 14. Computer vision subsystem.

essential, and we strive to sustain a high frame rate while accurately replicating the real-life suturing technique during each tissue-pulling procedure within a VR setting.

The solver computation time refers to the duration needed for the XPBD solver to calculate the physics, constraints, and collisions for all simulated particles. Measurements for solver time, render time, and FPS were recorded on a per-frame basis during the tissue-pulling stage and after the procedure when the system returned to its initial state.

Render time represents the time taken for each frame to be processed and displayed on the screen, directly influencing the simulator's responsiveness. Shorter render times indicate a smoother visual experience by ensuring that graphical updates keep pace with real-time interactions. Similarly, FPS serves as a direct indicator of the system's capacity to maintain real-time performance. A higher FPS contributes to a more fluid and immersive VR experience, while a lower FPS can introduce lag and diminish realism.

All performance measurements represent averages calculated for each phase. Testing was conducted on a system equipped with an Intel(R) Core(TM) i7-12700KF 3.6 GHz CPU, 16 GB RAM, and NVIDIA GeForce RTX 3080. We evaluated two distinct scenarios to assess performance: in the first, eight sets of sutures (each containing seven sutures) were used, and in the second, 14 sets of sutures (each containing four sutures) were implemented. Although both scenarios involved a total of 56 sutures, data was recorded for each suture set during the tissue-pulling process. The FPS ranged between 55 and 59 in both scenarios, indicating that the simulator maintained smooth, real-time performance throughout. Figures 15 and 16 illustrate these FPS values, suggesting minimal performance degradation despite the computational demands of the tissue-pulling technique.

Furthermore, render times ranged from 3.9ms to 4.3ms, and solver times ranged from 16ms to 18ms, as shown in Figures 17 and 18. These results confirm that graphical updates were processed efficiently, ensuring a seamless visual experience, and that the system effectively handled real-time physics calculations without significant slowdowns.

3.2 Latency Results

In this section, we will discuss the latencies associated with a) the developed USB-based input devices, and b) the developed computer vision-based input devices. Both types of input devices are used in the simulator system.

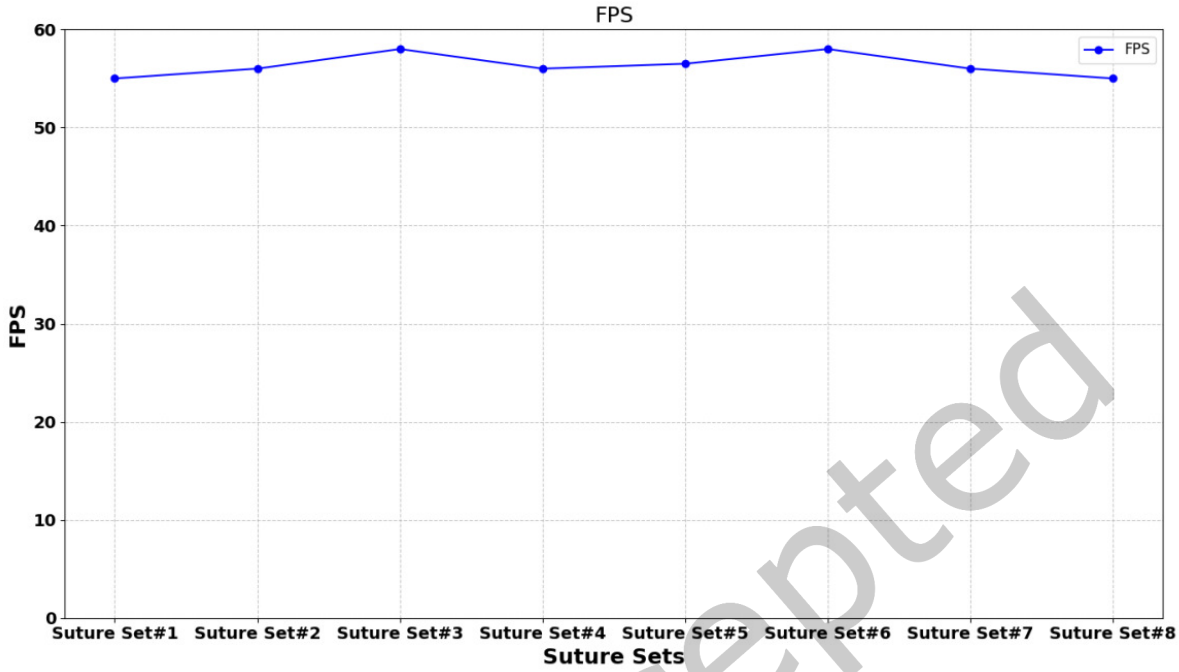

Fig. 15. Frame rate performance of the simulator featuring eight sets of sutures, each comprising seven sutures.

3.2.1 Latency associated with the USB input devices. The USB input devices described in Section 2.2 are ARM Cortex-M7 devices and operate as bare metal systems. Each Cortex-M7 device has core and peripheral bus clock periods well below the 1-microsecond level, and each of these is interfaced to two optical encoders using dedicated encoder peripherals. These special hardware units track encoder positions independent of the host Cortex-M7 software and, therefore, will have less than a 1-microsecond encoder position "read" delay.

One of the Cortex-M7 bare metal systems runs a 1000 Hz local haptic feedback loop using timer interrupts, therefore, the delay in haptic feedback will be at most a couple of milliseconds. This estimate does include the electrical response time of the motor drivers and the low inertia of the coreless DC motors. Without these overheads, the local haptic feedback response time will be 1ms. The haptic feedback software running on the Cortex-M7 also has a passivity guarantee to eliminate potential oscillations.

Both Cortex-M7 bare metal systems also run a USB stack at a 100 Hz update rate. Namely, encoder and button positions are reported to the host PC every 10ms over the USB bus using the HID protocol. In summary, we have two USB-based input devices, and from the host PC's side, they are seen as standard USB gamepads. The end user of the simulator may experience delays exceeding 10ms, possibly due to the load on the host PC or low-performance graphics hardware. However, the developed USB input devices and associated latencies will be as good as any other USB gamepad. Finally, the 100 Hz HID device update rate is configurable, but we haven't explored the effects of shorter or longer update periods on the overall simulator performance.

3.2.2 Latency assciated with the Computer Vision System. The latency associated with the computer vision system and its measurement is a relatively more involved topic. In this study, we performed three different latency measurements:

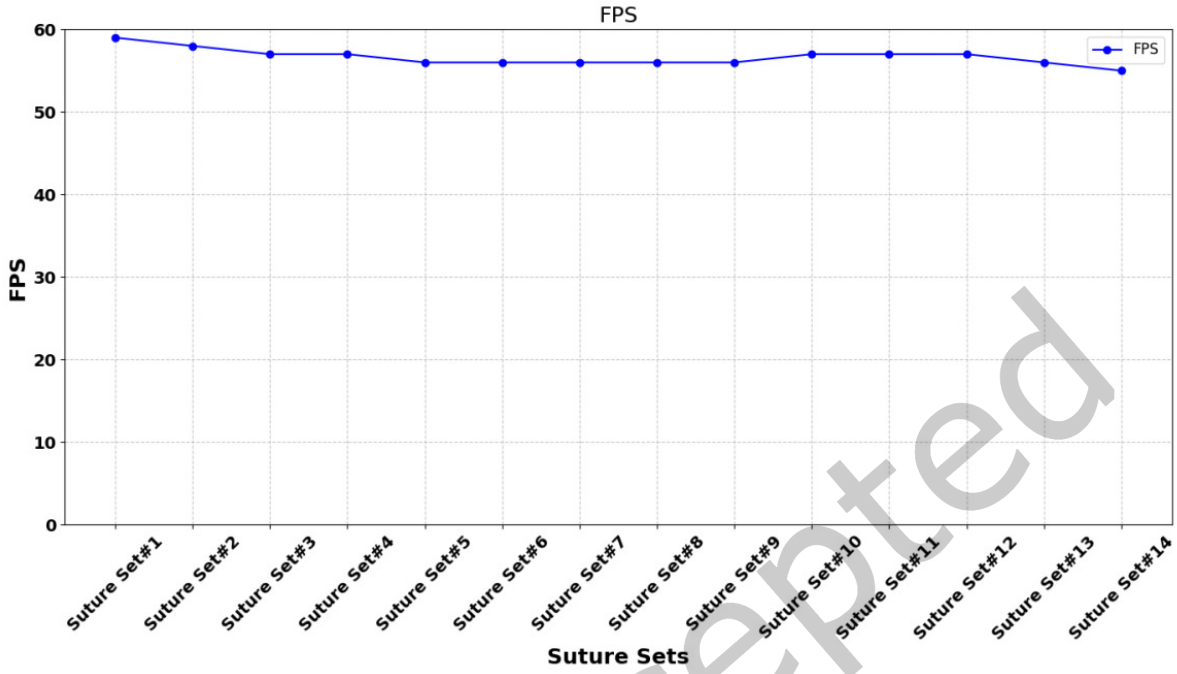

Fig. 16. Frame rate performance of the simulator featuring 14 sets of sutures, each comprising four sutures.

i. A 30-FPS image sensor directly interfaced with an FPGA.
ii. A 147-FPS global shutter Gigabit Ethernet camera interfaced to a host PC via a PCI Express.
iii. CA 30-FPS standard webcam interfaced to the host PC via USB.

For the first test, we used an image sensor directly interfaced with the FPGA. In this test setup, as seen in Figure 19, there is an image sensor (camera) directly connected to an FPGA monitor and continuously monitors the position of the hook (Surgical device). The FPGA board generates an electrical signal based on the detected position of the hook. The FPGA board also generates a VGA signal for monitoring/diagnostic purposes. The whole process is monitored by a global shutter 240 FPS GoPro observer. A frame-by-frame analysis of the GoPro recording is used to study the latency. Because of the 240-FPS rate, time measurement accuracy will be around 4ms levels. The FPGA camera is a low-end Omni Vision OV7670, 640x480 resolution, rolling shutter, and 30 FPS image sensor. The FPGA is a Xilinx Artix7-100T system that reads the camera output and does the following two tasks in parallel using dedicated hardware parallelism:

- The image sensor outputs data and sends it directly to the VGA monitor without any intermediate agent, system, computer, or software.
- The image sensor outputs data and performs a mathematical operation (In this case, it computes the average pixel intensity). The result of the mathematical operation must be ready after one (or a few) pixel clock cycles after completing the data transfer cycle. In this setup, the pixel clock cycle is below 0.1 microseconds.

The FPGA reads the image sensor output and based on what is seen by the image sensor, the FPGA board generates an output signal. Since the image sensor is 30 FPS, this process takes about 33ms. Using higher FPS image sensors or image sensors that support region of interest features will result in shorter data transfer cycles

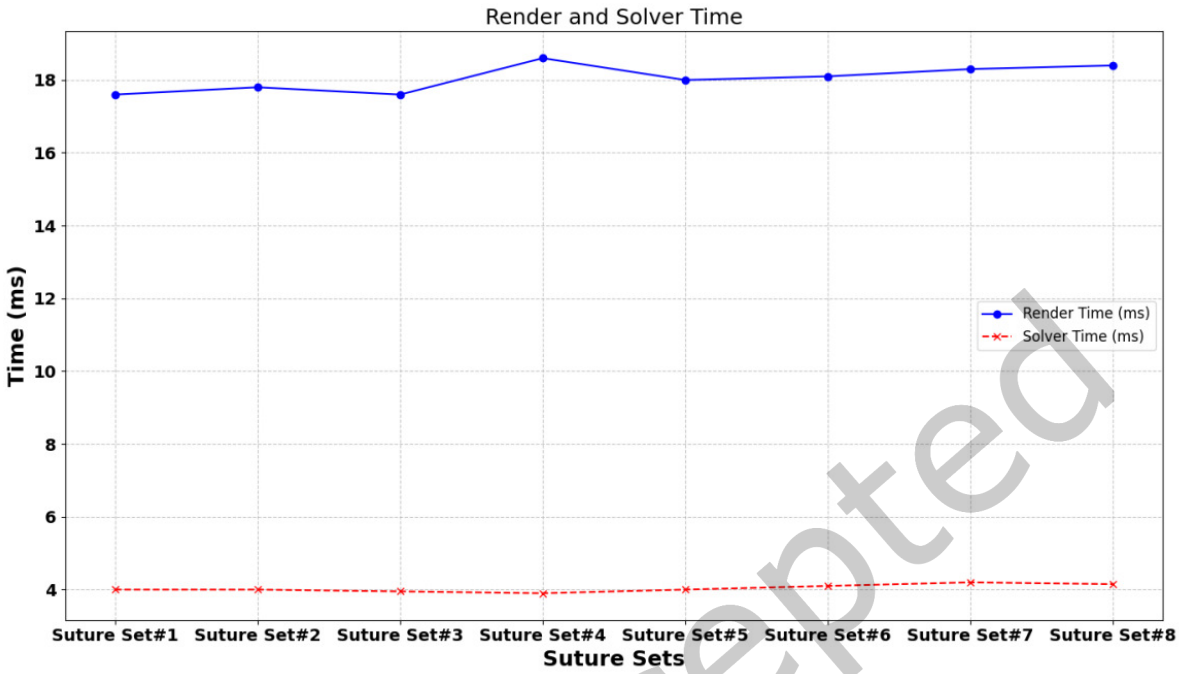

Fig. 17. Render and solver times, measured in milliseconds, for the simulator with eight sets of sutures, each consisting of 7 sutures.

and reduced latencies. Because of the rolling shutter feature of the particular image sensor, it is possible to barely miss the motion event in a single frame duration of 33ms. This may double the latency, but this problem can easily be mitigated by using a global shutter image sensor.

In the following figure, Figure 20, we present a portion of the frame-by-frame analysis of the GoPro observer. The blue line corresponds to the average pixel intensity seen by the GoPro observer for the randomly blinking yellow LED, and the red line corresponds to the average pixel intensity of the FPGA-controlled green LED seen by the GoPro observer. After this frame-by-frame analysis, the data is centered around 0 and normalized.

The marker locations are 48.650 seconds and 48.683 seconds; therefore, the latency of the computer vision system is 33ms. In other words, it is equal to the frame duration or the inverse of the frames per second rating of the image sensor.

For the next test, we used a global shutter gigabit ethernet camera (147-FPS) interfaced to the host PC via PCI Express. In this latency test, we used a 147-FPS global shutter Gigabit Ethernet camera interfaced to the host PC via PCI Express. We repeated the same experiment and observed a 74.85ms average delay with a standard deviation of 11.38ms. The test setup for Gigabit Ethernet camera latency can be seen in Figure 21, and the histogram of the latency can be seen in Figure 22.

In Figure 23, we present the measured latency versus time. As seen by the plot, latency changes with time, which is completely normal because of using a multitasking host OS. The measured latency rose to 120ms on a single occasion, but this was observed only once in almost 1000 random blinks.

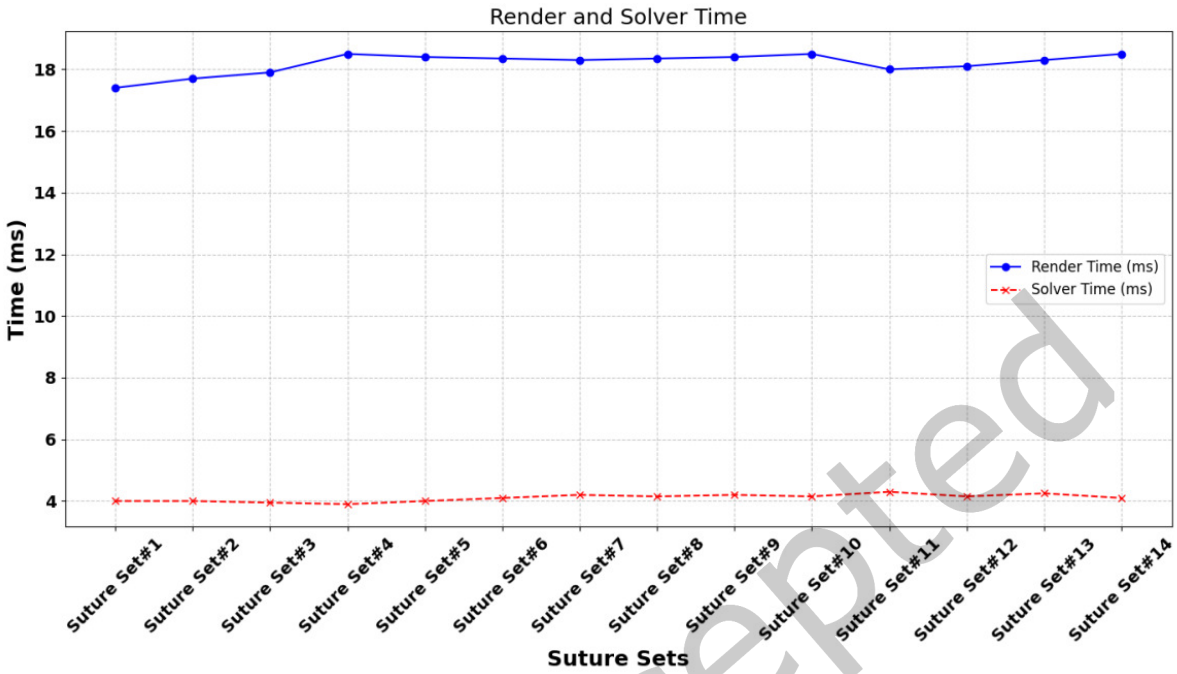

Fig. 18. Render and solver times, measured in milliseconds, for the simulator with 14 sets of sutures, each consisting of 4 sutures.

In the graphs in Figure 24, we present the timing waveforms corresponding to the average pixel intensity measured by the GoPro observer. As in the previous FPGA experiment, the video recording of the GoPro is analyzed frame by frame, and the waveforms shown in the following figures are obtained.

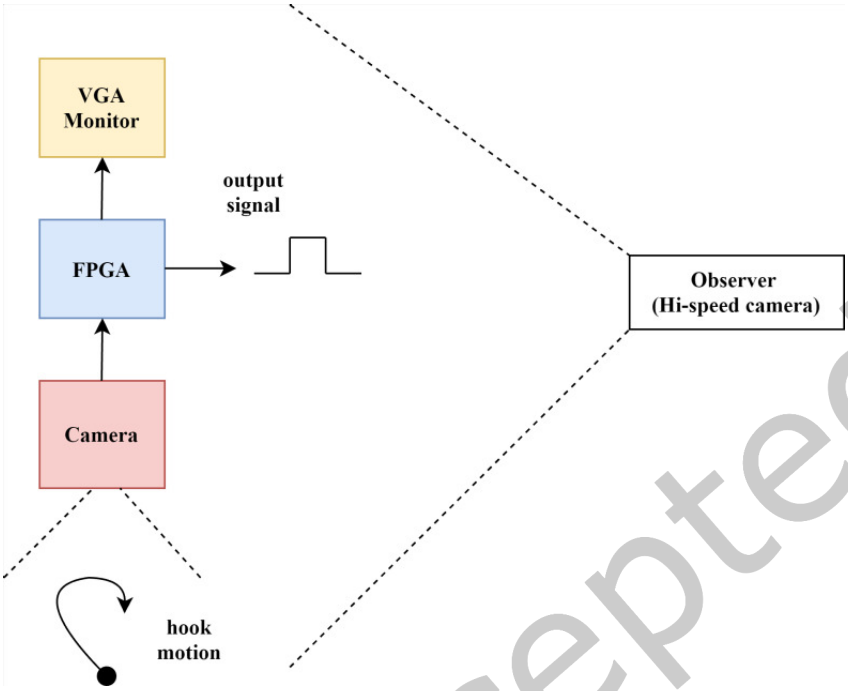

Fig. 19. FPGA-based Image Sensor latency measurement setup. A microcontroller board is used to generate a randomly blinking yellow LED stimulus. The FPGA is directly interfaced with the image sensor and continuously monitors the data output of the image sensor. As soon as a brightness level increase/decrease is detected based on the image sensor output, the green LED on the FPGA board is turned on or off. A GoPro camera running at 240 FPS is used as an observer.

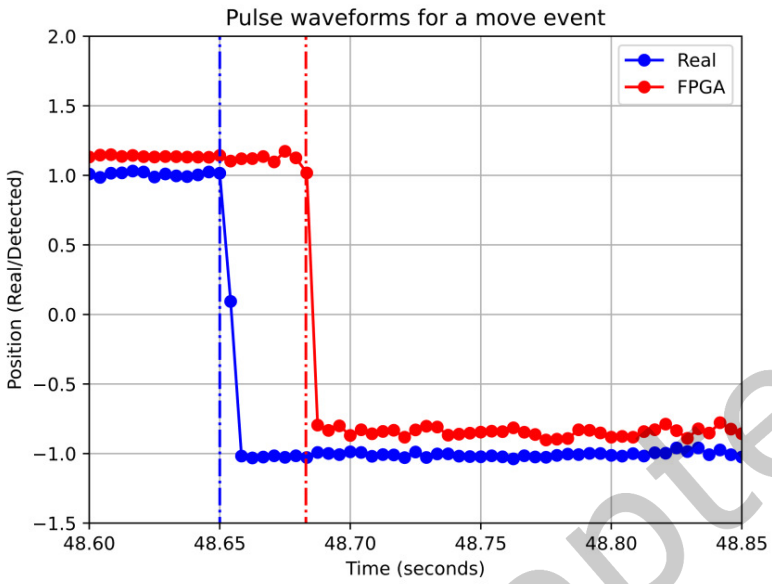

Fig. 20. The blue line corresponds to the average pixel intensity seen by the GoPro observer, and the red line corresponds to the FPGA output seen by the GoPro observer.

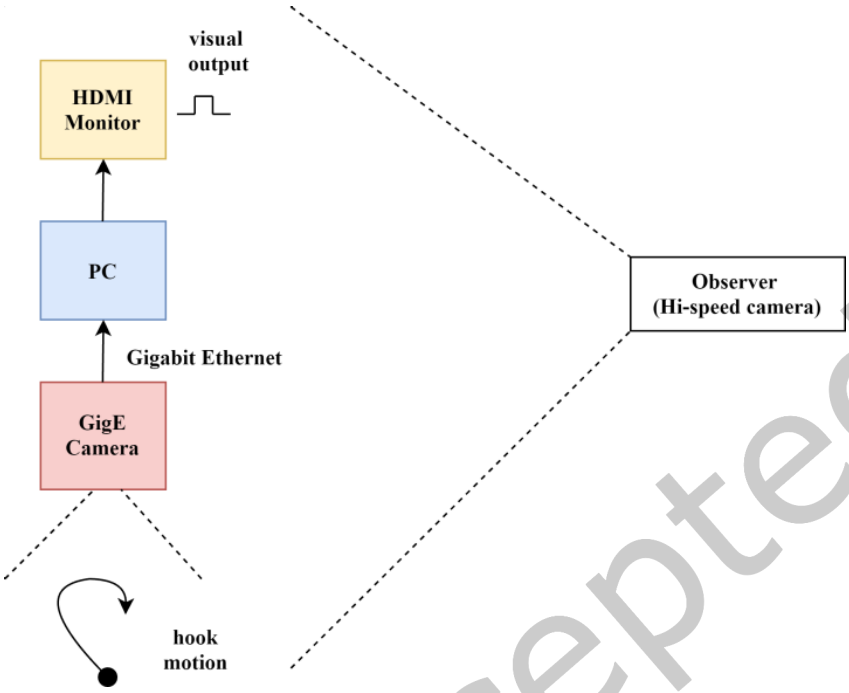

Fig. 21. Gigabit Ethernet camera latency test setup.

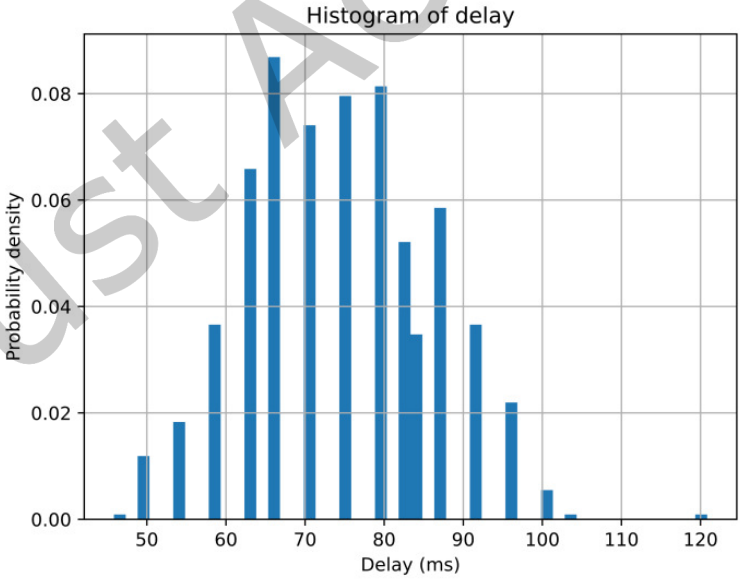

Fig. 22. Histogram of latency.

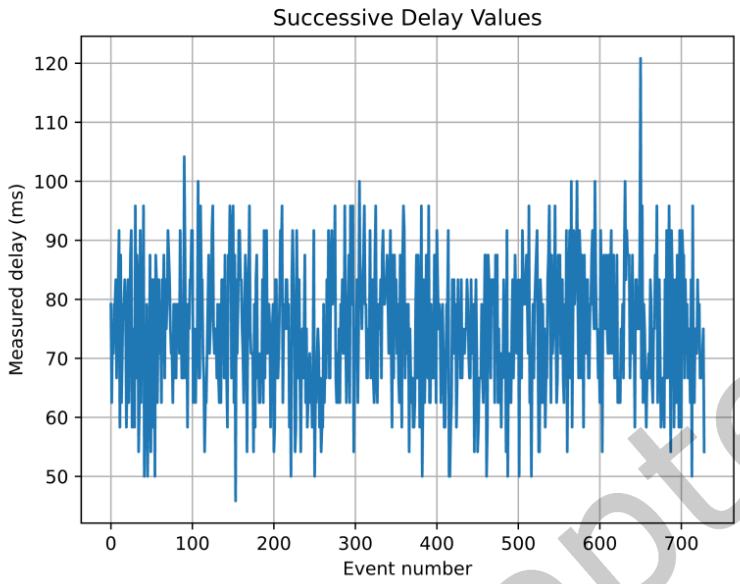

Fig. 23. Latency versus time.

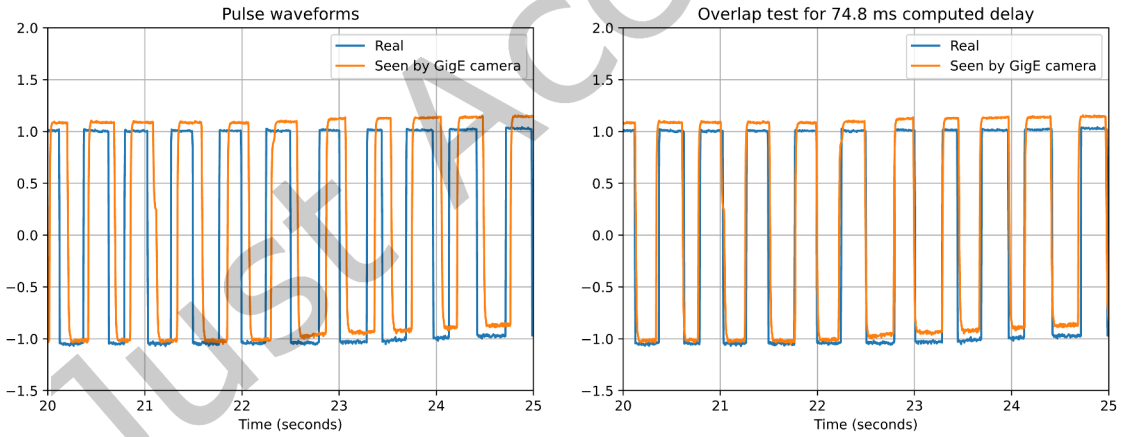

Fig. 24. Timing waveforms and almost perfect alignment after 74.83ms shift. This is used to verify our delay measurement.

The final test was with a webcam (30-FPS) interfaced to the host PC via USB. Our final latency test was done by using a USB webcam, and we measured latencies around 140ms on average. This average is obtained after 500 random blinking tests. The test setup and the procedure are similar to the Gigabit Ethernet camera case.

4 Discussion

In this section, we present a detailed discussion on system delay measurement, performance metrics, and computational performance.

4.1 System Delay Measurement

In our study, system delay is defined as the interval between the occurrence of an event and its visual representation on the simulator's monitor. When a movement event is initiated by the surgical tool, it is first detected directly and then later observed on the monitor after hardware and software processing delays. To rigorously quantify this delay, it is necessary to repeat the experiment multiple times to generate delay histograms and compute statistical measures such as the mean and standard deviation. This process can be both repetitive and tedious if done manually.

To streamline this process, we substituted the mechanical movement event of the surgical tool with an electronically triggered LED off event. This substitution simplified automation and maintained the dual-observation approach: one observation is made directly and the other on the simulator's monitor. A global shutter high-speed camera operating at 240 FPS was employed to capture the events, which is critical given the spatial differences between the LED and its corresponding pixels on the monitor. This setup allowed us to measure event timing with an error margin of approximately 4ms per frame. By analyzing the high-speed footage using image processing scripts, we identified the frame in which the LED turns off both directly and on the monitor. For instance, if the LED is observed to turn off at frame 1500 directly and at frame 1520 on the monitor, the delay is calculated as 20 frames (i.e., $20 \times 1/240$ seconds). This automated approach ensured robust and repeatable delay measurements, as a result enhanced the reliability of our system performance evaluation.

4.2 Performance Metrics

In line with previous findings [22], system delay has a significant impact on overall performance. Delays of up to 100ms are generally deemed acceptable; however, minimizing latency remains a priority. In [38], dV-Trainer, an immersive surgical simulator that provides realistic haptic feedback, was used to evaluate the impact of latency, with results indicating that latencies below 200ms were ideal. For instance, a delay of 300ms results in a perceptible lag between the actuation of a surgical instrument and its visual representation on the simulator's HMD. This lag can disrupt the operator's hand-eye coordination, compelling them to slow their movements and thereby reducing the simulator's overall efficacy. Consequently, precise delay measurement and maintaining latencies below 100ms are critical.

Additionally, frame rate is a key performance metric of immersive systems. Frame rates falling below 24 FPS can impair hand-eye coordination, leading to a slowdown in task execution and a consequent decline in simulator effectiveness [8, 17]. Based on these considerations, our criteria for successful system performance are defined as a system delay below 100ms, and a frame rate significantly exceeding 24 FPS during the simulation. By these metrics, the developed system meets the defined standards for success.

4.3 Computational Performance

The results show that increasing the number of suture sets has a minor effect on the frame rate performance of the simulator. Specifically, the simulator with 14 suture sets achieves an average FPS of 54.9, while the simulator with eight suture sets maintains a relatively stable average FPS of 55.8. Changes do not influence the performance

of the pulling procedure in suture sets or suture counts. The solver time appears consistent across different suture sets in both simulators, showing that the computational complexity for solving remains relatively constant. The simulator with eight suture sets exhibits a computational time of 17.7ms, while the 14 suture sets show 18ms, suggesting that the computational complexity may rise with the number of sutures per set. In general, render times remain consistently low across all tests, indicating that variations do not significantly influence the simulator's performance in suture sets or counts. We have observed end-to-end delay below 75ms and achieved a 1000 Hz haptic feedback loop. In our current setup, the spring and damper parameters of the haptic feedback are fixed. However, we plan to develop a more advanced setup where the game engine can control these spring-damper parameters and, hence, control the level and type of haptic feedback.

5 Conclusion

In this work, we presented the design and development of the ViBE simulator, which replicates the steps of the ESG within a virtual environment. The objective of this work is to enhance the learning curve for endoscopic suturing and ESG techniques, ultimately extending these skills safely to a broader patient base. The simulator demonstrates robust performance, maintaining an average frame rate of 55 FPS, low render times, and consistent solver times. The end-to-end delay of the hardware is below 75ms, ensuring responsiveness. Furthermore, the haptic feedback system provides precise tactile feedback with a 1000 Hz update frequency. These features underscore the ViBE simulator's potential as an effective training tool for ESG. Planned enhancements, such as dynamic control of haptic feedback, will further improve its fidelity. Overall, the ViBE simulator offers a cost-effective and realistic platform for ESG training, enhancing surgeon proficiency and ultimately benefiting patients through improved surgical outcomes.

Acknowledgments

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

References

[1] Ahmad Aziz, Muhammad Farhan, Sahaab Noor, Sudhair Alam, Malik Olatunde Oduoye, and Abdullah Bin Kamran. 2024. Virtual reality and simulation-based training in Pakistan for uniformity in neurosurgery training programs. *Neurosurgical Review* 47, 1 (May 2024), 244. doi:10.1007/s10143-024-02474-1

[2] John D. Birkmeyer, Jonathan F. Finks, Amanda O'Reilly, Mary Oerline, Arthur M. Carlin, Andre R. Nunn, Justin Dimick, Mousumi Banerjee, and Nancy J.O. Birkmeyer. 2013. Surgical Skill and Complication Rates after Bariatric Surgery. *New England Journal of Medicine* 369, 15 (Oct. 2013), 1434–1442. doi:10.1056/NEJMsa1300625

[3] Sanne M. B. I. Botden, Sonja N. Buzink, Marlies P. Schijven, and Jack J. Jakimowicz. 2007. Augmented versus Virtual Reality Laparoscopic Simulation: What Is the Difference? *World Journal of Surgery* 31, 4 (April 2007), 764–772. doi:10.1007/s00268-006-0724-y

[4] Sanne M. B. I. Botden and Jack J. Jakimowicz. 2009. What is going on in augmented reality simulation in laparoscopic surgery? *Surgical Endoscopy* 23, 8 (Aug. 2009), 1693–1700. doi:10.1007/s00464-008-0144-1

[5] Barham K. Abu Dayyeh, Andres Acosta, Michael Camilleri, Manpreet S. Mundi, Elizabeth Rajan, Mark D. Topazian, and Christopher J. Gostout. 2017. Endoscopic Sleeve Gastroplasty Alters Gastric Physiology and Induces Loss of Body Weight in Obese Individuals. *Clinical Gastroenterology and Hepatology* 15, 1 (Jan. 2017), 37–43.e1. doi:10.1016/j.cgh.2015.12.030

[6] Luiz Gonzaga de Moura Júnior, Paulo Roberto Leitão de Vasconcelos, Francisco Vagnaldo Fechine, Mayra Sabiá de Moura, Régis Luiz Sabiá de Moura, Hermano Alexandre Lima Rocha, and Manoel Odorico de Moraes Filho. 2018. Endosuture trainer box simulator as a tool for training and teaching in bariatric laparoscopic surgery. *BMC Surgery* 18, 1 (Oct. 2018), 83. doi:10.1186/s12893-018-0412-5

[7] James Dials, Doga Demirel, Tansel Halic, Suvranu De, Adam Ryason, Shanker Kundumadam, Mohammad Al-Haddad, and Mark A. Gromski. 2022. Hierarchical task analysis of endoscopic sleeve gastroplasty. *Surgical Endoscopy* 36 (2022), 5167–5182. doi:10.1007/s00464-021-08893-1

[8] Amilcar do Carmo Lucas, Sven Heithecker, Peter Rüffer, Rolf Ernst, Holger Rückert, Gerhard Wischermann, Karin Gebel, Reinhard Fach, Wolfgang Huther, Stefan Eichner, and Gunter Scheller. 2006. 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*. IEEE, 1–6. doi:10.1109/DATE.2006.244085

[9] Manoel dos P. Galvão-Neto, Eduardo Grecco, Thiago Ferreira de Souza, Luiz Gustavo de Quadros, Lyz Bezerra Silva, and Josemberg Marins Campos. 2016. ENDOSCOPIC SLEEVE GASTROPLASTY - MINIMALLY INVASIVE THERAPY FOR PRIMARY OBESITY TREATMENT. *ABCD Arq. Bras. Cir. Dig. São Paulo* 29 (2016), 95–97. doi:10.1590/0102-6720201600S10023

[10] A. J. Duffy, N. J. Hogle, H. McCarthy, J. I. Lew, A. Egan, P. Christos, and D. L. Fowler. 2005. Construct validity for the LAPSIM laparoscopic surgical simulator. *Surgical Endoscopy and Other Interventional Techniques* 19, 3 (March 2005), 401–405. doi:10.1007/s00464-004-8202-9

[11] Utku Erden, Doga Demirel, Onur Toker, Suvranu De, and Mark A. Gromski. 2023. S1644 Development of a Virtual Reality-Based Endoscopic Sleeve Gastroplasty Simulator. *Official journal of the American College of Gastroenterology| ACG* 118, 10S (2023), S1231. doi:10.14309/01.ajg.0000956216.41760.6c

[12] Utku Erden, Mark A. Gromski, Suvranu De, and Doga Demirel. 2024. Preliminary Validation of the Virtual Bariatric Endoscopic Simulator. *iGIE* 3, 4 (2024), 453–462. doi:10.1016/j.igie.2024.08.003

[13] Domenico Giannotti, Gregorio Patrizi, Giovanni Casella, Giorgio Di Rocco, Massimiliano Marchetti, Francesca Frezzotti, Maria Giulia Bernieri, Anna Rita Vestri, and Adriano Redler. 2014. Can virtual reality simulators be a certification tool for bariatric surgeons? *Surgical Endoscopy* 28, 1 (Jan. 2014), 242–248. doi:10.1007/s00464-013-3179-x

[14] Tansel Halic, Suvranu De, James Dials, Mark A. Gromski, Doga Demirel, Adam Ryason, Ashley C. Gilmore, Mohammad A. Al-Haddad, and Shanker Kundumadam. 2020. S1191 Task Analysis and Performance Metrics of Endoscopic Sleeve Gastroplasty: Preparation for Virtual Simulation Development. *Official journal of the American College of Gastroenterology ACG* 115 (Oct. 2020), S595. doi:10.14309/01.ajg.0000706812.30100.05

[15] Peter Halvax, Balazs Nemeth, Istivan Kiss, Andras Papp, and Andras Vereczkei. 2023. Endoluminal Suture-technique for the Stomach Closure of an Experimental Model. *Anticancer Res.* 43, 1 (Jan. 2023), 59–61. doi:10.21873/anticanres.16134

[16] Christine Hill, Mohamad El Zein, Abhishek Agnihotri, Margo Dunlap, Angela Chang, Alison Agrawal, Sindhu Barola, Saowanee Ngamruengphong, Yen-I Chen, Anthony N. Kalloo, Mouen A. Khashab, and Vivek Kumbhari. 2017. Endoscopic sleeve gastroplasty: the learning curve. *Endoscopy international open* 5, 9 (Sept. 2017), E900–E904. doi:10.1055/s-0043-115387

[17] Yasar C. Kakdas, Doga Demirel, Jacob R. Barker, Tansel Halic, Jeffrey Keane, John Mitchell, Stephanie Jones, Daniel B. Jones, Suvranu De, and Cullen Jackson. 2024. Virtual Reality-Based Donning and Doffing Simulator. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 214–222. doi:10.1109/VRW62533.2024.00043

[18] Murali Karnam, Marek Zelechowski, Philippe C. Cattin, Georg Rauter, and Nicolas Gerig. 2025. User-specified inverse kinematics taught in virtual reality reduce time and effort to hand-guide redundant surgical robots. *Commununications Engineering* 4, 1 (2025), 20. doi:10.1038/s44172-025-00357-x

[19] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. 2007. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games (I3D '07)*. Association for Computing Machinery, New York, NY, USA, 39–46. doi:10.1145/1230100.1230107

[20] Hermann J. Kissler and Utz Settmacher. 2013. Bariatric Surgery to Treat Obesity. *Semin. Nephrol.* 33, 1 (Jan. 2013), 75–89. doi:10.1016/j.semnephrol.2012.12.004

[21] Lan Li, Fei Yu, Dongquan Shi, Jianping Shi, Zongjun Tian, Jiquan Yang, Xingsong Wang, and Qing Jiang. 2017. Application of virtual reality technology in clinical medicine. *American Journal of Translational Research* 9, 9 (Sept. 2017), 3867–3880.

[22] Yang Li, Nicholas Raison, Sebastien Ourselin, Toktam Mahmoodi, Prokar Dasgupta, and Alejandro Granados. 2024. AI solutions for overcoming delays in telesurgery and telementoring to enhance surgical practice and education. *Journal of Robotic Surgery* 18, 1 (Nov. 2024), 403. doi:10.1007/s11701-024-02153-9

[23] Edward Luke, Eric Collins, and Eric Blades. 2012. A fast mesh deformation method using explicit interpolation. *J. Comput. Phys.* 231, 2 (Jan. 2012), 586–601. doi:10.1016/j.jcp.2011.09.021

[24] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. Association for Computing Machinery, New York, NY, USA, 49–54. doi:10.1145/2994258.2994272

[25] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified particle physics for real-time applications. *ACM Transactions on Graphics* 33, 4 (July 2014), 153:1–153:12. doi:10.1145/2601097.2601152

[26] Jeffrey I. Mechanick, Adrienne Youdim, Daniel B. Jones, W. Timothy Garvey, Daniel L. Hurley, Molly McMahon, Leslie J. Heinberg, Robert Kushner, Ted D. Adams, Scott Shikora, John B. Dixon, and Stacy Brethauer. 2013. Clinical practice guidelines for the perioperative nutritional, metabolic, and nonsurgical support of the bariatric surgery patient—2013 update: cosponsored by American Association of Clinical Endocrinologists, the Obesity Society, and American Society for Metabolic & Bariatric Surgery. *Surgery for Obesity and Related Diseases* 9, 2 (March 2013), 159–191. doi:10.1016/j.soard.2012.12.010

[27] Matthias Müller, Jan Bender, Nuttapong Chentanez, and Miles Macklin. 2016. A robust method to extract the rotational part of deformations. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. Association for Computing Machinery, New York, NY, USA, 55–60. doi:10.1145/2994258.2994269

[28] Matthias Müller and Nuttapong Chentanez. 2011. Solid simulation with oriented particles. In *ACM SIGGRAPH 2011 papers (SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/1964921.1964987
[29] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. 2005. Meshless deformations based on shape matching. *ACM Transactions on Graphics* 24, 3 (July 2005), 471–478. doi:10.1145/1073204.1073216
[30] Tatsuma Nomura, Shinya Sugimoto, Taishi Temma, Jun Oyamada, Keichi Ito, and Akira Kamei. 2023. Suturing techniques with endoscopic clips and special devices after endoscopic resection. *Digestive Endoscopy* 35, 3 (2023), 287–301. doi:10.1111/den.14427
[31] Cynthia L. Ogden, Margaret D. Carroll, Brian K. Kit, and Katherine M. Flegal. 2014. Prevalence of Childhood and Adult Obesity in the United States, 2011-2012. *JAMA* 311, 8 (Feb. 2014), 806–814. doi:10.1001/jama.2014.732
[32] Lucian Panait, Ehab Akkary, Robert L. Bell, Kurt E. Roberts, Stanley J. Dudrick, and Andrew J. Duffy. 2009. The Role of Haptic Feedback in Laparoscopic Simulation Training. *Journal of Surgical Research* 156, 2 (Oct. 2009), 312–316. doi:10.1016/j.jss.2009.04.018
[33] Michael A. Scaffidi, Rishad Khan, Catharine M. Walsh, Matthew Pearl, Kathleen Winger, Ruben Kalaichandran, Peter Lin, and Samir C. Grover. 2019. Protocol for a randomised trial evaluating the effect of applying gamification to simulation-based endoscopy training. *BMJ Open* 9, 2 (2019), e024134. doi:10.1136/bmjopen-2018-024134
[34] Francisco Schlottmann, Fernando A. M. Herbella, and Marco G. Patti. 2021. Simulation for Foregut and Bariatric Surgery: Current Status and Future Directions. *Journal of Laparoendoscopic & Advanced Surgical Techniques* 31, 5 (May 2021), 546–550. doi:10.1089/lap.2021.0080
[35] Reem Z. Sharaiha, Nikhil A. Kumta, Monica Saumoy, Amit P. Desai, Alex M. Sarkisian, Andrea Benevenuto, Amy Tyberg, Rekha Kumar, Leon Igel, Elizabeth C. Verna, Robert Schwartz, Christina Frissora, Alpana Shukla, Louis J. Aronne, and Michel Kahaleh. 2017. Endoscopic Sleeve Gastroplasty Significantly Reduces Body Mass Index and Metabolic Complications in Obese Patients. *Clinical Gastroenterology and Hepatology* 15, 4 (April 2017), 504–510. doi:10.1016/j.cgh.2016.12.012
[36] Keith Siau, James Hodson, Peter Neville, Jeff Turner, Amanda Beale, Susi Green, Aravinth Murugananthan, Paul Dunckley, and Neil D Hawkes. 2020. Impact of a simulation-based induction programme in gastroscopy on trainee outcomes and learning curves. *World journal of gastrointestinal endoscopy* 12, 3 (2020), 98. doi:10.4253/wjge.v12.i3.98
[37] Konstantinos Triantafyllou, Lazaros Dimitrios Lazaridis, and George D. Dimitriadis. 2014. Virtual reality simulators for gastrointestinal endoscopy training. *World journal of gastrointestinal endoscopy* 6, 1 (2014), 6. doi:10.4253/wjge.v6.i1.6
[38] Song Xu, Manuela Perez, Kun Yang, Cyril Perrenot, Jacques Felblinger, and Jacques Hubert. 2014. Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer® simulator. *Surgical Endoscopy* 28, 9 (Sept. 2014), 2569–2576. doi:10.1007/s00464-014-3504-z
[39] Wei-Wei Xu and Kun Zhou. 2009. Gradient Domain Mesh Deformation — A Survey. *Journal of Computer Science and Technology* 24, 1 (Jan. 2009), 6–18. doi:10.1007/s11390-009-9209-4
[40] Eugenia Yiannakopoulou, Nikolaos Nikiteas, Despina Perrea, and Christos Tsigris. 2015. Virtual reality simulators and training in laparoscopic surgery. *International Journal of Surgery* 13 (Jan. 2015), 60–64. doi:10.1016/j.ijsu.2014.11.014
[41] Wenlan Zhang, Xin Liu, and Bin Zheng. 2021. Virtual reality simulation in training endoscopic skills: a systematic review. *Laparoscopic, Endoscopic and Robotic Surgery* 4, 4 (2021), 97–104. doi:10.1016/j.lers.2021.09.002

Received 30 September 2024; revised 9 June 2025; accepted 23 June 2025