2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE) | 979-8-3503-5317-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISCE62493.2024.10653048

Haptic Interaction Method Based on vision with object details

Meiqi Zhao
University of Chinese Academy of Sciences, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
meiqi_zhao@163.com

Ge Yu
Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
* Corresponding author: yuge@csu.ac.cn

Yu He
Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
heyu@csu.ac.cn

Han Bi
University of Chinese Academy of Sciences, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
18811627928@163.com

***Abstract*—Haptic human-computer interaction technology is a key step towards natural human-computer interaction through multisensory fusion. Haptic rendering for object details can significantly enhance the naturalness and immersion of human-computer interaction. This paper proposed an online visual-haptic integrated 6DOF haptic rendering method for interaction with the details of objects. Based on detailed surface images captured by sensors, our algorithm achieved the online construction of object details' 3D mesh model. Subsequently, an improved adaptive medial axis approximation method is utilized to build a sphere-tree model of the object details in real time. Collision detection based on the sphere-tree model and constraint-based six-degree-of-freedom collision response are utilized for force and torque calculations, which are then fed back to the operator's hands by the haptic device. Experimental results demonstrate that this algorithm meets the requirements of realism, real-time performance, and effectiveness, achieving online real-time haptic interaction for details of objects.**

Keywords-Haptic human-computer interaction; haptic rendering with details; sphere-tree model

I. INTRODUCTION

The five human senses include touch, vision, hearing, smell, and taste. Through the sense of touch, key information about the world can be perceived, such as hardness, temperature, pressure, vibration, shape, surface texture, etc.

Haptic human-computer interaction technology enables operators to obtain realistic real-time haptic interaction experiences when interacting with virtual objects through haptic interaction devices, representing a crucial step towards multi-sensory natural human-computer interaction.

Traditional data-driven haptic rendering algorithms are a method of simulating and generating haptic perception using pre-collected real haptic data. These methods can effectively simulate real haptic sensations, but require specialized data collection equipment and have a discontinuous data collection-rendering-feedback process. In contrast, vision-based haptic rendering algorithms do not require specific data collection equipment; only a camera is needed as the main data collection device. Moreover, they can achieve real-time online data collection-rendering-feedback processes.

Achieving haptic rendering of object details is essential for enhancing the immersion and realism of haptic interaction systems. Some new vision sensors (such as GelSight), through collecting and processing visual images, can capture geometric detail data of any material surface with micron-level precision, and are convenient and easy to use.

This paper proposed a vision-based haptic rendering method for object details, enabling stable real-time haptic rendering of object details. The main contributions include the following three points:

(1) Online geometric detail acquisition and modeling: High-resolution visual images of arbitrary real object surfaces are captured in real-time using portable sensors. Then, based on the geometric details and shape information in these images, an improved adaptive medial axis approximation method is employed to construct an online sphere-tree model of the object's geometric details.

(2) Collision detection algorithm based on the sphere-tree model: Employing a sphere-tree collision detection algorithm to determine whether collisions occur between objects and tools, and outputting collision sphere pair information. This approach addresses the complex calculation problem of six-degree-of-freedom mesh models and achieves fast and effective computation.

(3) Constraint-based six-degree-of-freedom collision response and virtual force/torque calculation: Determining optimization objectives based on the principle of minimum total potential energy, establishing constraint equations based on collision detection information, using the efficient set method to solve for the position/orientation of the graphical tool, calculating six degrees of freedom force and torque, and providing feedback to the operator's hand.

II. RELATED WORK

A. Data-Driven Haptic Rendering Method

Data-driven haptic rendering algorithms model and simulate haptic perception by offline capturing real haptic data collected from the interaction device.

Amit [1] proposed a novel data-driven approach for haptic modeling of normal interactions with uniformly viscoelastic deformable objects. The method employs regression using random forest machine learning techniques to acquire discrete-time interaction data from many automatic cyclic compressions of the deformable object. Results demonstrate that models trained with only 10% of the training data can accurately model unseen complex normal homogeneous interactions, thereby enabling handling of large and complex datasets. Abdulali [4] developed an end-to-end framework enabling users to collect tactile feedback data from real materials, build models, and render them in real-time finite element simulations, aiming to simulate realistic deformations and provide a genuine tactile experience. Experimental results indicate that the force feedback is at a reasonably realistic level, with relative errors lower than noticeable human force perception differences.

Joolekha [2] applied data-driven methods to haptic texture modeling and rendering for rigid interactions based on handwriting pens. They designed a complete end-to-end data-driven framework, synthesizing accelerated profiles based on the proposed deep spatio-temporal network. Joolekha [3] also proposed a novel data-driven approach for tactile texture modeling using a deep multimodal network. The network is trained with contact acceleration data collected while scanning handwriting pens on texture surfaces with different speeds, directions, and forces, aiming to recreate acceleration curves in real-time.

B. Vision-based haptic rendering

In recent years, with the rapid development of deep learning and computer vision, many researchers have started investigating the tactile channel through the visual channel. This includes studying the correlation between visual features and tactile features, estimating tactile properties from visual images, computing shear and normal forces using visual methods, and implementing strategies for multi-modal control of robotic arm actions based on visual-tactile fusion.

Takahashi [5] installed a network camera and uSkin force sensor on the end effector of a Sawyer robot to collect RGB images and corresponding force sensor data from 25 different material surfaces. They proposed a model to estimate tactile properties solely from visual perception (e.g., degree of slipperiness or roughness) and learned to correlate important tactile attributes with images. Cai [6] introduced a deep learning-based method that uses visual images of material surfaces as visual data and accelerometer signals caused by sliding motion on the surface as force data. By utilizing the framework of Generative Adversarial Networks (GAN), they significantly improved the performance of cross-modal data generation. Heravi [7] proposed a learned action-conditioned model that predicts sensed accelerations usable for providing haptic vibration feedback to users, using data from vision-based sensors (Gelsight) and user actions as input. The model was trained on the publicly available Penn Haptic Texture Toolkit dataset and generalized to new actions and instances of material categories in the dataset.

Additionally, some methods utilize visual 3D modeling techniques to construct virtual models, implement force feedback interactions, and provide force feedback to users' hands. Vasudevan [8] described an algorithm for generating force from two-dimensional bitmap digital images, allowing users to feel realistic force feedback of image contours using haptic devices for static image haptic interaction. Li [9] described a method for generating force from two-dimensional static images, creating a height map from the shape in the shadow using the Tsai & Shah algorithm, and using haptic devices to output the force field of the newly generated haptic texture rendering model.

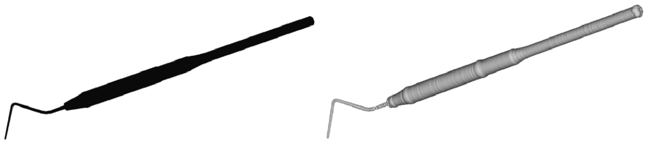
III. 6DOF HAPTIC RENDERING METHOD BASED ON VISION SENSOR DATA

A. Construction of a sphere-tree model for object details based on vision

To simulate realistic haptic feedback, it is necessary to perform three-dimensional modeling of both the tool and the object. As the complexity of the model increases, it becomes challenging to establish an effective set of constraint equations between mesh-mesh interaction of multi- objects. The sphere-tree model offers a possible solution to this challenge. The sphere-tree model is a hierarchical octree-based data structure where each node can have up to eight child nodes. It accurately describes containment, intersection, tangency, and separation states based on sphere center distances. The constraint types are simple and unified, and collision detection between sphere-sphere is easily accomplished due to the isotropic nature of spheres.

For object modeling, any object can be selected to make contact with the surface of the GelSight sensor, and a three-dimensional mesh model of the object can be reconstructed based on the visual images captured by GelSight. Next, the mesh model is converted into a sphere-tree model using the Adaptive Medial-Axis Approximation method [10]. The core idea of this method is to use the Voronoi diagram to find the object's medial axis. A combination algorithm is then employed to generate a sphere tree with points on the axis serving as sphere centers. Finally, a sphere reduction algorithm is applied to produce the optimal fit of the three-dimensional object shape with the minimum number of spheres.

Selecting a tool suitable for fine operations as the virtual tool to perceive object details, a three-level sphere-tree model is generated, representing the tool shape with 512 spheres. The shape of the virtual tool's mesh model and sphere-tree model are depicted in the figure 1.



(a) (b)

Figure 1. Virtual Tool Modeling. Virtual tool modeling. (a) 3D mesh model (b) sphere-tree model

B. Collision Detection Based on Sphere-Tree Model

The goal of collision detection algorithm is to determine whether a collision occurs between the object and the tool. The advantage of sphere-tree to sphere-tree collision detection lies in its simplicity, as it does not require considering complex constraint conditions. It only involves computing the distance between the centers of the spheres and comparing it with the sum of their radii to determine if a collision occurs.

In the collision detection process based on the sphere-tree model, the algorithm utilizes the hierarchical structure of the sphere-tree model to perform collision search from top to bottom, gradually narrowing down the search scope. Initially, it checks whether the root spheres of the two sphere trees intersect. If they intersect, collision detection is performed on the child spheres of the intersecting spheres at the next level, and so on until collision detection is carried out for all leaf child spheres or no spheres intersect.

The output of the collision detection algorithm is the information of intersecting sphere pairs, i.e., the collection of all intersecting pairs of leaf node spheres. If no intersection is detected, the result is empty, indicating that no collision occurs between the object and the tool. The position/orientation of the graphical tool remains the same as the physical tool and proceeds to the next haptic rendering loop.

If a collision occurs, collision response calculation is performed based on the contact information to obtain the position/orientation of the graphical tool. This will be detailed in the next section.

C. Constraint-Based 6DOF rigid body Collision Response

The goal of collision response is to determine the position/orientation of the graphical tool. This paper utilizes a constraint optimization approach based on the results of collision detection.

Following the principle of minimizing the total potential energy, the graphical tool and the physical tool are connected by translational and torsional springs. When the haptic tool intersects with an object in the virtual environment, the total potential energy of the springs needs to be minimized while keeping the positions of the two tools separated. Based on the principle of minimizing total potential energy, the optimization objective of the model is minimized using the least squares difference minimization method. The formula is given by:

$$Minimize: \frac{1}{2}(q_g^t - q_h^t)^T G(q_g^t - q_h^t) \tag{1}$$

where q_g^t represents the pose of the graphical tool in the current simulation loop, q_h^t represents the pose of the physical tool in the current simulation loop, and G is a diagonal stiffness matrix.

The optimization model's constraint equations are established using the collision detection information from the previous section. Upon detecting intersecting sphere pairs between the tool and objects, the contact sphere pair information in the virtual environment is mapped to the tool's configuration space, transforming it into constraints on the tool's configuration. The constraint conditions for the optimization model are as follows:

$$(x_T - x_O)^2 + (y_T - y_O)^2 + (z_T - z_O)^2 \geq (r_T + r_O)^2 \tag{2}$$

where $s_T = (x_T,\ y_T,\ z_T,\ r_T)$ $s_O = (x_O,\ y_O,\ z_O,\ r_O)$, $(x,\ y,\ z,\ r)$ represents the center and radius of a sphere in global coordinates, and subscripts T and O indicate spheres in the tool and object, respectively.

This model is a typical quadratic programming problem with inequality constraints, often solved using the active-set method [11]. The core idea is to convert inequality constraints into equality constraints at each iteration, thereby using the Karush–Kuhn–Tucker (KKT) conditions to solve the quadratic programming problem with equality constraints.

Due to the continuous motion of the tool and high update rate of haptic rendering loop, the position/orientation of the graphical tool at the current moment is very close to its position at the previous moment in configuration space. Therefore, the initial iteration point for the current moment can be chosen as the pose at the previous moment q_g^{t-1}.

D. Force and Torque Calculation

A spring force model [12] is adopted to calculate the 6-DoF feedback (i.e., feedback force and torque) by multiplying the values of the diagonal stiffness matrix by the difference in position/orientation variables between the physical tool and the graphical tool. The resulting force and torque are then sent to the haptic interaction device. The formula is as follows:

$$F = G(q_g^t - q_h^t) \tag{3}$$

Where $q_g^t = (x_g^t, y_g^t, z_g^t, \gamma_g^t, \beta_g^t, \alpha_g^t)^T$ is obtained through collision response calculation, and q_h^t is obtained by reading from the device before each simulation loop, t represents the current simulation loop and g represents the graphical tool. The values on the diagonal of the 6x6 matrix are selected as $G = \{kt, kt, kt, kr, kr, kr\}$.

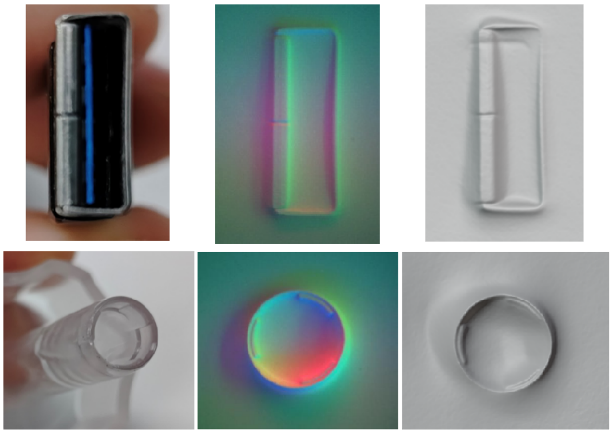
IV. EXPERIMENTS

To evaluate the online visual-haptic fusion 6DOF haptic rendering method for geometric details, the performance of the haptic rendering algorithm is assessed in two scenarios: multi-region contact and geometric detail contact. The evaluation focuses on aspects such as interactive real-time capability, feedback force stability, and realism.

A haptic interaction interface is developed using OpenGL for graphical display. The computer specifications include an Intel(R) Xeon(R) Silver 4210R CPU running at 2.40 GHz, 2.39 GHz, 16 GB of RAM, and an NVIDIA Quadro RTX 6000 graphics card.

Randomly select two objects with detailed surfaces (figure 2 a). Use the Gelsight sensor to capture real-time visual images (figure 2 b) and model the surfaces of objects with details as three-dimensional mesh models online (figure 2 c).

Experiments Setup is shown in Figure 3. The interactive interface provides real-time visual perception, while the Geomagic Touch haptic interaction device provides real-time haptic feedback, a setup that enables a real-time, stable visual-haptic interaction experience.



(a) (b) (c)

Figure 2. Online geometric detail acquisition and modeling. (a) Detailed surfaces of two real objects. (b) Real-time visual images captured by Gelsight sensors. (c) Online modeling of 3D mesh models

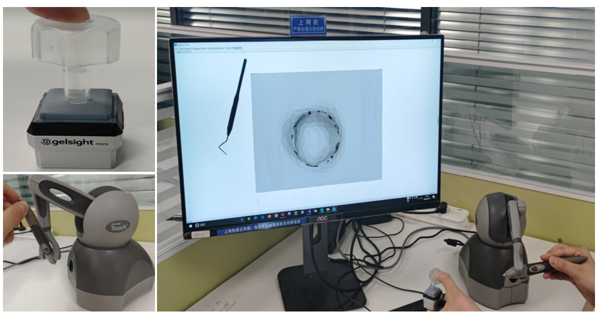


Figure 3. Experiments setup

A. Multi-region Interaction

The interactive experiment of multi-region contact can well test the real-time and stability of the rendering algorithm when dealing with multiple contact points simultaneously. During the experiment, by observing the position of the interactive tool in the interactive interface, the operator operated the virtual tool to make multi-zone contact with the virtual object model. The force, moment and time consumption of the force rendering cycle corresponding to object 1 and object 2 are shown in the figure 4.

As can be seen from the figure 4, the tip of the virtual tool is in contact with the plane of the object, while the handle of the tool is also in contact with the raised details of the object. In such a multi-region contact interaction scenario, the magnitude and direction of the force and torque generated by the interaction are relatively stable, and the feedback curve remains stable as a whole. And the time consumption of each force rendering cycle is less than 1ms, which can strictly ensure the refresh frequency of 1KHz of force rendering cycle and meet the real-time requirements.

B. Geometric Detail Interaction

The rendering effects of virtual object detail features need to be tested through interactive experiments involving contact with geometric detail areas. The virtual tool is operated to slide continuously and steadily on the surfaces of two virtual objects, referred to as Object 1 and Object 2. The corresponding forces, torques, and time consumption for haptic rendering are shown in figure 5.

From the feedback force/torque curves of the two objects, it can be observed that when sliding on the relatively flat areas of the objects, the force and torque remain stable. However, when sliding onto the raised geometric detail areas of both objects, there is a periodic fluctuation in the vertical force component, characterized by a slight increase followed by a decrease. Additionally, from the curve of Object 1, it is evident that the curve corresponding to the first raised area has a longer duration, higher absolute force value, and smoother peak compared to the curve corresponding to the second raised area, consistent with the geometric features of Object 1. Similarly, from the curve of Object 2, a slight decrease followed by an increase in the vertical torque component can be observed when sliding onto the raised geometric detail areas, consistent with the geometric features of Object 2.

Overall, the results from the two interactive experiments indicate good performance during interaction with the object models, whether in multi-region contact or geometric detail contact scenarios. The feedback forces remain stable throughout the interaction process, with no abrupt changes or significant spikes observed in any direction. Moreover, the total time consumption for the haptic rendering loop is less than 1 ms, demonstrating that the proposed online visual-haptic fusion 6DOF haptic rendering method meets the design requirements for realism, real-time performance, and stability in complex haptic interaction scenarios involving multi-region and geometric detail contact.

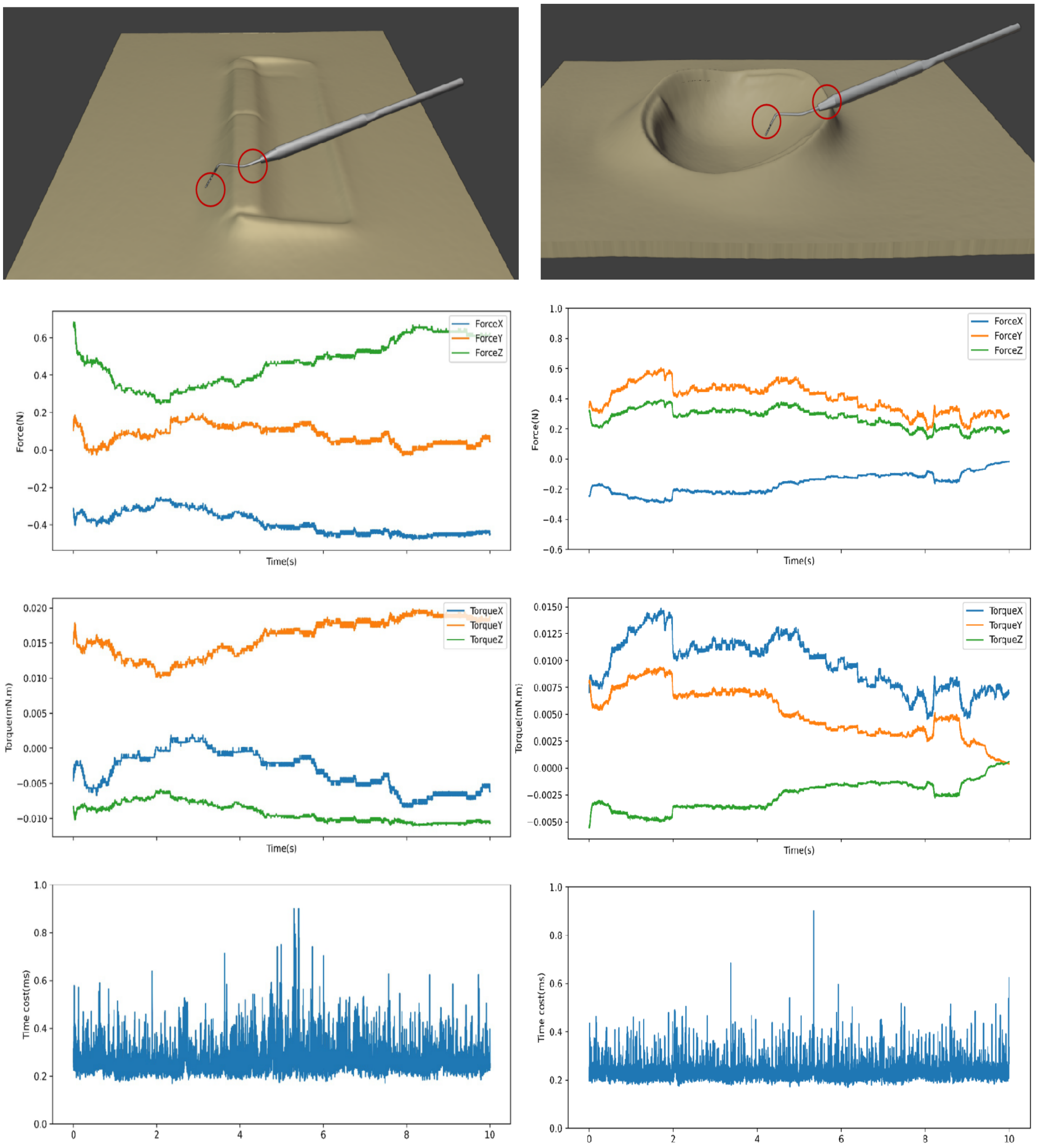


Figure 4. Force/Torque Curves of Multi-Region Interaction

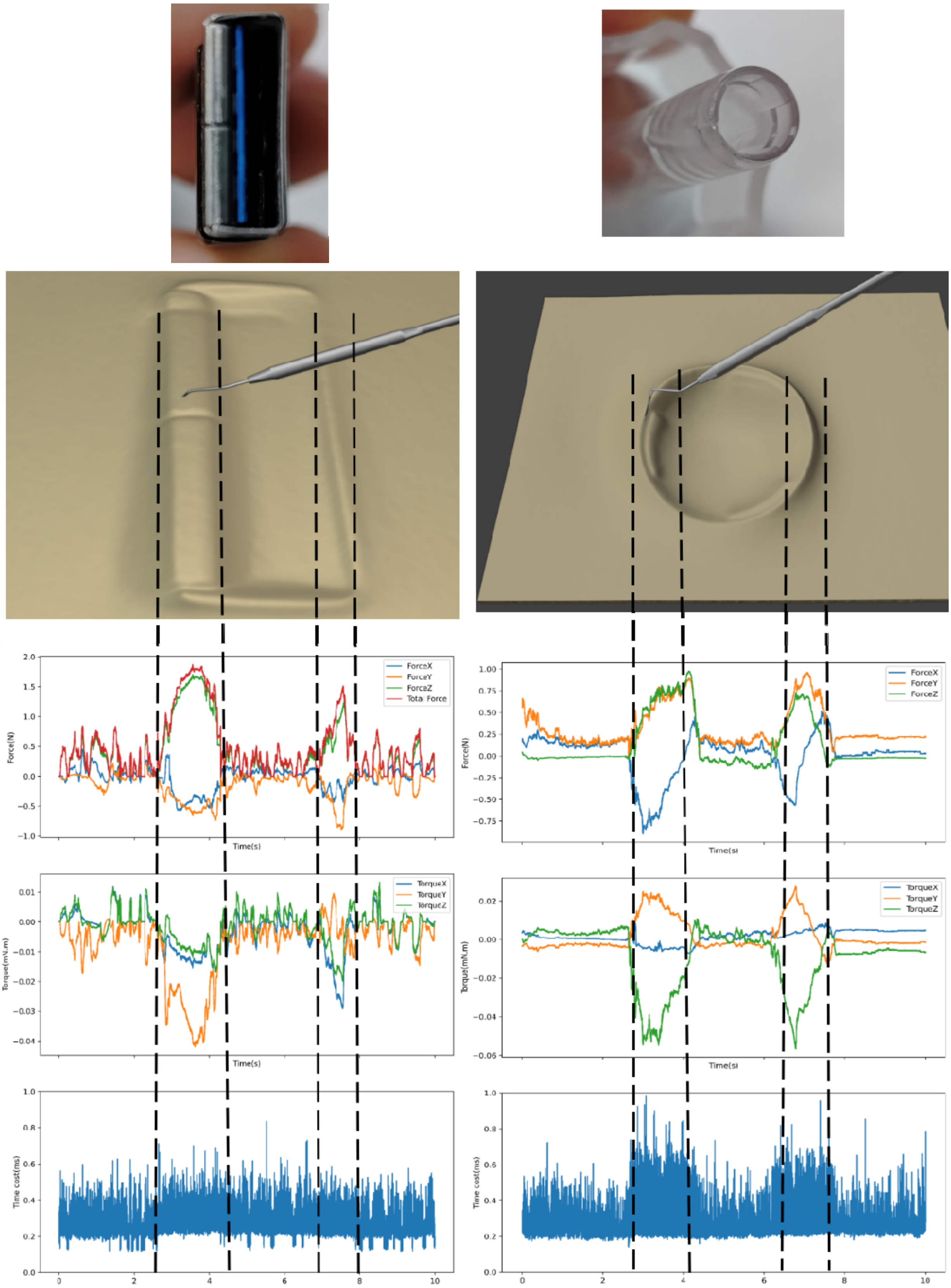


Figure 5. Force/Torque Curves for Geometric Detail Interaction

V. CONCLUSIONS

In summary, this paper introduces a novel approach to haptic interaction of object details based on vision, aiming to provide users with a realistic haptic experience. Through the proposed method, users can manipulate tools and perceive real haptic sensations in the virtual environment, enhancing the naturalness of human-computer interaction. The interactive experiments conducted verified the real-time performance and stability of the haptic rendering algorithm, indicating its practical application potential in fields such as robotics, virtual reality, and artificial intelligence.

REFERENCES

[1] Bhardwaj A, Cha H, Choi S. Data-driven haptic modeling of normal interactions on viscoelastic deformable objects using a random forest[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1379-1386.

[2] Abdulali A, Jeon S. Data-driven haptic modeling of plastic flow via inverse reinforcement learning[C]//2021 IEEE world haptics conference (WHC). IEEE, 2021: 115-120.

[3] Joolee J B, Jeon S. Data-driven haptic texture modeling and

rendering based on deep spatio-temporal networks[J]. IEEE Transactions on Haptics, 2021, 15(1): 62-67.

[4] Joolee J B, Jeon S. Data-driven haptic texture modeling and rendering based on deep spatio-temporal networks[J]. IEEE Transactions on Haptics, 2021, 15(1): 62-67.

[5] Takahashi K, Tan J. Deep visuo-tactile learning: Estimation of tactile properties from images[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 8951-8957.

[6] Cai S, Zhu K, Ban Y, et al. Visual-tactile cross-modal data generation using residue-fusion gan with feature-matching and perceptual losses[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7525-7532.

[7] Heravi N, Yuan W, Okamura A M, et al. Learning an action-conditional model for haptic texture generation[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 11088-11095.

[8] Vasudevan H, Manivannan M. Tangible images: runtime generation of haptic textures from images[C]//2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. IEEE, 2008: 357-360.

[9] Li J, Song A, Zhang X. Haptic texture rendering using single texture image[C]//2010 International Symposium on Computational Intelligence and Design. IEEE, 2010, 1: 7-10.

[10] Bradshaw G, O'Sullivan C. Adaptive medial-axis approximation for sphere-tree construction[J]. ACM Transactions on Graphics (TOG), 2004, 23(1): 1-26.

[11] E. Ruffaldi, D. Morris, F. Barbagli, K. Salisbury and M. Bergamasco, "Voxel-Based Haptic Rendering Using Implicit Sphere Trees", Proc. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2008.

[12] Yu G, Wang D, Zhang Y, et al. Simulating sharp geometric features in six degrees-of-freedom haptic rendering[J]. IEEE Transactions on Haptics, 2014, 8(1): 67-78.