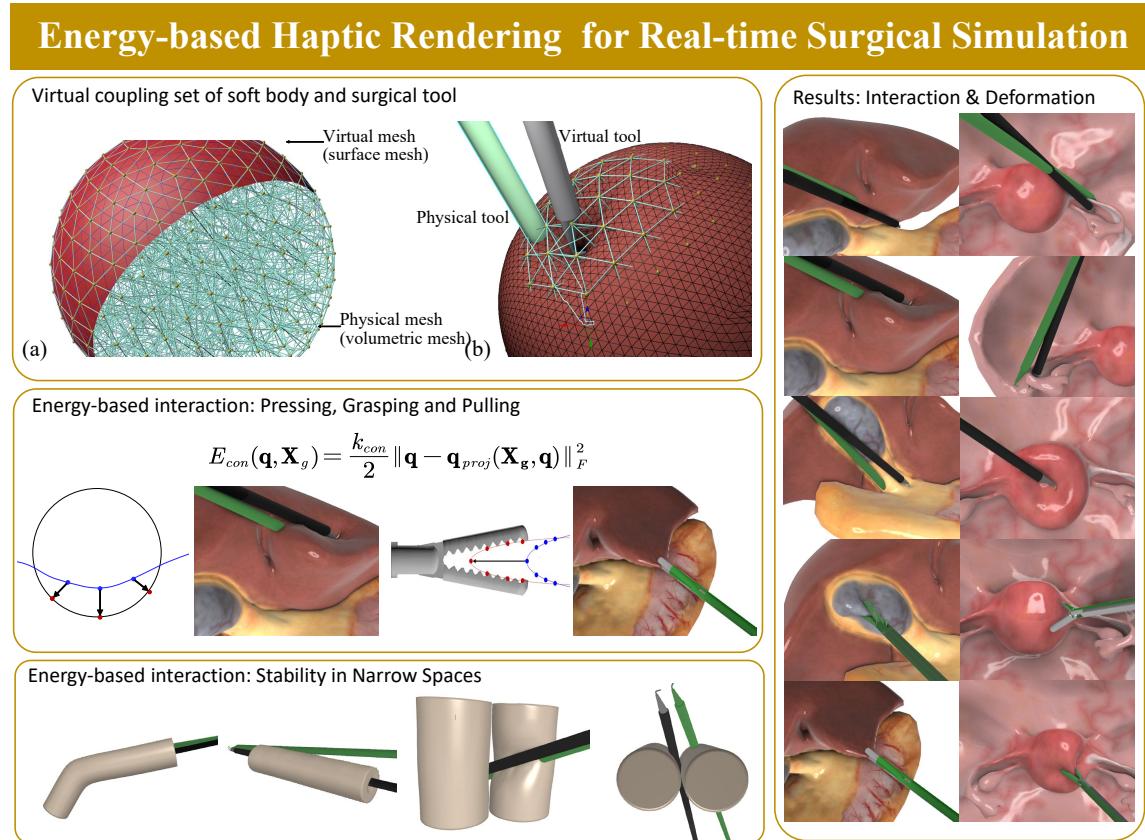


Graphical Abstract

Energy-based Haptic Rendering for Real-time Surgical Simulation

Lei He, Mingbo Hu, Wenli Xiu, Hongyu Wu, Siming Zheng, Shuai Li, Qian Dong, Aimin Hao



Highlights

Energy-based Haptic Rendering for Real-time Surgical Simulation

Lei He, Mingbo Hu, Wenli Xiu, Hongyu Wu, Siming Zheng, Shuai Li, Qian Dong, Aimin Hao

- Providing an energy-based simulation system with smooth and high-frequency haptic interaction and large deformation.
- A novel simulation model for soft bodies using virtual coupling.
- A simulation system with multiple virtual coupling objects to enable interaction among systems with differing update rates.

Energy-based Haptic Rendering for Real-time Surgical Simulation

Lei He^a, Mingbo Hu^a, Wenli Xiu^b, Hongyu Wu^{a,*}, Siming Zheng^c, Shuai Li^a, Qian Dong^b, Aimin Hao^a

^a*State Key Laboratory of Virtual Reality Technology and Systems, QRI & SCSE, Beihang University, Beijing, 100191, Beijing, China*

^b*Department of Pediatric Surgery, The Affiliated Hospital of Qingdao University, Qingdao, 266000, Shandong, China*

^c*Department of Hepatobiliary and Pancreatic Surgery, The First Affiliated Hospital of Ningbo University, Ningbo, 315000, Zhejiang, China*

Abstract

Haptic-based surgical simulation is widely utilized for training surgical skills. However, simulating the interaction between rigid surgical instruments and soft tissues presents significant technical challenges. In this paper, we propose an energy-based haptic rendering method to achieve both large deformations and rigid-soft haptic interaction. Different from existing methods, both the rigid tools and soft tissues are modeled by an energy-based virtual coupling system. The constraints of soft deformation, tool-object interaction and haptic rendering are defined by potential energy. Benefit from energy-based constraints, we can realize complex surgical operations, such as inserting tools into soft tissue. The virtual coupling of soft tissue enables the separation of haptic interaction into two components: soft deformation with high computational complexity, and high-frequency haptic rendering. The soft deformation with shape constraints is accelerated GPU at a relatively low frequency($60\text{Hz} \sim 100\text{Hz}$), while the haptic rendering runs in another thread at a high frequency ($\geq 1000\text{Hz}$). We have implemented haptic simulation for two commonly used surgical operations, pressing and pulling. The experimental results show that our method can achieve stable feedback force and non-penetration between the tool and soft tissue under the condition of large soft deformation. The code is released on <https://github.com/helei0147/EnergyBasedHaptic.git>

Keywords: Haptic Rendering, Soft Dynamics, Surgical Simulation.

*Corresponding author: whyvrlab@buaa.edu.cn

1. Introduction

Virtual reality-based surgical simulation is a powerful tool for training fundamental surgical skills. By creating highly realistic virtual environments, surgeons can practice various procedures without any risk, thus enhancing their technical proficiency and decision-making skills. This simulation technology not only replicates the visual and haptic feedback of real surgeries but also mimics complex surgical scenarios and unexpected situations, allowing doctors to gain experience in a safe and controlled setting.

In the context of virtual surgery simulation, the interaction between surgical tools and soft tissues is a fundamental aspect of simulation environments. However, simulating the interaction between rigid tools and soft tissues poses a significant challenge. This difficulty arises from the need to simultaneously achieve high-frequency haptic rendering and realistic soft tissue deformation, which remain insufficiently addressed. Addressing this issue is crucial for advancing the realism and utility of virtual surgical training platforms.

Force feedback is a critical element in virtual surgical simulation, as it delivers haptic sensations that significantly enhance the realism of the experience. Research indicates that a haptic rendering rate of at least 500 Hz is necessary to achieve smooth and natural human perception [1], whereas visual updates typically require only around 60 Hz. Current popular methods, such as force-based haptic rendering [2], enable real-time rigid-soft interactions through reduced deformable models. However, these methods are unable to achieve large-scale deformations comparable to those of soft biological organs. On the other hand, position-based haptic rendering [3], which facilitates rigid-rigid interactions via hard no-penetration constraints, often fails when the tool is inserted into narrow spaces between soft tissues—a scenario frequently encountered in virtual surgical simulations.

In this paper, we propose an energy-based virtual coupling system to achieve high frequency haptic interaction between rigid tool and soft tissue. Specifically, the avatar directly controlled by the haptic device is called the physical tool. The virtual tool is defined as the non-penetration pose of tool. Similarly, the simulation of soft bodies involves two interconnected copies: the physical soft body and the virtual soft body, linked by virtual coupling constraint. All the constraints of our system are defined by elastic potential energy. To minimize the energy of the system, we adopt Projective Dynamics [4] to update pose of soft body and surgical tool. Since the elastic potential energy is time-independent, soft deformation and haptic rendering can be executed asynchronously. This flexibility accommodates high-frequency tool position update and GPU accelerated soft deformation.

The main contributions of this paper are:

- Providing an energy-based simulation system with smooth and high-frequency haptic interaction and large deformation.
- A novel simulation model for soft bodies using virtual coupling system.
- A simulation system with virtual coupling objects to enable interaction among systems with differing update rates.

2. Related Works

2.1. Surgery Simulation

Virtual surgery is an emerging interdisciplinary field that integrates various complex disciplines. The biomechanical characteristics of soft tissue is measured to reproduce the characteristics of soft tissue in simulation system. Mechanical property of tissues implies elasticity, stiffness or apparent viscosity. Several previous works [5][6] provides methods to measure the mechanical characteristics of soft tissue. After the measurement, we can use these characteristics in virtual surgery system to get more realism of simulation.

Virtual surgery platforms can simulate soft body deformation and give operator haptic force feedback to bring more sense of presence. Existing virtual surgery platforms includes vascular suturing virtual surgery system developed by Stanford University[7], virtual surgery system based on the filled-ball model[8]. These years significant progress has been made in visual representation and haptic simulation in virtual surgery systems proposed by[9][10][11] [12]. A review of simulation training for robotic surgery education can be found in [13].

2.2. Soft Body Dynamics

Several approaches have been proposed to simulate the soft object deformation during haptic interaction, such as mass-spring method[14] and shape matching method[15].

The basic principle of soft body deformation is that the resistance of an elastic object to deformations is quantified using an elastic potential energy. Variational derivative of elastic energy leads to the elastic force[16]. But elastic forces are usually nonlinear even for simple elastic materials. This non-linearity of force makes the motion of points more complicated. Finite element method has become the mainstream method for haptics-based deformation simulation, such as co-rotational linear FEM[17], extended FEM[18], and so on.

Position based dynamics[19] uses the number of iterations to control how strictly the constraints are enforced and how stiff an object behaves. The dependency on

iteration number is solved in XPBD[20] by introducing a Lagrange multiplier. Its convergence rate drops as the mesh resolution increases. To speed up the convergence, researchers proposed to enforce the constraints in a multi-resolution fashion[21]. [22][23] use graph coloring to maximize parallelism of simulation.

Projective dynamics(PD)[4] is a simulation method for deformable material based on energy constraints projection, which has good physical significance. One of the most important characteristic of projective dynamics is that it use a local-global solver to project the vertex to acceptable place for all the related constraints. Wang pointed out in [24] that the Chebyshev approach and the Jacobi iterative method outperform any other combination on GPU, as it is highly compatible with parallel computing. Lan et al. [25] propose a novel GPU algorithm named A-Jacobi for faster linear solve at the global step of PD.

2.3. Haptic Rendering

Early haptic rendering methods focus mainly on rigid-rigid interaction. However, haptic rendering for deformable objects is more challenging and has attracted more attention in recent years. There are two difficulties in the progress of haptic rendering for deformable objects, the first one is physics simulation for deformable object is more expensive than dynamic simulation of rigid object; The second difficulty is high-haptic update rate cannot be easily achieved for complex deformable models.

Haptic rendering can provide feedback force during the simulation, which will make the user interact with objects in simulation system with both visual and haptic feedback in real-time. Classical algorithms for 3 DOF haptic rendering include methods such as [1][26]. Current methods for 6 DOF haptic rendering are discussed in [27][2][28][29][30]. These years, new haptic devices for virtual surgery immerge, such as [31][32][33], these devices may provide more realism in virtual surgery and extend the haptic feedback force algorithm.

Two main updating methods have been researched for computing the pose of the haptic tool: dynamic simulation with numerical integration [34][35] and quasi-static equilibrium with integration (the QSA approach) [36][37][2]. In [37], collision forces and torques are calculated for each elemental section of the object. These forces and torques are then applied to the tool and output directly to the haptic device. However, this method can introduce instability due to rapid fluctuations in collision states. The virtual coupling method [34] was proposed to address the instability issues associated with penalty-based methods. This approach maintains a virtual object separate from the physical object manipulated by the user with a manipulandum, calculating forces and torques based on the differences between the two objects. Notable works employing virtual coupling include [2], [27], and [38].

3. Energy-based Virtual Coupling

Our energy-based virtual coupling simulation system takes the tool pose information from haptic device and the 3D model of soft body as input of the system. After deformation of soft body and movement of surgical tool, the system will render the virtual copy of soft body and surgical tool and output the feedback force to haptic device. In this section, we will introduce our energy-based virtual coupling in simulation system, including the definition of virtual coupling set and constraint energy.

3.1. Virtual Coupling for Soft and Rigid Objects

Virtual coupling technology is frequently used in automated control systems to achieve stability. Haptic rendering requires a high frequency simulation ($\geq 1000\text{Hz}$). However it is difficult to achieve such a high frame rate for the volumetric soft dynamics. We use the virtual coupling system to simulate the haptic interaction between rigid tool and soft body. As Figure 1 shows, The virtual coupling of soft body is an asynchronous simulation:

- The physical mesh is employed for shape-constrained deformation to maintain the shape of soft body, which is simulated at a relatively low frequency. Physical represents this mesh is used to keep the physical characteristic of soft body.
- The virtual mesh is used for surface tension constrained deformation and haptic interaction with surgical tools, which requires less computation for distance constraint. This allows the surface mesh to be simulated at high frequency. Virtual means this mesh is for interactions and display in virtual simulation space.

Furthermore, the virtual mesh with a higher vertex density allows for more fined surface deformation and more accurate collision detection. To ensure consistent deformation between the two meshes, they are connected through vertex pairs $[\mathbf{q}_p, \mathbf{q}_s]$. For each vertex in the virtual mesh, the nearest vertices in the physical mesh can be matched to construct vertex pair, in our work, three vertices are matched.

The virtual coupling of a surgical tool consists of two parts: the physical tool, which is directly controlled by the haptic device, and the virtual tool, which is used for graphic rendering and preventing from interpenetration with the soft body. Physical tool keeps the physical characteristic such as 6 DOF pose of haptic device, while virtual tool is designed for interaction and display in virtual simulation space.

With virtual coupling modeling, both the tool and the soft body are divided into two elastically coupled components, which facilitates the simultaneously implementation of haptic interaction and soft deformation.

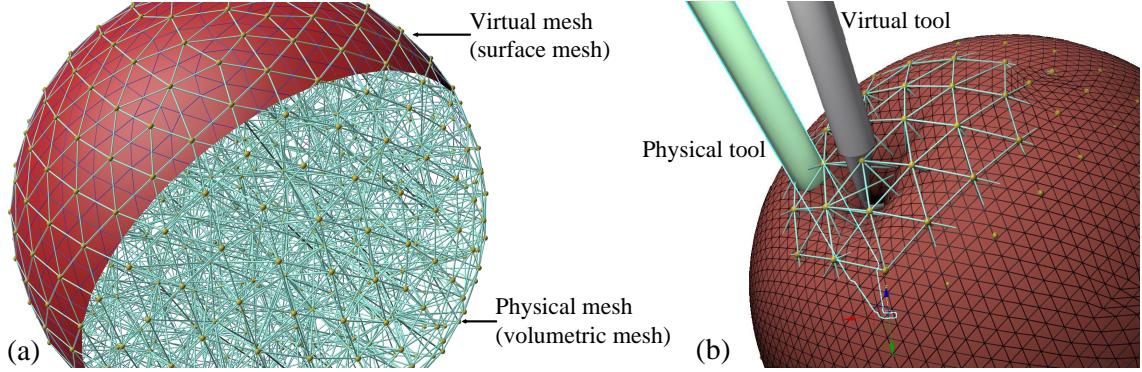


Figure 1: Virtual coupling system of soft-rigid interaction. (a) The virtual mesh is represented by a triangle mesh rendered in red, while the physical mesh is constructed using a tetrahedral mesh in cyan. (b) Interaction between virtual tool and virtual mesh. Physical tool is rendered in light green and virtual tool is rendered in gray.

3.2. Energy Definition of the System

Our system consists of four components: virtual mesh, physical mesh, virtual tool, and physical tool. Both the shape-preserving constraints and coupling constraints between them are defined by energy. Based on The Hooke's Law: $E = \frac{1}{2}k\Delta d^2$, in our system the the energy is defined in the Frobenius norm: $E = \frac{1}{2}k \|\mathbf{p}_1 - \mathbf{p}_2\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm of the matrix, k is the constraint stiffness set by user and \mathbf{p}_1 represents the array of elements involved in the constraint and \mathbf{p}_2 represents the array of target state of \mathbf{p}_1 which can minimize the constraint energy. For abbreviation and clarification, in the following part of this paper, variables in bold represents they are matrices including all involved vectors. The total energy of the system is defined as:

$$E = E_{tvc} + E_{con} + E_s + E_{svc} + E_k + E_{vol} \quad (1)$$

Each term in Equation 1 is defined as follows:

E_{tvc} is the virtual coupling energy between virtual tool and physical tool:

$$E_{tvc}(\mathbf{X}_g) = \frac{k_{vcs}}{2} \|\mathbf{P}_g - \mathbf{P}_h\|_F^2 + \frac{k_{vct}}{2} \theta(\omega_g, \omega_h)^2 \quad (2)$$

Here, $\mathbf{X} = [\mathbf{P}, \omega]$ is the 6-DOF position of tool. \mathbf{P}_g and \mathbf{P}_h are the 3-DOF position of virtual tool and physical tool respectively. ω_g and ω_h are the axis-angle rotation [39] of virtual tool and physical tool respectively. $\theta(\cdot, \cdot)$ is the rotation difference between virtual tool and physical tool. k_{vcs} and k_{vct} are the stiffness of shift energy

and torsion energy respectively. In the following part of this paper, variables in bold represents they are matrices including all involved vectors.

The contact energy E_{con} is defined by the projection distance of the collided vertices. The contact energy on virtual mesh and virtual tool is defined as:

$$E_{con}(\mathbf{q}, \mathbf{X}_g) = \frac{k_{con}}{2} \|\mathbf{q} - \mathbf{q}_{proj}(\mathbf{X}_g, \mathbf{q})\|_F^2 \quad (3)$$

where, k_{con} is the stiffness of the collision constraint. $\mathbf{q}_{proj}(\mathbf{X}_g, \mathbf{q})$ is the projected position of collided vertices, which is obtained by the simulation of corresponding surgical operation. When calculating constraint energy on virtual mesh, \mathbf{X}_g is constant and \mathbf{q} is variable. While calculating constraint energy on virtual tool, \mathbf{q} is constant and \mathbf{X}_g is variable.

The distance energy E_s is used to maintain the shape of the virtual mesh and it is defined on the two set of vertices \mathbf{q}_1 and \mathbf{q}_2 of surface edge:

$$E_s(\mathbf{q}_1, \mathbf{q}_2) = \frac{k_s}{2} \|\mathbf{q}_1 - \mathbf{q}_2 - \mathbf{R}(\mathbf{q}_1^0 - \mathbf{q}_2^0)\|_F^2 \quad (4)$$

where \mathbf{q}_1^0 and \mathbf{q}_2^0 are the initial positions. \mathbf{R} is the rotation matrix which rotate the original edge orientation to the current edge orientation, and it can be generated using the rotational axis and the angle between two directions.

The virtual coupling energy E_{svc} is defined by distance of the pair of physical soft body vertices \mathbf{q}_p and virtual soft body vertices \mathbf{q}_s . The virtual coupling energy of physical and virtual mesh is defined as:

$$E_{svc}(\mathbf{q}_p, \mathbf{q}_s) = \frac{k_{svc}}{2} \|\mathbf{q}_p - \mathbf{q}_s\|_F^2 \quad (5)$$

When calculating constraint energy on virtual mesh, \mathbf{q}_p is constant and \mathbf{q}_s is variable. While calculating constraint energy on physical mesh, \mathbf{q}_s is constant and \mathbf{q}_p is variable.

E_k is the kinetic energy of the soft body vertex, which is defined based on the classical kinetic energy $E = 1/2mv^2$:

$$E_k(\mathbf{q}) = \frac{1}{2} \left\| \mathbf{M}^{\frac{1}{2}} \frac{(\mathbf{q} - \mathbf{s})}{h} \right\|_F^2 \quad (6)$$

Here, \mathbf{q} is the current vertices positions, \mathbf{s} is the vertices positions predicted using Newton's first law, \mathbf{M} is the vertex mass matrix and h is the time step.

E_{vol} is the volumetric elasticity potential energy, which is defined to measure the deformation of soft body. We adopt as-rigid-as-possible constraint proposed by[40][15] as the constraint energy of tetrahedra deformation:

$$E_{vol}(\mathbf{F}) = k_{vol} \|\mathbf{F} - \mathbf{R}\|_F^2 \quad (7)$$

where k_{vol} is the stiffness of shape constraint, $\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}$ measures the difference between the current state of the tetrahedron and the original state of the tetrahedron. $\mathbf{D}_s \in \mathbb{R}^{3 \times 3}$ represents the shape and rotation of a tetrahedron in current state. The matrix is composed by three columns, each column represent the vector pointing from the tetrahedron's origin point to other vertices. $\mathbf{D}_m \in \mathbb{R}^{3 \times 3}$ is the shape and rotation matrix of a tetrahedron in initial state. \mathbf{R} is the rotational matrix calculated by polar decomposition of $\mathbf{F} = \mathbf{RS}$, while \mathbf{S} represents the deformation of a tetrahedron.

4. Asynchronous Simulation

The haptic interaction between the soft body and rigid tool is the minimization of the system's energy. In our paper, we use iterative optimization to update the vertexes of soft body and 6 DOF position of virtual tools respectively to satisfy all constraints in the system.

4.1. 6-DOF Position Optimization of Virtual Tool

The total energy of tool is $E_{tool} = E_{tvc} + E_{con}$. Updating virtual tool position is to minimize the total energy E_{tool} , so that the virtual tool is kept on the surface of soft body. To achieve energy minimization, we build the partial derivative equation around the position of current virtual tool:

$$\frac{\partial E_{tool}}{\partial \mathbf{X}_g} = \left[\frac{\partial E_{tool}}{\partial \mathbf{P}_g}, \frac{\partial E_{tool}}{\partial \omega_g} \right] = \mathbf{0} \quad (8)$$

where $\partial E_{tool}/\partial \mathbf{X}_g \in \mathbb{R}^{1 \times 6}$ is the partial derivative of E_{tool} with respect to the virtual tool pose \mathbf{X}_g . When the first order partial derivative of energy becomes zero, the energy is local minimum.

After that, the 6-DOF position update $\Delta \mathbf{X}_g$ can be derived from the Newton's method:

$$\frac{\partial E_{tool}}{\partial \mathbf{X}_g} + \frac{\partial^2 E_{tool}}{\partial \mathbf{X}_g^2} \Delta \mathbf{X}_g = \mathbf{0} \quad (9)$$

In practice, $\Delta \mathbf{P}_g$ and $\Delta \omega$ of $\Delta \mathbf{X}_g$ is limited to a specific range to achieve the system stability and continuity of virtual tool poses between adjacent time steps.

Force produced by energy-based constraint can be defined as the partial derivative of energy with respect to position with opposite sign, while torque can be defined as the partial derivative of energy with respect to attitude with opposite sign. So virtual

coupling force $\mathbf{F}_{vc} = -\partial E_{tvc}/\partial \mathbf{P}_g$ and virtual coupling torque $\mathbf{T}_{vc} = -\partial E_{tvc}/\partial \omega_g$ can be derived from energy partial derivative. In quasi-static assumption, virtual coupling force and torque on virtual tool are balance with the contact force and torque. We output the contact force the soft body applied on the surgical tool, so we adopt the virtual coupling force/torque with opposite direction as the force/torque output to the haptic device, having $\mathbf{F}_{output} = \partial E_{tvc}/\partial \mathbf{P}_g$ and $\mathbf{T}_{output} = \partial E_{tvc}/\partial \omega_g$;

4.2. Soft Dynamics using Local-global Iteration

The energy of virtual mesh is: $E_{soft} = E_{con} + E_s + E_{svc} + E_k$, and the energy of physical mesh is: $E_{soft} = E_{con} + E_{svc} + E_k + E_{vol}$. Both of the two energy terms contain quadratic distance measures, the mathematical expression of energy cannot be differentiated. Thus, we use the projective dynamics [4] to minimize the total energy of soft body. Projective dynamics update vertex positions using Jacobi iteration. The Jacobi iteration consists of 2 steps: the new vertex position with minimized local constraint \mathbf{p} is computed for each single constraint in **local step**, the vertex position with minimized total constraint energy is calculated in **global step**.

4.3. Asynchronous Iterations

Through the virtual coupling system of soft objects and rigid tools, the minimization of the total system energy is achieved asynchronously via two cycles with distinct frequencies. More specifically, soft deformation of physical mesh is simulated at a relatively low frequency ($60\text{Hz} \sim 100\text{Hz}$), whereas haptic interactions are computed at a high frequency ($\geq 1000\text{Hz}$). Distance constraints from vertex pairs are employed to maintain the deformation consistency between physical mesh and virtual mesh. Since we adopt an energy-based constraint where energy is only related to distance, our asynchronous iteration method can ensure the conservation of system energy.

The physical mesh deformation is detailed in Algorithm 1. At the beginning of each cycle, the vertex pairs are synchronize with the virtual mesh deformation. Afterward, the vertex position is predicted by the Newton's first law. $ConstraintProjection(\cdot)$ refers to the procedure of solving one constraint to obtain a locally optimized solution \mathbf{p} . $GlobalSynthesis(\cdot)$ synthesizes the local solutions associated with a vertex to yield a global optimal solution that minimizes the total constraint energy at that vertex. Finally, the shape constraint energy, kinetic energy and virtual coupling energy are minimized by the local-global iterations to update the vertex of volumetric mesh. The physical mesh deformation can be fully accelerated by GPU.

The virtual mesh deformation and haptic interaction is detailed in Algorithm 2. Similar to physical mesh deformation, the core of this process lies in local-global

Algorithm 1: Physical Mesh Deformation

```
1  $\mathbf{q}^{n,0} \leftarrow \mathbf{q}^n + h_d \mathbf{v}^n + h_d^2 \mathbf{M}^{-1} \mathbf{f}_{ext}^n$ 
2  $\mathbf{C}_k \leftarrow UpdateKineticConstraint(\mathbf{q}^{n,0})$ 
3  $\mathbf{C}_{svc} \leftarrow UpdateVertexofVirtualMesh()$ 
4  $\mathbf{C} \leftarrow \{\mathbf{C}_{svc}, \mathbf{C}_{vol}, \mathbf{C}_k\}$ 
5 for  $i \leftarrow 0$  to  $iterNum$  do
6   foreach  $\mathbf{C}_j \in \mathbf{C}$  do
7      $\mathbf{p}_j \leftarrow ConstraintProjection(\mathbf{q}^{n,i}, \mathbf{C}_j)$ 
8   end
9    $\mathbf{q}^{n,i+1} \leftarrow GlobalSynthesis(\mathbf{q}^{n,i}, \mathbf{p}_0, \mathbf{p}_1, \dots)$ 
10 end
11  $\mathbf{q}^{n+1} \leftarrow \mathbf{q}^{n,iterNum}$     $\mathbf{v}^{n+1} \leftarrow (\mathbf{q}^{n+1} - \mathbf{q}^n) / h_d$ 
```

iterations. In each iteration, we minimize energy through variable control: first, the tool is assumed to be static, and collision detection is used to update contact constraints for virtual mesh deformation; second, the virtual mesh is assumed to be static, and another collision detection is employed to update contact constraints for optimizing the tool’s position. Following the local-global iterations, the feedback force for the haptic device is calculated. The physical mesh deformation can be fully accelerated by GPU, whereas the update of the tool’s position is computed by the CPU.

5. Surgical Operation Simulation

Pressing and pulling of soft tissues are among the most commonly performed surgical operations. Our method leverages contact constraints to achieve soft tissue deformation, haptic rendering, and non-penetration between surgical tools and soft tissues. In this section, we will elaborate on the construction of contact constraints \mathbf{C}_{con} for pressing and pulling operations. Specifically, we calculate the projected position \mathbf{q}_{proj} of each vertex and determine the non-penetration position \mathbf{X}_{proj} of the tools through collision detection.

5.1. Soft Tissue Pressing

During the pressing operation, the contact points between the tool and the soft body change dynamically as the tool moves. We use the parameterized collider or signed distance field for collision detection. Given that the simulation frequency for haptic rendering and virtual mesh deformation reaches 1000Hz, the time step

Algorithm 2: Haptic Interaction

```
1  $\mathbf{C}_{svc} \leftarrow UpdateVertexofPhcialMesh()$ 
2  $\mathbf{C} \leftarrow \{\mathbf{C}_{svc}, \mathbf{C}_s\}$ 
3 foreach tools do
4   |  $\mathbf{X}_h \leftarrow UpdatePhysicalToolfromHapticDevice()$ 
5 end
6  $\mathbf{q}^{n,0} \leftarrow \mathbf{q}^n + h_d \mathbf{v}^n + h_d^2 \mathbf{M}^{-1} \mathbf{f}_{ext}^n$ 
7  $\mathbf{X}_g^{n,0} \leftarrow \mathbf{X}_g^n$ 
8 for  $i \leftarrow 0$  to iterNum do
9   | foreach tools do
10    |   |  $\mathbf{C}_{con} \leftarrow CollisionDetection(\mathbf{X}_g^{n,i}, \mathbf{q}^{n,i})$ 
11    |   |  $\mathbf{C}_{total} \leftarrow \mathbf{C} \cup \mathbf{C}_{scon}$ 
12    | end
13    | foreach  $\mathbf{C}_j \in \mathbf{C}_{total}$  do
14    |   |  $\mathbf{p}_j \leftarrow ConstraintProjection(\mathbf{q}^{n,i}, \mathbf{C}_j)$ 
15    | end
16    |  $\mathbf{q}^{n,i+1} \leftarrow GlobalSynthesis(\mathbf{q}^{n,i}, \mathbf{p}_0, \mathbf{p}_1, \dots)$ 
17    | foreach tools do
18    |   |  $\mathbf{C}_{con} \leftarrow CollisionDetection(\mathbf{X}_g^{n,i}, \mathbf{q}^n)$ 
19    |   |  $\mathbf{X}_g^{n,i+1} \leftarrow UpdateVirtualTool(\mathbf{X}_g^{n,i}, \mathbf{X}_h, \mathbf{C}_{con}, \mathbf{C}_{tvc})$ 
20    | end
21 end
22  $\mathbf{q}^{n+1} \leftarrow \mathbf{q}^{n,iterNum}$ 
23  $\mathbf{v}^{n+1} \leftarrow (\mathbf{q}^{n+1} - \mathbf{q}^n) / h_d$ 
24 foreach tools do
25   |  $\mathbf{X}_g^{n+1} \leftarrow \mathbf{X}_g^{n,iterNum}$ 
26   |  $\mathbf{F}_{output} \leftarrow \frac{\partial E_{tvc}}{\partial \mathbf{P}_g}$ 
27   |  $\mathbf{T}_{output} \leftarrow \frac{\partial E_{tvc}}{\partial \omega_g}$ 
28 end
```

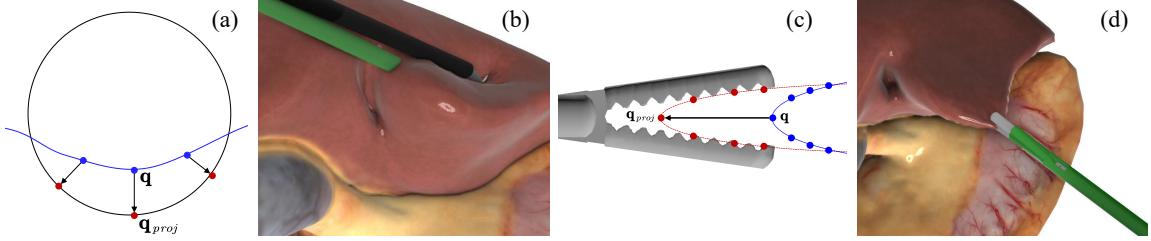


Figure 2: Soft tissue pressing and pulling. The blue dots are the collided vertices, while red dots are the projected position of vertices. The green tool rendered is physical tool. (a) Vertex projection of pressing, sphere is the cross section of surgical tool. (b) Result of pressing. (c) Vertex projection of pulling. (d) Result of grasping and pulling.

is sufficiently small to allow discrete collision detection to fully capture the collided vertices of each frame. Furthermore, continuous collision detection enables more effective capture of colliding vertices. Figure 2 (a) illustrates the cross-section of a cylinder-vertex collision, the vertices are projected onto the tool surface \mathbf{q}_{proj} along the projection direction pointing towards the surface. For simplicity, the projection direction of vertices can be set to the opposite of vertex normals. Figure 2 (b) shows a result of soft tissue pressing, the tool and soft tissue are capable of non-penetration with each other.

5.2. Soft Tissue Pulling

In our system, soft tissue pulling is performed after the grasping operation. As Figure 2 (c) shows, once the forceps are closed, the relative position between the vertex and the tool remains fixed throughout the pulling operation, and the projected vertices \mathbf{q}_{proj} are calculated by the translation-rotation matrix from virtual tool. Since the detection of grasped vertex occurs only once, collision detection for the pulling operation in each haptic frame is unnecessary.

Fig. 3 shows the the projection \mathbf{q}_{proj} of grasped vertices to simulate soft tissue flattening between forceps jaws. Collision detection between the forceps jaws and vertices is implemented using axis-aligned bounding boxes (AABBs). Collider of the forceps can be divided into two parts: hard collider representing the real size of forceps and soft collider representing the smoothing area of the forceps. The final grasp projection position \mathbf{q}_g of the grasped vertex is computed as a weighted sum of the corresponding vertex of the virtual mesh \mathbf{q}_s and the vertex projected onto the projection plane of the forceps jaw \mathbf{q}_f :

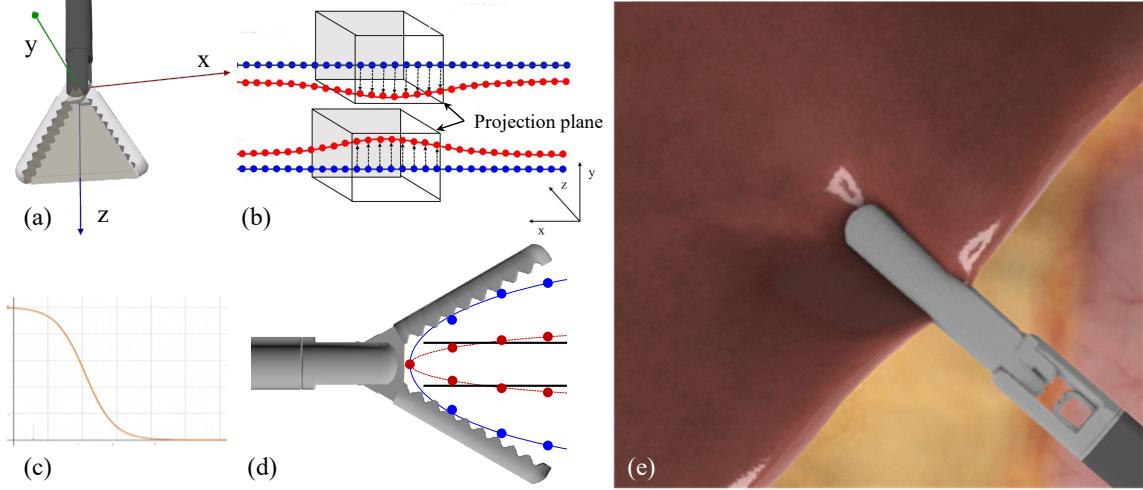


Figure 3: Soft Tissue Grasping. Dots in blue are the vertices before grasp, while dots in red are the position of vertices after grasping.(a) Local frame of the forceps. (b) (d) Vertex projection within the local frame of the forceps. (d) Mix weight function visualization. (e) Grasping effect on soft body.

$$q_g = w(d)q_f + (1 - w(d))q_s, \quad (10)$$

$$w(d) = \frac{1}{1 + \eta e^d}$$

Here, d represents the distance to the hard AABB collider. As illustrated in Figure 3 (d), $w(d)$ serves as the mixed weight between q_s and q_f , indicating that a vertex farther from the forceps jaw will be closer to the q_s . The parameter η is a smoothing factor used to visually control the elasticity of the soft tissues; larger values lead to a more rapid decline in the mixed weight function, which allows for the simulation of softer objects. Figure 2 (e) shows a result of grasping and pulling the soft tissue, the jaws of virtual tool can fit tightly to the soft tissue surface.

6. Results

6.1. Experimental Setup

Soft deformation and haptic rendering are implemented using C++ and CUDA, while graphical results are rendered by OpenGL. Our system utilizes two modified Geomagic Touch devices as haptic interfaces, which have been enhanced forceps with fitted encoders. The computer is powered by an AMD 7700X processor and an

NVIDIA RTX 3080ti graphic card. In cases where the number of physical units is large, an additional GPU can be employed to share the computational load of high-frequency virtual mesh deformation. We use another NVIDIA RTX 3070 graphic card to accelerate the virtual mesh deformation. To show the ability of simulating basic operations in surgery, we simulate two types of surgical tools: electrocoagulation hook for pressing soft body and forceps for grasping and pulling soft body. The collider of those tools are represented as cylinders and spheres. In the following sections, we present the simulation results and performance metrics.

6.2. Haptic Interaction with Soft Body in Narrow Space

The haptic interaction of inserting of into soft body is shown in Figure 4. Our method can provide smooth pose of virtual surgical tool during the interaction procedure and gives stable feedback force to the user. To achieve 1000Hz deformation, [2] employs a reduced model that cannot undergo large deformation. Compared with their method, our asynchronous virtual coupling scheme can achieve large-scale deformation of soft body by separating the large-scale deformation simulation from the high frequency haptic rendering. Wang et al. [3] utilize position-based constraints to achieve strict non-penetration between rigid bodies. Consequently, their method fails to enable insertion into gaps narrower than the tool, as the non-penetration constraint would in such cases cause the tool to remain on the object’s surface. In contrast to the position-based constraints, our energy-based constraints allow for the user movements when inserting into soft tissues during the tool position updating.

6.3. Surgical Operation Simulation

We have constructed two surgical scenarios: cholecystectomy and gynecological. We simulated two types of surgical instruments, namely the electrocoagulation hook and the grasper. The electrocoagulation hook is used to press soft tissues, and the grasper is used to pull soft tissues. The number of physical units is shown in Table 1. Figure 5 shows the haptic interaction of pressing and pulling. Our approach enables large-scale deformation of organs while, via the tool’s virtual coupling system, preventing the tool from penetrating into the organs. Our method can simulate the indentation at the tip of the electrocoagulation hook. Through the feedback force, users can feel the contact force between the tool and the soft body.

The stiffness of constraint can be set by user. The stiffness of soft body is measured using Poisson ratio and Young’ modulus, several previous works[41][42] is dedicated to test biochemical criterion of human organs. We can simply set shape constraint stiffness to Young’s modulus of soft body and get similar hardness of real soft body. The greater the constraint stiffness is, the easier the constraint be satisfied. The stiffness of other types of constraints can be set to a value that makes the

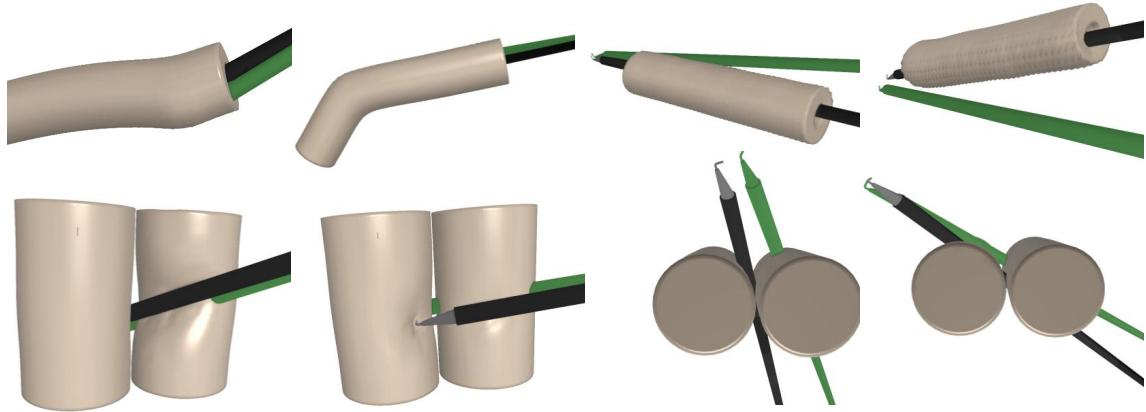


Figure 4: Deformation results of inserting stick shaped surgical tool into narrow space and tube. Virtual surgical tool rendered in black and the haptic tool is rendered in green. The virtual tool can smoothly insert into the soft body tube without trembling and jump over of position.

constraint energy of the same order of magnitude as the shape constraint energy. For the virtual tool, greater stiffness of contact constraint will make the contact depth smaller, which will prevent the virtual tool from mistakenly penetration, as shown in Figure 6(a) and (c). Greater virtual coupling constraint stiffness will make the virtual tool align to the physical tool easier, which will increase contact depth of soft body vertex and make the deformation of soft body easier, as shown in Figure 6(a) and (b). Comparing Figure 6(a) and (d), though both k_c and k_{vc} are set to different value, but k_c/k_{vc} are identical in these two configurations, so the deformation of soft body are similar. Set the shape stiffness according to Young's modulus and set other constraint stiffness following the stiffness ratio with shape constraint stiffness will make the simulation more realistic.

6.4. Performance

The force is recorded in haptic end with update rate over 1000Hz. The feedback force and torque on the x-axis, y-axis, and z-axis during the operation of pressing and pulling soft tissue in virtual surgery are figured in Figure 7. Magnitude of force and torque are smooth even in narrow spaces. In our experiments, computational stability of the virtual tool and force/torque smoothness are maintained during the surgery procedure. No abrupt jump or vibration of the virtual tool occurs between two adjacent haptic time steps.

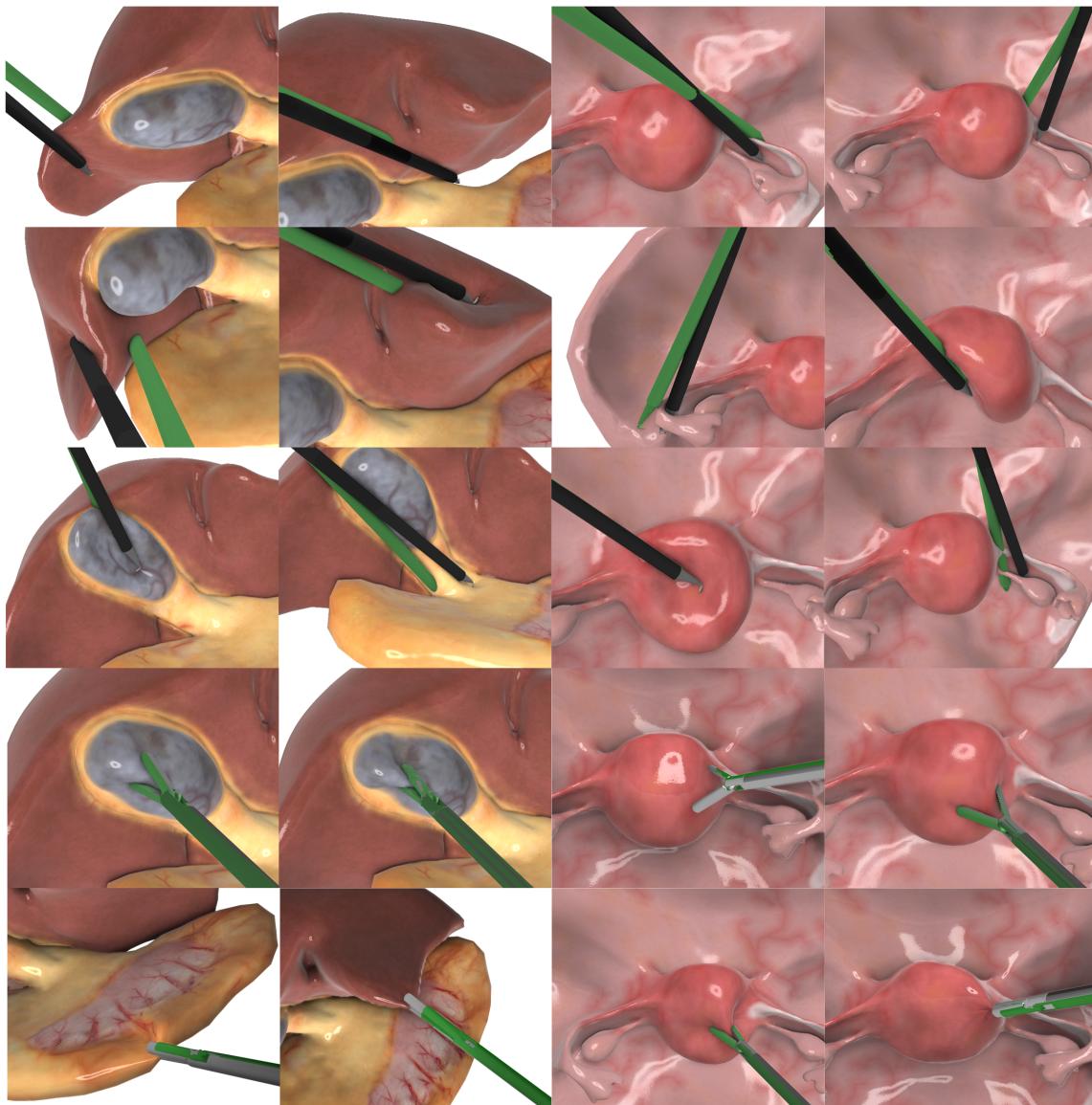


Figure 5: Left side shows the result of cholecystectomy surgery, and right side shows the result of gynecological surgery. The first three rows shows the soft body pressing and the last two rows shows the soft body pulling. Virtual tool is rendered by original color, while the haptic tool is rendered in green. The surgical tool is well kept on the surface of soft body, preventing from wrongly penetration.

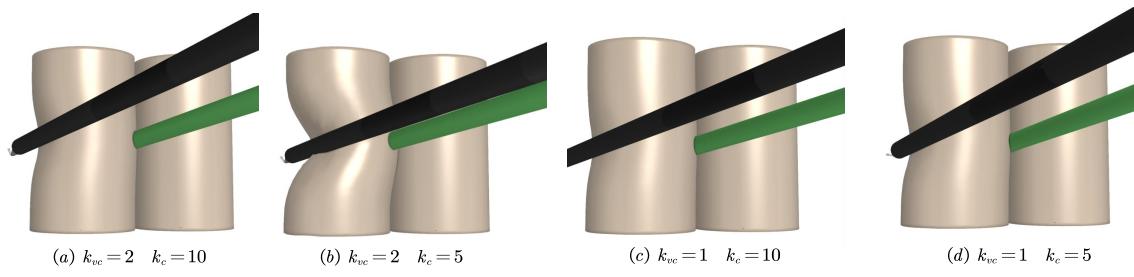


Figure 6: Haptic interaction with different stiffness of contact energy and virtual coupling energy.

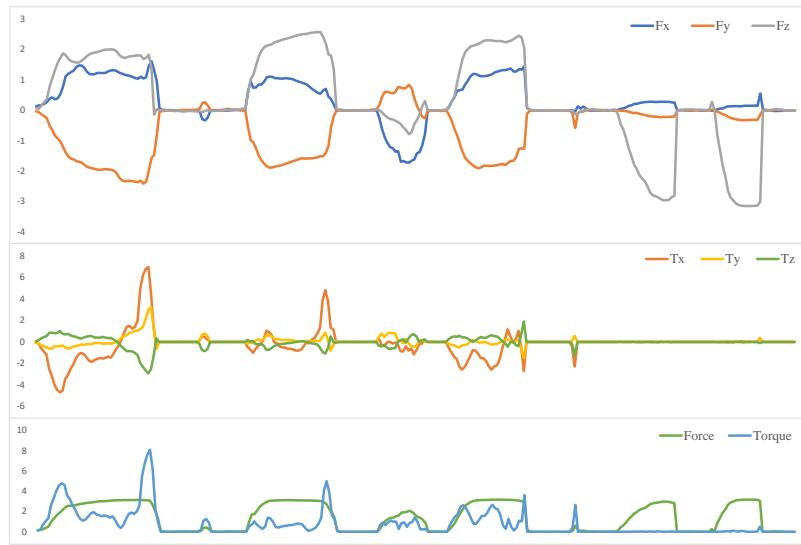


Figure 7: Force/Torque of pressing and pulling in cholecystectomy surgery simulation.

For the scene with the operation of pulling grasping and deformation, time consumption of operation is listed in Table 1. All the experiments are done ten times to get the average frame per second for each scene. The time per frame is the total time a frame takes, which is smaller than the sum of all stages because some stages are parallelized using multiple threads. The FPS include rendering time of each frame. Our method can accomplish real-time interaction in complex virtual surgery scenarios.

Data	cholecystectomy	gynaecology	tube	gap
Tetrahedra Num.	44178	16848	8068	16630
Triangle Num.	82748	29102	14588	30812
Sim FPS(pressing)	125.1	131.0	137.8	140.0
Sim FPS(grasping)	123.3	132.2	140.3	140.1
Haptic FPS	1003.2	1004.1	998.1	1007.5

Table 1: The number of physical units and simulation efficiency

6.5. User Study

All procedures were approved by the Biological and Medical Ethics Committee of Beihang University(BM20240098). We invited 20 surgeons from the department of pediatric surgery of the affiliated hospital of Qingdao University to experience our simulation system for cholecystectomy surgery and gynecological surgery for about 20 minutes. Each of the user will practice the pressing, grabbing operations in the simulation system. After the evaluation, they will complete a questionnaire to provide a feedback for the simulation system. The table includes the evaluation of the realism of the feedback forces during the grasping and pressing processes in the surgical procedure. The doctors participating in the experiment scored the feedback forces provided by the virtual surgical simulation system during the grasping and pressing processes based on their previous surgical experience and comparison with the current virtual surgical simulation system. 20 questionnaires are collected and analyzed to verify the effectiveness of our simulation system. The rating scale is set from 1 to 10, where 1 indicates extreme dissatisfaction with the effect and very poor authenticity, while 10 indicates extreme satisfaction with the effect and high authenticity. They have previously used virtual surgery systems that lacked haptic force feedback. During the experiment, most participants gave high scores to the surgical simulation system, indicating that participants generally found the feel of operating the instruments to be quite realistic, with a relatively low variability in their responses. The statistical data of our questionnaire We displayed in Table. 2.

Question	Evaluation Indicator	AVG	SD
1	Haptic Feedback of Pressing	8.8	0.59
2	Haptic Feedback of Grasping and Pulling	9	0.53
3	Overall Impression	8.93	0.62

Table 2: Statistical data of our questionnaire. Most of the participants gives a high score to the realistic of feedback force during operations such as pressing and pulling soft body.

7. Conclusion

This paper presents an energy-based haptic rendering method for real-time virtual surgery simulation. Both rigid surgical tools and soft tissues are modeled via an energy-based virtual coupling system, enabling the simulation of complex haptic operations, such as soft tissue pressing, soft tissue grasping and pulling, insertion of tools into soft tissue.

This work has two limitations: first, it cannot effectively simulate rigid body collisions; second, when the number of physical units is large, achieving high-frequency haptic rendering becomes challenging. In future work, we plan to adopt a Multi-Grid method to accelerate soft tissue deformation and the collision detection will be carried out on different scale, thereby ensuring the rapid convergence of interactive soft tissue deformation. Suturing, tissue cutting and other surgical operations will be further added to the simulation system in the future.

8. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62132021, Shandong Provincial Natural Science Foundation under Grant ZR2025MS1106, Key R&D Program (Major Science and Technology Innovation Project) of Shandong Province under Grant 2025CXGC010604, Ningbo Science and Technology Innovation Project under Grant 2024Z301, Guangxi Science and Technology Major Program under Grant GuiKeAA24206017, Science and Technology Program of the Joint Fund of Scientific Research for the Public Hospitals of Inner Mongolia Academy of Medical Sciences under Grant 2023GLLH0334, Capital’s Funds for Health Improvement and Research under Grant CFH 2024-2-40611, Beijing Nova Program under Grant 20250484786, and the Fundamental Research Funds for the Central Universities Funds for the Central Universities.

References

- [1] M. Ortega, S. Redon, S. Coquillart, A six degree-of-freedom god-object method for haptic display of rigid bodies, in: IEEE Virtual Reality Conference (VR 2006), 2006, pp. 191–198.
- [2] J. Barbic, D. L. James, Six-dof haptic rendering of contact between geometrically complex reduced deformable models, *IEEE Transactions on Haptics* 1 (1) (2008) 39–52.
- [3] D. Wang, Y. Shi, S. Liu, Y. Zhang, J. Xiao, Haptic simulation of organ deformation and hybrid contacts in dental operations, *IEEE Transactions on Haptics* 7 (1) (2014) 48–60.
- [4] S. Bouaziz, S. Martin, T. Liu, L. Kavan, M. Pauly, Projective dynamics: Fusing constraint projections for fast simulation, *Seminal Graphics Papers: Pushing the Boundaries*, Volume 2 (2023).
- [5] R. Akhtar, M. J. Sherratt, J. K. Cruickshank, B. Derby, Characterizing the elastic properties of tissues, *Materials Today* 14 (3) (2011) 96–105.
- [6] O. Al-Mai, M. Ahmadi, J. Albert, A compliant 3-axis fiber-optic force sensor for biomechanical measurement, *IEEE Sensors Journal* 17 (20) (2017) 6549–6557.
- [7] J. Brown, S. Sorkin, C. Bruyns, J.-C. Latombe, K. Montgomery, M. Stephanides, Real-time simulation of deformable objects: Tools and application, in: Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No. 01TH8596), IEEE, 2001, pp. 228–258.
- [8] S. Suzuki, N. Suzuki, A. Hattori, A. Uchiyama, S. Kobayashi, Sphere-filled organ model for virtual surgery system, *IEEE transactions on medical imaging* 23 (6) (2004) 714–722.
- [9] A. G. Cope, J. J. Lazaro-Weiss, B. E. Willborg, E. D. Lindstrom, K. C. Mara, C. C. Destephano, M. H. Vetter, G. E. Glaser, C. L. Langstraat, A. H. Chen, M. A. Martino, T. A. Dinh, R. Salani, I. C. Green, Surgical science–symbionix robotic hysterectomy simulator: Validating a new tool, *Journal of Minimally Invasive Gynecology* 29 (6) (2022) 759–766.
- [10] J. Pan, J. Bai, X. Zhao, A. Hao, H. Qin, Dissection of hybrid soft tissue models using position-based dynamics, in: Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology, 2014, pp. 219–220.

- [11] A. Margalit, K. V. Suresh, M. Marrache, J. M. Lentz, R. Lee, J. Tis, R. Varghese, B. Hayashi, A. Jain, D. Laporte, Evaluation of a slipped capital femoral epiphysis virtual reality surgical simulation for the orthopaedic trainee, *JAAOS Global Research & Reviews* 6 (4) (2022) e22.
- [12] L. Frizziero, G. Trisolino, G. M. Santi, G. Alessandri, S. Agazzani, A. Liverani, G. C. Menozzi, G. L. Di Gennaro, G. M. G. Farella, A. Abbruzzese, et al., Computer-aided surgical simulation through digital dynamic 3d skeletal segments for correcting torsional deformities of the lower limbs in children with cerebral palsy, *Applied Sciences* 12 (15) (2022) 7918.
- [13] D. M. Costello, I. Huntington, G. Burke, B. Farrugia, A. J. O'Connor, A. J. Costello, B. C. Thomas, P. Dundee, A. Ghazi, N. Corcoran, A review of simulation training and new 3d computer-generated synthetic organs for robotic surgery education, *Journal of Robotic Surgery* (2022) 1–15.
- [14] T. Liu, A. W. Bargteil, J. F. O'Brien, L. Kavan, Fast simulation of mass-spring systems, *ACM Transactions on Graphics (TOG)* 32 (6) (2013) 1–7.
- [15] M. Müller, B. Heidelberger, M. Teschner, M. Gross, Meshless deformations based on shape matching, *ACM Trans. Graph.* 24 (3) (2005) 471–478.
- [16] E. Sifakis, J. Barbic, Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction, in: International Conference on Computer Graphics and Interactive Techniques, 2012.
- [17] I. Peterlik, M. Nouicer, C. Duriez, S. Cotin, A. Kheddar, Constraint-based haptic rendering of multirate compliant mechanisms, *IEEE Transactions on Haptics* 4 (3) (2011) 175–187.
- [18] Z. Li, Haptic dissection of deformable objects using extended finite element method, Ph.D. thesis, Université d'Ottawa/University of Ottawa (2014).
- [19] M. Müller, B. Heidelberger, M. Hennix, J. Ratcliff, Position based dynamics, *Journal of Visual Communication and Image Representation* 18 (2) (2007) 109–118.
- [20] M. Macklin, M. Müller, N. Chentanez, Xpbd: position-based simulation of compliant constrained dynamics, in: Proceedings of the 9th International Conference on Motion in Games, 2016, pp. 49–54.

- [21] M. Müller, Hierarchical Position Based Dynamics, in: F. Faure, M. Teschner (Eds.), Workshop in Virtual Reality Interactions and Physical Simulation "VRI-PHYS" (2008), The Eurographics Association, 2008.
- [22] M. Fratarcangeli, V. Tibaldo, F. Pellacini, et al., Vivace: a practical gauss-seidel method for stable soft body dynamics., ACM Trans. Graph. 35 (6) (2016) 214–1.
- [23] A. H. Chen, Z. Liu, Y. Yang, C. Yuksel, Vertex block descent, arXiv preprint arXiv:2403.06321 (2024).
- [24] H. Wang, A chebyshev semi-iterative approach for accelerating projective and position-based dynamics, ACM Transactions on Graphics (TOG) 34 (6) (2015) 1–9.
- [25] L. Lan, G. Ma, Y. Yang, C. Zheng, M. Li, C. Jiang, Penetration-free projective dynamics on the gpu, ACM Trans. Graph. 41 (4) (jul 2022).
- [26] T. H. Massie, J. K. Salisbury, et al., The phantom haptic interface: A device for probing virtual objects, in: Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems, Vol. 55, Chicago, IL, 1994, pp. 295–300.
- [27] M. Ortega, S. Redon, S. Coquillart, A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties, IEEE transactions on visualization and computer graphics 13 (3) (2007) 458–469.
- [28] D. E. Johnson, P. Willemsen, E. Cohen, Six degree-of-freedom haptic rendering using spatialized normal cone search, IEEE Transactions on Visualization and Computer Graphics 11 (6) (2005) 661–670.
- [29] H. Qin, A. Song, Y. Liu, G. Jiang, B. Zhou, Design and calibration of a new 6 dof haptic device, Sensors 15 (12) (2015) 31293–31313.
- [30] H. Xu, J. Barbic, 6-dof haptic rendering using continuous collision detection between points and signed distance fields, IEEE transactions on haptics 10 (2) (2016) 151–161.
- [31] L. Jin, X. Duan, C. Li, Q. Shi, H. Wen, J. Wang, H. Li, Design of a novel parallel mechanism for haptic device, Journal of Mechanisms and Robotics 13 (4) (2021) 045001.

- [32] V. K. Pediredla, K. Chandrasekaran, S. Annamraju, A. Thondiyath, Design, analysis, and control of a 3-dof novel haptic device displaying stiffness, texture, shape, and shear, *IEEE Access* 9 (2021) 72055–72065.
- [33] J. Dearden, C. Grames, B. D. Jensen, S. P. Magleby, L. L. Howell, Inverted l-arm gripper compliant mechanism, *Journal of Medical Devices* 11 (3) (2017) 034502.
- [34] M. A. Otaduy, M. C. Lin, A modular haptic rendering algorithm for stable and transparent 6-dof manipulation, *IEEE Transactions on Robotics* 22 (4) (2006) 751–762.
- [35] W. A. McNeely, K. D. Puterbaugh, J. J. Troy, Six degree-of-freedom haptic rendering using voxel sampling, *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999).
- [36] W. A. McNeely, K. D. Puterbaugh, J. J. Troy, B. Phantom, Voxel-based 6-dof haptic rendering improvements, 2006.
- [37] M. Wan, W. McNeely, Quasi-static approach approximation for 6 degrees-of-freedom haptic rendering, in: *IEEE Visualization, 2003. VIS 2003.*, 2003, pp. 257–262.
- [38] M. C. Lin, M. Otaduy, *Haptic rendering: foundations, algorithms, and applications*, CRC Press, 2008.
- [39] M. Wan, W. A. McNeely, Quasi-static approach approximation for 6 degrees-of-freedom haptic rendering, *IEEE Visualization, 2003. VIS 2003.* (2003) 257–262.
- [40] O. Sorkine-Hornung, M. Alexa, As-rigid-as-possible surface modeling, in: *Eurographics Symposium on Geometry Processing*, 2007.
- [41] S. Franchi-Abella, L. Corno, E. Gonzales, G. Antoni, M. Fabre, B. Ducot, D. Pariente, J.-L. Gennisson, M. Tanter, J.-M. Corréas, Feasibility and diagnostic accuracy of supersonic shear-wave elastography for the assessment of liver stiffness and liver fibrosis in children: a pilot study of 96 patients, *Radiology* 278 (2) (2016) 554–562.
- [42] P. Galina, E. Alexopoulou, A. Zellos, V. Grigoraki, T. Siananidou, N. L. Kelekis, M. Zarifi, Performance of two-dimensional ultrasound shear wave elastography: reference values of normal liver stiffness in children, *Pediatric Radiology* 49 (1) (2019) 91–98.

- [43] J. Barbic, Real-time reduced large-deformation models and distributed contact for computer graphics and haptics, Carnegie Mellon University, 2007.

Appendix A. Solving partial derivative equation on Virtual Tool

The pose of virtual surgical tool $\mathbf{X}_g = [\mathbf{P}_g, \omega_g]$ can be dealt as concatenating 3 DOF position of rotational center and 3 DOF rotation. In the following part, we will describe the position of surgical tool's rotational center as the position of surgical tool for abbreviation. The position of surgical tool is independent from the rotation of surgical tool, as the position is defined as the position of rotational center, rotation the surgical tool will not influence the position of rotational center of surgical tool.

Solving following partial derivative equation on virtual tool will get the pose update $\Delta\mathbf{X}_g$:

$$\frac{\partial E_{tool}}{\partial \mathbf{X}_g} + \frac{\partial^2 E_{tool}}{\partial \mathbf{X}_g^2} \Delta \mathbf{X}_g = \mathbf{0} \quad (\text{A.1})$$

In this section we will describe how to calculate the first order partial derivative and second order derivative in eq. (A.1) with more details.

Appendix A.1. First-order Partial Derivative of Tool Energy

Our energy-based constraints are built following Hooke's law and the derivative of virtual coupling energy have well-defined physical meanings.

As the translation \mathbf{P}_g is independent to rotation ω_g , calculating the partial derivatives of energy can be dealt with respect to position and omega separately. The first order derivative of constraint energy on virtual tool in eq. (A.1) is:

$$\frac{\partial E_{tool}}{\partial \mathbf{X}_g} = \left[\frac{\partial E_{tool}}{\partial \mathbf{P}_g}, \frac{\partial E_{tool}}{\partial \omega_g} \right] = \left[\frac{\partial E_{tvc}}{\partial \mathbf{P}_g} + \frac{\partial E_{con}}{\partial \mathbf{P}_g}, \frac{\partial E_{tvc}}{\partial \omega_g} + \frac{\partial E_{con}}{\partial \omega_g} \right] \quad (\text{A.2})$$

The derivative of virtual coupling energy on virtual tool position can be dealt as the force applied on the virtual tool. The first-order partial derivative of virtual coupling energy with respect to the virtual tool position as follows:

$$\frac{\partial E_{tvc}}{\partial \mathbf{P}_g} = k_{vcs}(\mathbf{P}_g - \mathbf{P}_h) \quad (\text{A.3})$$

where k_{vcs} is the stiffness of virtual coupling force specified by user, \mathbf{P}_h is the position of physical tool and \mathbf{P}_g is the position of virtual tool. The position of the tool is defined as the rotational center of tool.

The first-order partial derivative of virtual coupling energy with respect to the virtual tool rotation can be dealt as virtual coupling torque, having:

$$\frac{\partial E_{tvc}}{\partial \omega_g} = -k_{vct}vec(Quat(\omega_h) \cdot Quat(\omega_g)^{-1}) \quad (\text{A.4})$$

where $Quat(\cdot)$ is a function which can convert attitude in axis-angle to quaternion and $vec(\cdot)$ is a function returns the virtual part of quaternion. k_{vct} is the stiffness of virtual coupling torque set by user.

The first-order partial derivative of contact constraint energy with respect to the position of virtual tool is:

$$\frac{\partial E_{con}}{\partial \mathbf{P}_g} = - \sum_{i=1}^n k_c d(\mathbf{q}_i, \mathbf{P}_g) \mathbf{N}_i \quad (\text{A.5})$$

in this formula, n is the number of soft body particles in collision, $d(\cdot, \cdot)$ is a function returns the insertion depth of collided soft body particles and \mathbf{N}_i is the non-penetration direction of the i th collided vertex. For vertex on the surface of soft body, the non-penetration direction is set to the opposite direction of surface normal. k_c is the stiffness of collision constraint.

The first-order partial derivative of contact constraint energy with respect to the rotation of virtual tool is calculated as follows:

$$\frac{\partial E_{con}}{\partial \omega} = \sum_{i=1}^n \tilde{\mathbf{r}}^i \left(\frac{\partial E_{con}^i}{\partial \mathbf{P}_g} \right)^T \quad (\text{A.6})$$

where $\mathbf{r}^i = \mathbf{q}^i - \mathbf{P}_g$ is a vector from the rotation center of virtual tool to the collided vertex \mathbf{q}^i , E_{con}^i is the contact energy on the i th vertex. $\tilde{\cdot}$ is an operator of skew symmetric matrix. The construction of skew symmetric matrix of a 3 dimensional vector \mathbf{r} is:

$$\tilde{\mathbf{r}} = \begin{bmatrix} 0 & -\mathbf{r}_2 & \mathbf{r}_1 \\ \mathbf{r}_2 & 0 & -\mathbf{r}_0 \\ -\mathbf{r}_1 & \mathbf{r}_0 & 0 \end{bmatrix} \quad (\text{A.7})$$

Appendix A.2. Second-order Partial Derivative of Tool Energy

To get the minimized constraint energy on virtual tool, we can build partial derivative equation on virtual tool, as described in eq. (A.1). To solve this equation, we adopt Newton's method to get the update of virtual tool pose. This need the second-order partial derivative of tool energy. With first-order partial derivatives of constraint energy, we can further get second-order partial derivatives of constraint energy with following equation:

$$\frac{\partial^2 E_{tool}}{\partial \mathbf{X}_g^2} = \frac{\partial}{\partial \mathbf{X}_g} \left(\frac{\partial E_{tool}}{\partial \mathbf{X}_g} \right) = \begin{bmatrix} \frac{\partial^2 E_{tvc}}{\partial \mathbf{P}_g^2} + \frac{\partial^2 E_{con}}{\partial \mathbf{P}_g^2} & \frac{\partial^2 E_{tvc}}{\partial \mathbf{P}_g \omega_g} + \frac{\partial^2 E_{con}}{\partial \mathbf{P}_g \omega_g} \\ \frac{\partial^2 E_{tvc}}{\partial \omega_g \mathbf{P}_g} + \frac{\partial^2 E_{con}}{\partial \omega_g \mathbf{P}_g} & \frac{\partial^2 E_{tvc}}{\partial \omega_g^2} + \frac{\partial^2 E_{con}}{\partial \omega_g^2} \end{bmatrix} \quad (\text{A.8})$$

First-order partial derivative of energy on virtual tool with respect to 6 DOF virtual tool pose is a 6 dimensional vector. So the second-order partial derivative of energy on virtual tool with respect to 6 DOF pose is a 6×6 matrix. The calculating method of each second-order partial derivatives of contact energy are listed as follows.

Partial squared of E_{con} over partial \mathbf{P}_g squared is divided using eq. (A.5), having:

$$\frac{\partial^2 E_{con}}{\partial \mathbf{P}_g^2} = \frac{\partial}{\partial \mathbf{P}_g} \left(\frac{\partial E_{con}}{\partial \mathbf{P}_g} \right) = k_c \mathbf{N} \mathbf{N}^T \quad (\text{A.9})$$

Partial squared of E_{con} over partial \mathbf{P}_g partial ω_g is derived using eq. (A.5):

$$\frac{\partial^2 E_{con}}{\partial \mathbf{P}_g \partial \omega_g} = \frac{\partial}{\partial \omega_g} \left(\frac{\partial E_{con}}{\partial \mathbf{P}_g} \right) = -k_c \mathbf{N} \mathbf{N}^T \tilde{\mathbf{r}} - k_c d(\mathbf{q}_i, \mathbf{P}_g) \tilde{\mathbf{N}} \quad (\text{A.10})$$

Partial squared of E_{con} over partial ω_g partial \mathbf{P}_g is derived using eq. (A.6):

$$\frac{\partial^2 E_{con}}{\partial \omega_g \partial \mathbf{P}_g} = \frac{\partial}{\partial \mathbf{P}_g} \left(\frac{\partial E_{con}}{\partial \omega_g} \right) = k_c \tilde{\mathbf{r}} \mathbf{N} \mathbf{N}^T - \tilde{\mathbf{F}}_c \quad (\text{A.11})$$

where $\tilde{\mathbf{F}}_c$ is identical to $-\partial E_{con}/\partial \mathbf{P}_g$.

Partial squared of E_{con} over partial ω_g partial \mathbf{P}_g following [43] is derived using eq. (A.6):

$$\frac{\partial^2 E_{con}}{\partial \omega_g^2} = \frac{\partial}{\partial \omega_g} \left(\frac{\partial E_{con}}{\partial \omega_g} \right) = \tilde{\mathbf{F}}_c \tilde{\mathbf{r}} - k \tilde{\mathbf{r}} \mathbf{N} \mathbf{N}^T \tilde{\mathbf{r}} - k_c d(\mathbf{q}_i, \mathbf{P}_g) \tilde{\mathbf{r}} \tilde{\mathbf{N}} \quad (\text{A.12})$$

The second-order partial derivatives of virtual coupling energy on virtual surgical tool are derived from the first order partial derivatives of E_{tvc} with respect to position and rotation. Partial squared E_{tvc} over partial \mathbf{P}_g squared is derived from eq. (A.3), having:

$$\frac{\partial^2 E_{tvc}}{\partial \mathbf{P}_g^2} = \frac{\partial}{\partial \mathbf{P}_g} \left(\frac{\partial E_{tvc}}{\partial \mathbf{P}_g} \right) = k_{vc} \mathbf{I}^{3 \times 3} \quad (\text{A.13})$$

Partial squared E_{tvc} over partial \mathbf{P}_g partial ω_g is derived from eq. (A.3). As ω_g is irrelevant to \mathbf{P}_g , so ω_g also irrelevant to the first order derivative in eq. (A.3), having:

$$\frac{\partial^2 E_{tvc}}{\partial \mathbf{P}_g \partial \omega_g} = \frac{\partial}{\partial \omega_g} \left(\frac{\partial E_{tvc}}{\partial \mathbf{P}_g} \right) = \mathbf{0} \quad (\text{A.14})$$

Partial squared E_{tvc} over partial ω_g partial \mathbf{P}_g is derived from eq. (A.4). As \mathbf{P}_g is irrelevant to ω_g , so \mathbf{P}_g also irrelevant to the first order derivative ineq. (A.4), having:

$$\frac{\partial^2 E_{tvc}}{\partial \omega_g \partial \mathbf{P}_g} = \frac{\partial}{\partial \mathbf{P}_g} \left(\frac{\partial E_{tvc}}{\partial \omega_g} \right) = \mathbf{0} \quad (\text{A.15})$$

Partial squared E_{tvc} over partial ω_g squared is derived from eq. (A.4) following the derivation steps in [43], having:

$$\frac{\partial^2 E_{tvc}}{\partial \omega_g^2} = \frac{\partial}{\partial \omega_g} \left(\frac{\partial E_{tvc}}{\partial \omega_g} \right) = -\frac{1}{2} \left(scalar(Quat(\omega_h) \cdot Quat(\omega_g)^{-1}) \mathbf{I}^{3 \times 3} - \widetilde{\mathbf{T}_{vc}} \right) \quad (\text{A.16})$$

where \mathbf{T}_{vc} is identical to $-\partial E_{tvc}/\partial \omega_g$.