공학석사학위논문

치과시술 햅틱시뮬레이터의 Z-width 향상 기법 연구

Dental Simulator with Increased Z-width of Haptic Rendering

2019년 2월

서울대학교 대학원

기계항공공학부

박 효 준

Abstract

Dental Simulator with Increased Z-width of Haptic Rendering

Hyojoon Park
Mechanical & Aerospace Engineering
The Graduate School
Seoul National University

A novel haptic rendering framework for dental simulators is introduced. The primary focus is rendering a virtual tooth as stiff as it is in real world. In short, our framework displays around ten times larger Z-width than the standard explicit method.

For development, we use passive midpoint integration (PMI) to enhance system stability, and exploit on the transparent virtual coupling framework using passive decomposition to render stiffer force feedback beyond the limits in haptic device's physical damping. Also, we derive and use momentum-based human force observer for estimating human forces acting on the device, which is required for passive decomposition. This observer enables our framework to be applicable to other multi-DoF haptic devices with no force sensors and large physical damping. In addition, we incorporate dynamics augmentation controls into passive decomposition to render respective dynamics for different virtual tools. Lastly, we exploit on the full octree-based voxel rendering for real-time collision detection and drilling of the virtual tooth.

Keywords: Z-width, Haptic rendering, Interactive simulation, Dental simulator, Passive midpoint integration, Passive decomposition, Momentum-based human force observer, Full octree-based voxel rendering, Dynamics augmentation control, Virtual reality.
Student Number: *2017-24482*

Contents

List of Figures

Abbreviations

DoF	**D**egrees of **F**reedom
PMI	**P**assive **M**idpoint **I**ntegration
SE(3)	**S**pecial **E**uclidean Group in **3**-Dimensional Space
SO(3)	**S**pecial **O**rthogonal Group in **3**-Dimensional Space
E(3)	**E**uclidean Group in **3**-Dimensional Space
so(3)	The Lie algebra of **SO(3)**
DOB	**D**isturbance **OB**server

Chapter 1

Introduction

1.1 Motivation and Objectives

Virtual reality dental simulators with haptic force feedback are known to have significant potential benefits in teaching manual skills in dentistry [5]. It is desirable for dental simulators to display large haptic stiffness since it enables the virtual tooth to *feel* more realistic, increasing the quality of haptic rendering overall. However, most commercially available dental simulators are unable to render virtual tooth as stiff as it is in real world, due to stability issues, and limited physical damping of haptic devices.

Our objective is to overcome these issues by developing a framework for dental simulators which can render stiff haptic force feedback by maintaining stability of dynamical system, applicable to general multi-DoF haptic devices (i.e., with no force sensors and limited physical damping (Fig. 1.1)).

FIGURE 1.1: Commercially available haptic devices (3D Systems).

To assess the displayed stiffness, we use the concept of Z-width computed as the ratio of total force feedback to the maximum penetration depth. By definition, Z-width is the dynamic range of impedance that can be rendered by a haptic device; increasing the maximum impedance is a way of increasing the Z-width of haptic rendering. In general, larger Z-width renders *better* feeling of the virtual environment, thus the *better* quality of haptic rendering [6]. For this reason, we focus on increasing the Z-width of haptic rendering for our framework, analogous to rendering a virtual tooth as stiff as it is in real world.

1.2 Related Works

The issue of instability in haptic devices, with respect to Z-width, is addressed in [6]. To briefly review, passivity is a useful tool for studying both the stability and performance of haptic rendering; an arbitrary virtual environment is guaranteed passive if a haptic rendering is passive. Here, Z-width is the range of dynamic impedance that can be rendered by a haptic display while maintaining passivity. Since large Z-width implies better quality of haptic rendering, it is desirable to maintain passivity while increasing the maximum impedance.

Meanwhile, haptic rendering and interactive simulation using passive midpoint integration (PMI) is studied and developed for 6-DoF dynamical systems in [7]. The study provides passivity analysis and comparisons between different numerical methods including explicit Euler integration, implicit Euler integration, and symplectic Euler integration. Due to the energy-conserving property of PMI, the PMI-based haptic rendering and interactive simulation (in short, *haptic rendering* in the rest of the paper) framework is stable against light virtual proxy inertia, large stiffness of virtual coupling, and varying time-steps, compared to other integration methods. So far, however, there has been no verified implementation of the 6-DoF haptic rendering framework using PMI, which makes our work the first to fully implement it.

Also, the transparent virtual coupling work is developed in [8] utilizing PMI and passive decomposition with 3-DoF human force observer. The framework uses dynamics of the haptic device for computing feedforward actions, and overcomes the device's upper bounded physical damping. As a result, the coordination between the haptic device and the virtual proxy is maintained even during contacts. Moreover, the transparent virtual coupling framework allows a human operator to feel the stiffness of the virtual environment directly by the feedforward actions.

However, the framework is developed in generalized coordinates which makes it incompatible for virtual objects with SE(3) motion, and is limited to 3-DoF devices due to its human force observer. In this work, we derive expressions for passive decomposition for SE(3) motion, operated using 6-DoF haptic device.

Passive decomposition necessitates measuring of human force acting on the device. Most haptic devices, however, do not have force sensors. Instead of attaching bulky and relatively expensive force sensor to the device, our framework estimates the human force using the momentum-based human force observer. The momentum-based human force observer is derived from disturbance observer (DOB) which is widely used in many practical applications due to its simple structure and high performance, but cannot be directly applied to nonlinearities and couplings in the inertia matrix. By utilizing a generalized momentum, it is possible to exploit on DOB for nonlinear and coupled inertia matrix [9].

Lastly, fast update rate is required for maintaining the stability of interactive simulation. Many techniques exist to facilitate computationally expensive collision detection; one is octree-based voxel rendering. Although several varieties exist, full octree provides fast look up and simple neighbor search since it computes the index of any node based on its location in the tree, instead of pointers [10].

1.3 Contribution

The contribution of this paper is the development of the haptic rendering framework for dental simulators (Fig. 1.2), capable of displaying significantly increased Z-width of haptic rendering, and can be used with other multi-DoF haptic devices with no force sensors and large physical damping.

Consequently, our framework displays Z-width around ten times larger than that of the standard explicit virtual coupling framework. This implies the virtual tooth feels ten times stiffer, making it more haptically realistic.

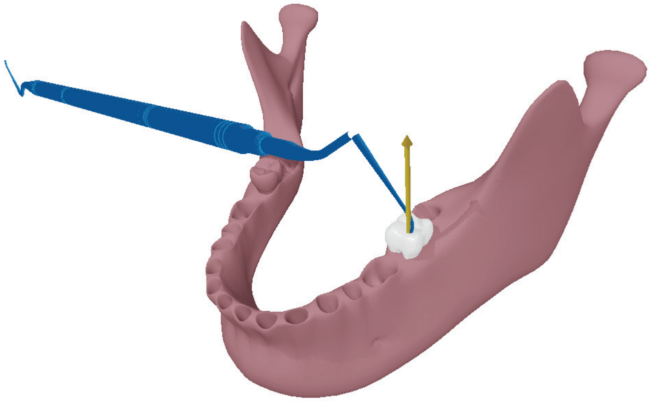


FIGURE 1.2: Dental simulator with increased Z-width of haptic rendering.

The significance of our framework is it maintains the system stability by exploiting on passive midpoint integration (PMI). This contributes to rendering of stiff haptic force feedback by allowing use of large spring stiffness due to its energy-conserving property and its fast and non-iterative property.

Moreover, using passive decomposition our framework can generate stiffer haptic force feedback beyond the limits of haptic device's intrinsic and uncontrollable physical damping. The momentum-based human force observer is also used for estimating human forces acting on the device, allowing our framework to be applicable to other general haptic devices with no force sensors.

Lastly, our utilization of full octree-based voxel rendering allows real-time collision detection and drilling of virtual tooth.

Chapter 2

System Modeling

2.1 Dynamics

System modeling begins with the dynamics of the haptic device and the virtual proxy. We model the device and the proxy (i.e., virtual tool) in the continuous-time domain, and utilize PMI to derive the governing equations in the discrete-time domain. Although our framework is applicable to other multi-DoF haptic devices, we use 6-DoF device with nonlinear robot dynamics.

2.1.1 Haptic Device

The device dynamics is derived using Euler-Lagrange equations in generalized coordinates, then converted into maximal coordinates. Joint frictions and gravity are assumed to be fully compensated. The device dynamics is modeled as

$$M_1(q)\ddot{q} + C_1(q,\dot{q})\dot{q} = \tau_1 + f_1 \tag{2.1}$$

where, $M_1, C_1 \in \mathcal{R}^{6\times 6}$ are inertia and Coriolis matrix, respectively; $\tau_1, f_1 \in \mathcal{R}^6$ are joint control input and human operator wrench, respectively, and $q \in \mathcal{R}^6$ is the joint angle measurement vector.

We further define $J_1 \in \mathcal{R}^{6\times 6}$, a modified Jacobian defined as

$$\dot{q} := J_1 \begin{bmatrix} \dot{x}_1^T & w_1^T \end{bmatrix}^T,$$

used for converting the device dynamics from generalized to maximal coordinates, and vice versa. The device dynamics in maximal coordinates is expressed as

$$J_1^T M_1 J_1 \begin{bmatrix} \ddot{x}_1 \\ \dot{w}_1 \end{bmatrix} + Q_1(x_1, R_1, \dots) \begin{bmatrix} \dot{x}_1 \\ w_1 \end{bmatrix} = T_1 + F_1. \tag{2.2}$$

In the discrete-time domain, position $q_{1,k}$ and velocity $dq_{1,k}$ of each joint are sampled at time t_k, and are used for the time duration $T_k = [t_k, t_{k+1})$. After sampling at frame k, the continuous dynamics (2.2) is expressed as

$$J_{1,k}^T M_{1,k} J_{1,k} \begin{bmatrix} \frac{v_{1,k}-v_{1,k-1}}{T_k} \\ \frac{w_{1,k}-w_{1,k-1}}{T_k} \end{bmatrix} + Q_{1,k}(x_{1,k}, R_{1,k}, \dots) \begin{bmatrix} v_{1,k} \\ w_{1,k} \end{bmatrix} = T_{1,k} + F_{1,k}.$$

2.1.2 Virtual Proxy

The virtual proxy is modeled as a rigid body in SE(3), with its center of mass located at the tool's tip. We follow the maximal coordinate modeling approach described in [7] as

$$M_2 \begin{bmatrix} \ddot{x}_2 \\ \dot{w}_2 \end{bmatrix} + Q_2(w_2) \begin{bmatrix} \dot{x}_2 \\ w_2 \end{bmatrix} = T_2 + F_2, \tag{2.3}$$

where $B_2 \in \mathcal{R}^{6\times 6}$ is the diagonal damping matrix of the proxy, and $T_2, F_2 \in \mathcal{R}^6$ are the control input and environment force, respectively. The proxy dynamics is comprised of the following translational $(\star)_x$ and rotational $(\star)_w$ terms.

$$m\ddot{x}_2 + b_x\dot{x}_2 = T_{2,x} + F_{2,x} \tag{2.4}$$

$$\begin{aligned} J\dot{w}_2 + w_2 \times Jw_2 + b_w w_2 &= T_{2,w} + F_{2,w} \\ \dot{R}_2 &= R_2 S(w) \end{aligned} \tag{2.5}$$

where, $x \in E(3)$ is the position with the mass $m > 0$, $b_x \in \mathcal{R}^{3\times 3}$ is positive semi-definite damping matrix. $T_x, F_x \in \mathcal{R}^3$ are the control and the environment force, respectively, all expressed in the inertial frame $\{O\}$; whereas, $R \in SO(3)$ is the rotation matrix of the body frame $\{B\}$ w.r.t. $\{O\}$, $w \in so(3)$ is the angular velocity, $J \in \mathcal{R}^{3\times 3}$ is the moment of inertia, $b_w \in \mathcal{R}^{3\times 3}$ positive semi-definite damping matrix, and $T_w, F_w \in \mathcal{R}^3$ are the control and the environment moment, respectively, all expressed in the body frame $\{B\}$ w.r.t $\{O\}$.

We adopt the PMI formulation for the virtual proxy dynamics (2.4)-(2.5), as in [7]. After deriving the discrete-time integrator map

$$\mathcal{M} : (x_k, v_k, f_k) \to (x_{k+1}, v_{k+1}),$$

and the representative variables $\hat{v}_k, \hat{f}_k$, such that the mapping $\mathcal{M}$ exactly matches the continuous-time passivity relation

$$\int_{t_k}^{t_{k+1}} fv dt = \frac{1}{2} m(v_{k+1}^2 - v_k^2) + \frac{1}{2} k(x_{k+1}^2 - x_k^2) + \int_{t_k}^{t_{k+1}} bv^2 dt, \tag{2.6}$$

we arrive at the following kinematics relation

$$\begin{aligned} \hat{v}_k &:= \frac{v_{k+1} + v_k}{2} = \frac{x_{k+1} - x_k}{T_k}, \\ \hat{x}_k &:= \frac{x_{k+1} + x_k}{2}, \\ \hat{f}_k &:= f_k. \end{aligned} \tag{2.7}$$

This relation is generalized and termed passive midpoint integration (PMI), and exactly matches the continuous-time passivity property (2.6) [7].

While the translation dynamics (2.4) can be formulated into the linear PMI using the kinematic relation (2.7), the attitude dynamics (2.5) can not due to its coupling term $w \times Jw$ in

$$\begin{aligned} m\frac{v_{k+1}-v_k}{T_k} + b_x\hat{v}_k &= T_{x,k} + F_{x,k} \\ J\frac{w_{k+1}-w_k}{T_k} - S(Jw_k)\hat{w}_k + b_w\hat{w}_k & \\ &= T_{w,k} + F_{w,k}, \end{aligned}$$

where $w \times Jw = -S(Jw)w$ is used with $S(Jw) \in \mathcal{R}^{3\times 3}$ being the skew-symmetric. This coupling term, however, is often omitted in computer graphic literature; thus, the attitude dynamics can also be reduced to the linear PMI as

$$J\frac{w_{k+1}-w_k}{T_k} + b_w\hat{w}_k = T_{w,k} + F_{w,k}.$$

As a result, the dynamics of the virtual proxy is (omitting the subscript $(\star)_2$)

$$M\frac{V_{k+1}-V_k}{T_k} + C_k(w_k)\hat{V}_k + B\hat{V}_k = T_k + F_k, \tag{2.8}$$

where $M := diag[mI_{3\times 3}, J] \in \mathcal{R}^{6\times 6}$ is the inertia matrix, $C_k := diag[0_{3\times 3}, -S(Jw_k)]$ with $C_k = -C_k^T, B := diag[b_x, b_w] \in \mathcal{R}^{6\times 6}$ is the damping matrix, and T_k, F_k are control and external force, respectively.

The control input T_k includes the control from virtual coupling (Sec. 2.2) and passive decomposition (Sec. 3.1), and the external force F_k includes the contact

(Sec. 2.3) and friction (Sec. 2.5).

2.2 Virtual Coupling

We establish the bilateral virtual spring-damper connection between the haptic device (i.e., master) and the virtual proxy (i.e., slave) as in Fig. 2.1. This connection is the standard virtual coupling defining how to coordinate the continuous-time haptic device and the discrete-time virtual proxy. The virtual coupling generates the coordinating force $T_{1,k}$ in the maximal coordinates for the device, and $T_{2,k}$ for the proxy [8].

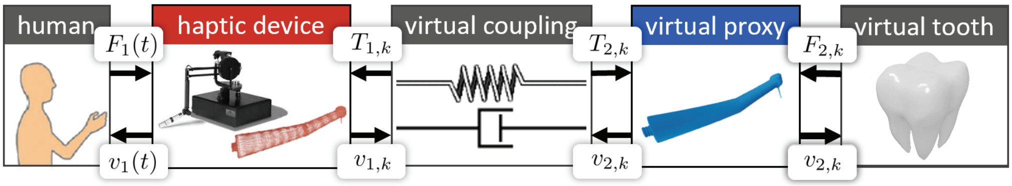


FIGURE 2.1: Spring-damper virtual coupling framework.

- **Haptic Device**

  The control input $T_{1,k}$ for coordination is expressed as

$$T_{1,k} = -K_{vc}\Delta X_{12,k} - B_{vc}\Delta V_{12,k}, \tag{2.9}$$

where $T_{1,k} \in \mathcal{R}^6$ is composed of the translational part $T_{1,x} \in \mathcal{R}^3$ and the rotational part $T_{1,w} \in \mathcal{R}^3$, formulated as

$$\begin{aligned}T_{1,x} =& -k_{vc,x}(x_{1,k} - x_{2,k}) - b_{vc,x}(v_{1,k} - \hat{v}_{2,k-1}),\\ T_{1,w} =& -k_{vc,w}(-\frac{\phi_{12}}{2\sin\phi_{12}}\bar{w}_{12} + \frac{T_k}{2}w_{1,k}) - b_{vc,w}(w_{1,k} - R_1^T R_2 \hat{w}_{2,k-1}),\end{aligned} \tag{2.10}$$

where $k_{vc}, b_{vc} > 0$ are the spring stiffness and damping coefficients, respectively. Also, the rotational spring-related terms are

$$\phi_{12} =: \cos^{-1}(\frac{1}{2}[tr(R_1^T R_2) - 1]), \tag{2.11}$$

with $\bar{w}_{12} \in \mathcal{R}^3$ defined s.t.

$$S(\bar{w}_{12}) =: R_1^T R_2 - R_2^T R_1. \tag{2.12}$$

(2.11) and (2.12) are derived from the discrete-time version of rotational kinematics $R_{k+1} = R_k \exp(S(\hat{w}_k T_k))$, where $exp(\star)$ is the exponential map with the SO(3) spring potential $\psi := \frac{1}{2}k_w\phi_{12}^2$, where $k_w > 0$. Here, ψ is the geodesic on a unit sphere.

After enforcing passivity only approximately for $d\psi_{w_k}$, we arrive at

$$d\psi_{w_k}^T \approx -\frac{k_w\phi_k}{2sin\phi_k}\bar{w}_k + \frac{k_w T_k}{2}\hat{w}_k. \tag{2.13}$$

This passive SO(3) spring force $d\psi_{w_k}^T$ is incorporated in $T_{1,w}$ (2.10) [7].

- **Virtual Proxy**

  Similarly, yet with difference in sampling time, the control input $T_{2,k}$ for coordination is expressed as

$$T_{2,k} = -K_{vc}\Delta X_{21,k} - B_{vc}\Delta V_{21,k}, \tag{2.14}$$

where $T_{2,k} \in \mathcal{R}^6$ is composed of the translational part $T_{2,x} \in \mathcal{R}^3$ and the rotational part $T_{2,w} \in \mathcal{R}^3$, formulated as

$$\begin{aligned} T_{2,x} &= -\, k_{vc,x}(\hat{x}_{2,k} - x_{1,k}) - b_{vc,x}(\hat{v}_{2,k} - v_{1,k}), \\ T_{2,w} &= -\, k_{vc,w}(-\frac{\phi_{21}}{2\sin\phi_{21}}\bar{w}_{21} + \frac{T_k}{2}\hat{w}_{2,k}) - b_{vc,w}(\hat{w}_{2,k} - R_2^T R_1 w_{1,k}). \end{aligned}$$

As a result, the device and the proxy converge to each other as if they are connected with a spring-damper (Fig. 2.2).

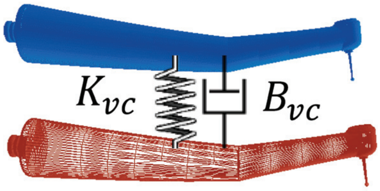


FIGURE 2.2: Spring-damper virtual coupling. Red is the device (master) and blue is the proxy (slave).

2.3 Penalty-Based Normal Contact Force

We employ the penalty-based method for 6-DoF normal contact force due to its simplicity and high performance, desirable for haptic rendering. We model the governing equation first, then derive it based on PMI to maintain passivity.

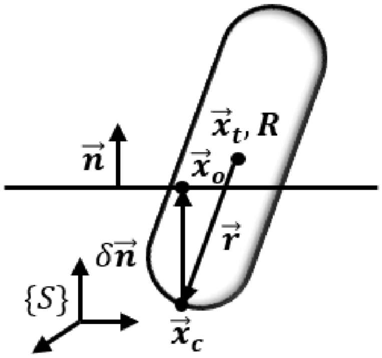


FIGURE 2.3: Penalty-based contact force modeling.

For simplicity in notations, we first define the contact point on the virtual proxy as

$$x_c =: x_t + r = x_t + Rr,$$

where $r =: x_c - x_t \in \mathcal{R}^3$ is vector from the center of the proxy to the contact point on proxy. $R \in SO(3)$ is the proxy's orientation matrix. Then, the penetration depth $\delta \in \mathcal{R}$ parallel to the normal force vector $n \in \mathcal{R}^3$ is

$$\delta =: n^T(x_o - x_c)$$

Then, the 6-DoF contact force is modeled as

$$F_2 =: \begin{bmatrix} f_n^s \\ \tau_n^b \end{bmatrix} = \begin{bmatrix} I \\ R^T S(r) \end{bmatrix} f_n^s, \tag{2.15}$$

where $f_n^s, \tau_n^b \in \mathcal{R}^3$ are the normal contact force w.r.t. spatial frame and the moment w.r.t. body frame of the proxy, respectively.

Now, the objective is to model the normal force f_n^s based on PMI as

$$f_n^s =: K_{tooth} \frac{\delta_{k+1} + \delta_k}{2} n_k = K_{tooth} \left(\frac{T_k}{2} \frac{\delta_{k+1} - \delta_k}{T_k} + \delta_k \right) n_k. \tag{2.16}$$

Here, we need to find the expression for $\delta_{k+1} - \delta_k$. For this, we make two assumptions for x_o and r to simplify our derivation, as

$$\begin{aligned} x_{o,k+1} &= x_{o,k}, \\ r_{k+1} &= r_k. \end{aligned} \tag{2.17}$$

Also, we use the approximation from [7]

$$R_k^T R_{k+1} = \exp\left(S(\hat{w}_k T_k)\right) \approx I + S(\hat{w}_k T_k). \tag{2.18}$$

Combining (2.16), (2.17), and (2.18) yields the following expression for f_n^s.

$$\begin{aligned} f_n^s &= K_{tooth}\left(\frac{T_k}{2}\frac{\delta_{k+1}-\delta_k}{T_k}+\delta_k\right)n_k \\ &= K_{tooth}\left(\frac{T_k}{2}n_k^T\begin{bmatrix}-I & R_k S(r_k)\end{bmatrix}\begin{bmatrix}\hat{v}_k \\ \hat{w}_k\end{bmatrix}+n_k^T(x_{o,k}-x_{c,k})\right)n_k \end{aligned}$$

Finally, we substitute f_n^s into (2.15) for 6-DoF contact force rendering.

Also, the contact damping is modeled as

$$F_2 = B_{tooth}\begin{bmatrix} I \\ R^T S(r)\end{bmatrix}\frac{\delta_{k+1}-\delta k}{T_k}n_k \tag{2.19}$$

As a result, the contact force on the virtual tooth is simulated as in Fig. 2.4.

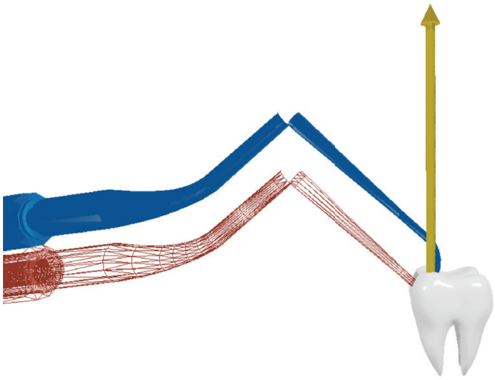


FIGURE 2.4: Penalty-based contact force rendering.

2.4 Constraint-Based Friction Force

We use constraint-based Coulomb friction model to simulate the friction on the virtual proxy movement $\hat{V}_{2,k} =: \left[\hat{v}_{2,k}\hat{w}_{2,k}\right]^T \in \mathcal{R}^6$.

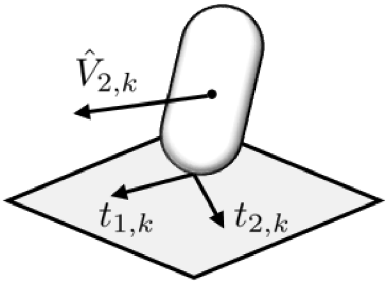


FIGURE 2.5: Coulomb friction modeling.

The friction force components are defined as $t_{1,k}\lambda_{1,k}, t_{2,k}\lambda_{2,k} \in \mathcal{R}^3$ summed to act in the opposite direction of $\hat{V}_{2,k}$. In the dynamics equation (2.8), F_k is comprised of the contact force $F_{contact,k}$ and the friction force $F_{fric,k}$. Expanding the expression (2.8) yields

$$M\frac{V_{k+1} - V_k}{T_k} + C_k(w_k)\hat{V}_k + B\hat{V}_k = T_k + F_{contact,k} + F_{fric,k}, \tag{2.20}$$

where $F_{fric,k} \in \mathcal{R}^6$ is comprised of the translational and rotational part as

$$F_{fric,k} = \begin{bmatrix} I \\ R_k^T S(r_k) \end{bmatrix} f_{fric,x},$$

with the translational part $f_{fric,x} \in \mathcal{R}^3$ defined as

$$f_{fric,x} := t_{1,k}\lambda_{1,k} + t_{2,k}\lambda_{2,k}, \tag{2.21}$$

where $\lambda_{1,k}, \lambda_{2,k} \in \mathcal{R}$ are the friction magnitudes in the direction of $t_{1,k}, t_{2,k} \in \mathcal{R}^3$, respectively. Note, the control force T_k is computed from the virtual coupling.

Now, the objective is to resist the virtual proxy movement, such that

$$t_{1,k}^T J_k \hat{V}_k \rightarrow 0,\ t_{2,k}^T J_k \hat{V}_k \rightarrow 0, \tag{2.22}$$

where $t_{1,k}, t_{2,k} \in \mathcal{R}^3$ are mutually orthogonal unit vectors, tangent to the tooth surface at a contact point, as illustrated in Fig. 2.5. J_k is defined as $F_{2,k} = J_k f_n^s \in \mathcal{R}^6$, such that

$$J_k =: \begin{bmatrix} I_{3\times 3} & S^T(r_k)R_k \end{bmatrix} \in \mathcal{R}^{6\times 3}.$$

Using (2.20) and (2.22), we compute for $\lambda_{1,k}$ and $\lambda_{2,k}$. Finally, the magnitude of friction force applied is computed as

$$\lambda_k = min(\lambda_k, \mu\delta_k),$$

where $\mu > 0$ is the friction coefficient, and δ_k is the penetration depth.

2.5 Virtual Tooth Rendering

2.5.1 Volume Rendering

Volume rendering is the technique of displaying volumetric data. It has advantages over surface rendering for its capability to represent the interior data of a volumetric model. This is important for the simulation of interactive cutting operations of virtual organs. Also, collision detection in complex virtual environment is simpler with volumetric models [11].

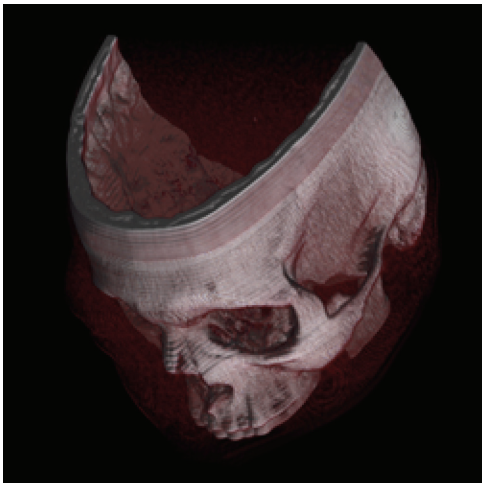


Figure 2.6: A volume rendered cadaver head [1].

Similarly, it is desirable for the virtual tooth in our simulator to have its interior information (e.g., stiffness) represented, and be interactively drilled using virtual tools. Also, since the haptic rendering required a fast update rate (i.e., above

1kHz), fast collision detection is desired. For this, we exploit on the volume-based rendering.

2.5.2 Octree

Octrees are data structures where each nodes has exactly eight children. They are frequently used for partitioning a 3D space by recursively subdividing it into eight octants (Fig. 2.7).

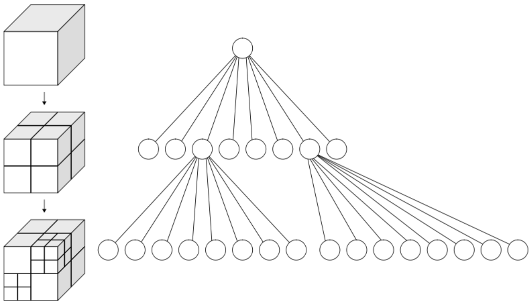


FIGURE 2.7: Recursive subdivision of a cube into octants (left), and the corresponding octree (right) [2].

Meanwhile, a voxel is a value on on a regular grid in 3D space, similar to a pixel in a 2D space. Voxels are widely used in the visualization and analysis of medical and scientific data. Similarly, voxel rendering is a rendering technique for volumetric models, which inherently offers more rapid collision detection than surface-based models. However, this high performance comes at a cost of approximated boundary representations. This degradation in accuracy can be mitigated by maximizing the resolution of voxel-based models [12], but with a cost of increased data and computational cost for accessing the voxels.

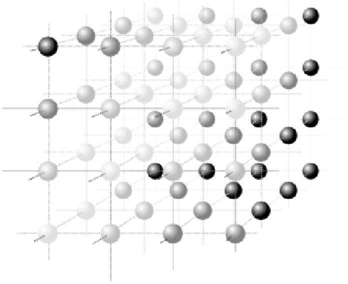


FIGURE 2.8: Illustration of a voxel grid, each containing a color value [3].

To cope with this problem, we exploit on octree for voxel rendering. Although many variants of octree exist, we adopt a full octree due to its advantage in simple implementation and performance, at a cost of high memory usage which is often trivial for rendering a single virtual tooth using modern computers.

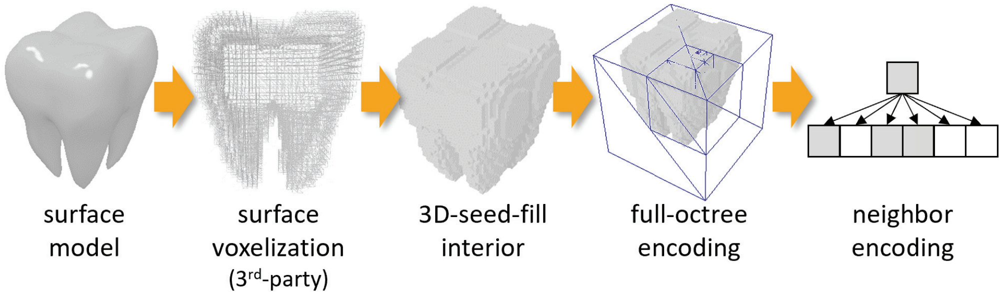


FIGURE 2.9: Steps for full octree-based voxel rendering for the virtual tooth.

The full octree computes the index of a node based on its location in the tree, instead of using pointers as with traditional pointer-based octree [10]. Although the full octree allocates unnecessary memory for non-existing voxels, it is simple to implement, and has faster lookup compared to the recursive pointer-based octree. The followings are brief descriptions on the key algorithms used in our framework for voxel rendering (Fig. 2.9).

- **3D Seed Filling** is used to pre-fill the voxelized surface model of the virtual tooth, generated using a third-party voxelizer [13]. We utilize the 3D seed filling algorithm similar to [14].

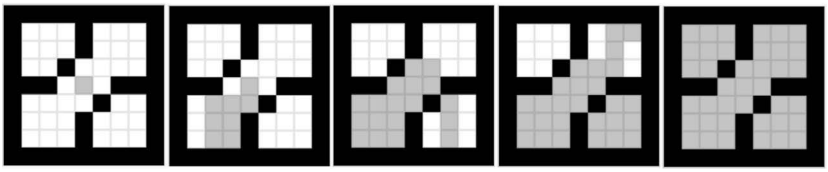


FIGURE 2.10: Illustration of seed filling in 2D [4].

- **Neighbor Encoding** is used for smooth contact rendering at the boundaries of adjacent voxels. For example, if two voxels are face-to-face adjacent, the overlapping edge of the two voxels is ignored when computing a contact force. Every time a voxel model is updated (i.e., drilled), its neighbor information on all 6 sides of every surface voxels are re-encoded.
- **Collision Detection** is facilitated using octree. At each frame, collision detection starts from the root node of the tree checking collision between each of its eight children and the tool's tip. The same process is repeated for the children nodes in collision, until the leaf nodes are reached. Then,

the leaf nodes in contact with the tool's tip are used for computing the contact force vector along with their neighbor information.

- **Drilling** is simulated simply by removing the drilled voxel. The drilling criteria for each voxel is defined using the total amount of work (i.e., in joule) done on the voxel by the handpiece drill. If the total amount of work surpasses a threshold, the tooth is removed.

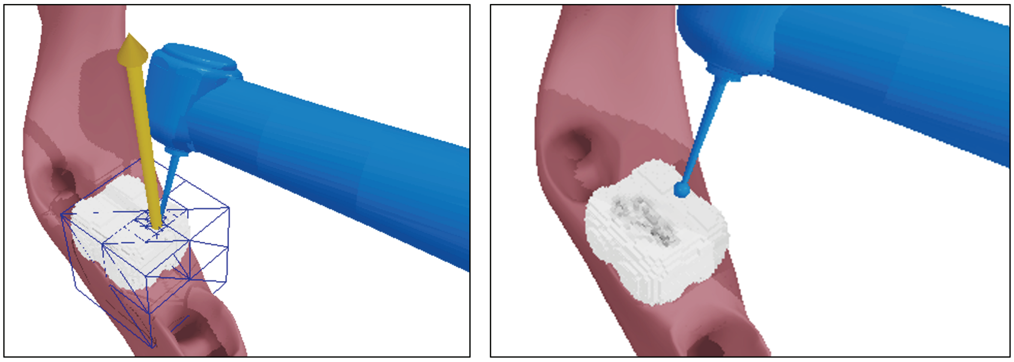


FIGURE 2.11: Full octree-based collision detection (left), and drilling of the virtual tooth (right).

Chapter 3

Transparent Virtual Coupling with Momentum-Based Human Observer

3.1 Passive Decomposition

PMI-based virtual coupling allows stable haptic rendering and interactive simulation using large spring stiffness (also low inertia, large time steps, and etc.). Recalling the objective is to render very stiff force feedback, large stiffness for the virtual coupling K_{vc} and the tooth K_{tooth} is required. The total displayed stiffness is computed using the two serially-connected spring model (Fig. 3.1) with stiffness K_{vc} and K_{tooth}, as

$$\Sigma K = \frac{K_{vc} K_{tooth}}{K_{vc} + K_{tooth}}.$$

Figure 3.1: Serially-connected springs.

This shows increasing K_{tooth} indefinitely makes the total stiffness converge to K_{vc}, which demands K_{vc} to be increased as well. However, K_{vc} is upper bounded due to the device's intrinsic and uncontrollable damping.

The supporting passivity analysis for the virtual coupling is given in [8], as

$$\begin{aligned} b_d &\geq b_{vc}(1 + \frac{T_k}{T_{k-1}}) + k_{vc} T_k, \\ b_v &\geq \frac{b_{vc}}{2}(\frac{T_k}{T_{k-1}} - 1) + \frac{k_{vc} T_k}{2}, \end{aligned}$$

where $b_d, b_v > 0$ are the device and proxy damping, respectively.

Here, the spring-damper gains $k_{vc}, b_{vc} > 0$ in virtual coupling cannot be increased indefinitely, since the device's intrinsic damping $b_d > 0$ is bounded and uncontrollable. Instead, we utilize passive decomposition for feedforward actions to mutually render human and environment force, without increasing k_{vc} beyond its limit. Passive decomposition decomposes the dynamics of the device and the proxy into that of the coordinated dynamics (i.e., the locked system) and the error dynamics (i.e., the shape system) as in Fig. 3.2.

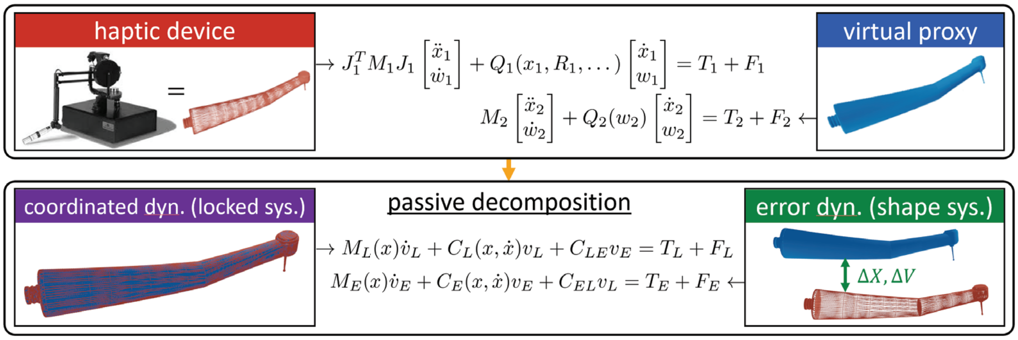


FIGURE 3.2: Passive decomposition

For this, we utilize the transparent virtual coupling framework using passive decomposition in generalized coordinates [8], and apply it to the maximal coordinates (2.3)-(2.2). We define the (non-singular) decomposition matrix $S \in \mathcal{R}^{6\times 6}$ as

$$S(x_1, x_2) =: \begin{bmatrix} I - \phi & \phi E^T \\ I & -E^T \end{bmatrix},$$

where

$$E =: \begin{bmatrix} I_3 & 0 \\ 0 & R_1^T R_2 \end{bmatrix},$$

$$\phi =: (M_1 + E^T M_2 E)^{-1} E^T M_2 E.$$

The decomposition matrix is defined such that it transforms the velocity (2.3) as

$$\begin{bmatrix} v_L \\ v_E \end{bmatrix} = S \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

with $v_1 := [\dot{x}_1; w_1]$ and $v_2 := [\dot{x}_2; w_2]$.

Using the decomposition matrix S, the dynamics of the device (2.2) and proxy (2.3) are decomposed into that of the *locked system* $(\star)_L$ and the *shape system* $(\star)_E$:

$$M_L(x)\dot{v}_L + C_L(x,\dot{x})v_L + C_{LE}v_E = T_L + F_L \tag{3.1}$$

$$M_E(x)\dot{v}_E + C_E(x,\dot{x})v_E + C_{EL}v_L = T_E + F_E, \tag{3.2}$$

where $x := [x_1; x_2] \in \mathcal{R}^6$, and $T_L, T_E \in \mathcal{R}^6$ are the controls and $F_L, F_E \in \mathcal{R}^6$ are the external forces acting on the locked $(\star)_L$ and the shape $(\star)_E$ system, respectively. The controls and forces acting on the locked and shape system have the following relations with the control forces T_1, T_2 and the external forces F_1, F_2 of the master $(\star)_1$ and the slave $(\star)_2$.

$$\begin{bmatrix} T_L + F_L \\ T_E + F_E \end{bmatrix} = S^{-T}(x_1, x_2) \begin{bmatrix} T_1 + F_1 \\ T_2 + F_2 \end{bmatrix}, \tag{3.3}$$

where the inverse of the decomposition matrix S is computed as

$$S^{-1} = \begin{bmatrix} I & \Phi \\ E & E(\Phi - I) \end{bmatrix}. \tag{3.4}$$

From (3.3), F_L and F_E have the following relation.

$$\begin{bmatrix} F_L \\ F_E \end{bmatrix} = S^{-T} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} F_1 + E^T F_2 \\ \Phi^T F_1 + (\Phi^T - I) E^T F_2 \end{bmatrix}.$$

$$\begin{bmatrix} M_L & 0 \\ 0 & M_E \end{bmatrix} =: S^{-T} \begin{bmatrix} J_1^T M_1 J_1 & 0 \\ 0 & M_2 \end{bmatrix} S^{-1},$$

$$\begin{bmatrix} C_L & C_{LE} \\ C_{EL} & C_E \end{bmatrix} =: S^{-T} \begin{bmatrix} J_1^T M_1 J_1 & 0 \\ 0 & M_2 \end{bmatrix} \frac{d}{dt}(S^{-1}) + S^{-T} \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} S^{-1}.$$

The objective is to design controls T_L, T_E for the shape system and the locked system, respectively. Then, applying the controls (3.1)-(3.2) are decoded into the control inputs $T_{1,k}, T_{2,k}$ for the device and proxy, respectively.

3.1.1 Shape System Control

The shape system control is for maintaining the coordination between device and proxy. Using the coordination error $x_E =: x_1 - x_2$, the velocity error $v_E = \dot{x}_E =$

$\dot{x}_1 - \dot{x}_2$, and the mismatched human and proxy forces F_E, the following control design is used [8].

$$T_E = -Kq_E - B\dot{q}_E - F_E + C_{EL}v_L \tag{3.5}$$

Note, the PD control portion $-Kq_E - B\dot{q}_E$ in (3.5) is analogous to the spring-damper virtual coupling force in (2.9) and (2.14).

3.1.2 Locked System Control

For the locked system control, we incorporate *dynamics augmentation control* which describes how the locked system behaves according to T_L and F_L.

We briefly explain the need for the dynamics augmentation controls for dental simulators. Each dental tool behaves differently according to its respective dynamics. For example, a dental handpiece should feel and behave differently (e.g., more resistant to external forces) from a relatively lighter dental scaler (Fig. 3.3). Also, depending on each tool, the center of mass would be at a different location. These different dynamics should be considered for haptic rendering.

For this, we replace the locked system dynamics with the following desired dynamics.

$$M_{L,d}\dot{v}_L + (C_{L,d} + D_d)v_L = F_L,$$

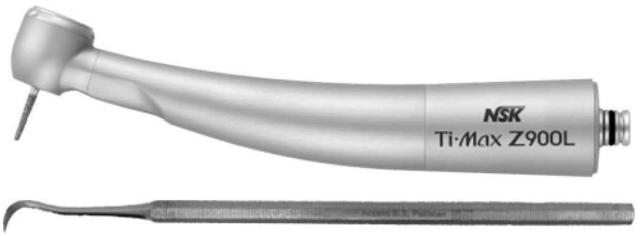


FIGURE 3.3: Dental handpiece from NSK (top) and scaler from J&J Instruments (bottom).

where $M_{L,d}, C_{L,d}, D_d \in \mathcal{R}^{6\times 6}$ are the desired inertia, Coriolis, and damping matrix, respectively, and F_L is the external force acting on the locked system.

Then, the control T_L for the locked system in (3.3) is designed as

$$T_L = C_{LE}\dot{v}_E + C_L v_L - F_L - M_L M_{L,d}^{-1}(C_{L,d}v_L + D_d v_L - F_L).$$

To maintain passivity, the desired locked system dynamics $M_{L,d}$ should have its diagonal terms similar or larger than those of the device M_1. Passivity analysis can be conducted in a similar way as in [8], where a simple scaling factor $\eta > 0$ is used instead. To check the passivity, we define Lyapunov function for the locked system as

$$V := \frac{1}{2}v_L^T M_{L,d} v_L,$$

where its time-derivative is

$$\dot{V} = -v_L^T D_d v_L + v^T F_L.$$

As a result, (3.1)-(3.2) are decoded into control inputs $T_{1,k}, T_{2,k}$ for the device and proxy, respectively, as

$$
\begin{aligned}
&T_{1,k} \\
&= (I-\Phi^T)(C_{LE}V_E + C_L V_L - F_L - M_L M_d^{-1}(D_d V_L - F_L)) - F_1 + C_{EL}V_L + T^{vc}_{1,k} \\
&T_{2,k} \\
&= E\Phi^T(C_{LE}V_E + C_L V_L - F_L - M_L M_d^{-1}(D_d V_L - F_L)) - EC_{EL}V_L + T^{vc}_{2,k},
\end{aligned}
\tag{3.6}
$$

where $T^{vc}_{1,k}, T^{vc}_{2,k} \in \mathcal{R}^6$ are the virtual coupling force for the device and the proxy, respectively.

3.2 Momentum-Based Human Force Observer

Passive decomposition necessitates measuring of human force acting on the device. Most haptic devices, however, do not have force sensors. Instead of attaching bulky and relatively expensive force sensor to the device, we estimate the human force using the momentum-based human force observer.

The momentum-based human force observer is derived from disturbance observer (DOB) which is widely used in many practical applications due to its simple structure and high performance, but cannot be directly applied to nonlinearities and couplings in the inertia matrix. By utilizing a generalized momentum, it is possible to exploit on DOB for nonlinear and coupled inertia matrix. The physical meaning of the momentum-based observer is that its outcome is the low-pass filtered value $\hat{F}_1$ of the actual human force $F_1 = J^{-T} f_1$, where $f_1 \in \mathcal{R}^6$ is the joint torque vector.

Here, we summarize the details given in [9]. From the Euler-Lagrangian dynamics of the device in (2.1), a generalized momentum $p \in \mathcal{R}^6$ is

$$p = M_1 \dot{q},$$

and its time derivative $\dot{p}$ is

$$\dot{p} = C_1^T \dot{q} + \tau_1 + f_1, \tag{3.7}$$

where the skew-symmetric relation $M_1 = C_1 + C_1^T$ is used. Then, the observer dynamics is defined as

$$\dot{\hat{p}} = C_1^T \dot{q} + \tau_1 + \hat{f}_1. \tag{3.8}$$

After applying Laplace transformation to $\dot{p} - \dot{\hat{p}}$, and defining $\hat{f}_1 \in \mathcal{R}^6$ as

$$\hat{f}_1 =: L(p - \hat{p}),$$

we obtain the estimates $\hat{f}_1$ of human force in generalized coordinates

$$\hat{f}_1 = \frac{L}{s+L} f_1,$$

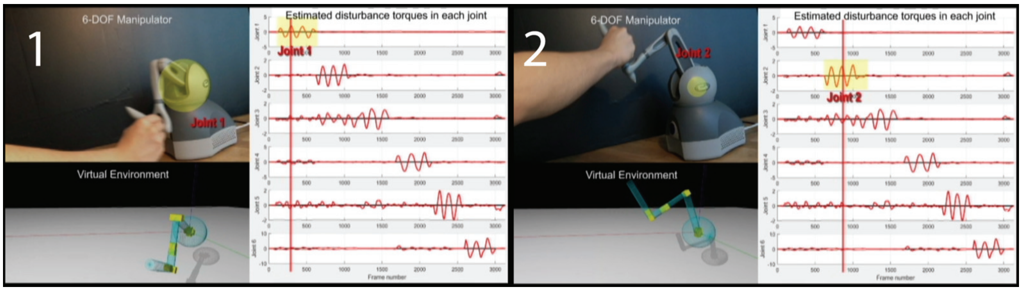
which is equivalent to the low-pass filter of the actual human force f_1.

For implementation, the following expression for $\hat{f}_1$ is used.

$$\hat{f}_1 = Lp - L \int_0^t \dot{\hat{p}} dt,$$

where $L \in \mathcal{R}^{6\times 6}$ is a diagonal gain matrix.

As a result, the human torques applied to each joint are estimated using the momentum-based human force observer (Fig. 3.4).



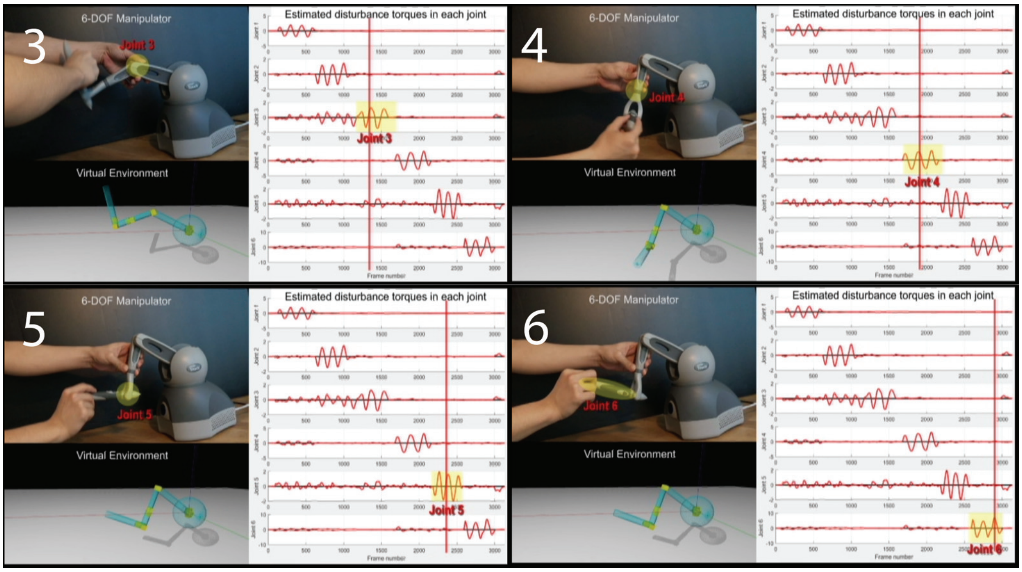


FIGURE 3.4: Estimation of human torques applied to each joint using the momentum-based human force observer.

Chapter 4

Experimental Results

4.1 Experimental Setup

The experimental setup is comprised of the simulation part (Sec. 4.1.1) and the hardware part (Sec. 4.1.2). For the graphics rendering, we use OpenGL 3.3 (Open Graphics Library) in C++.

Our haptic rendering framework for dental simulators is structured following the block diagram in Fig. 4.1.

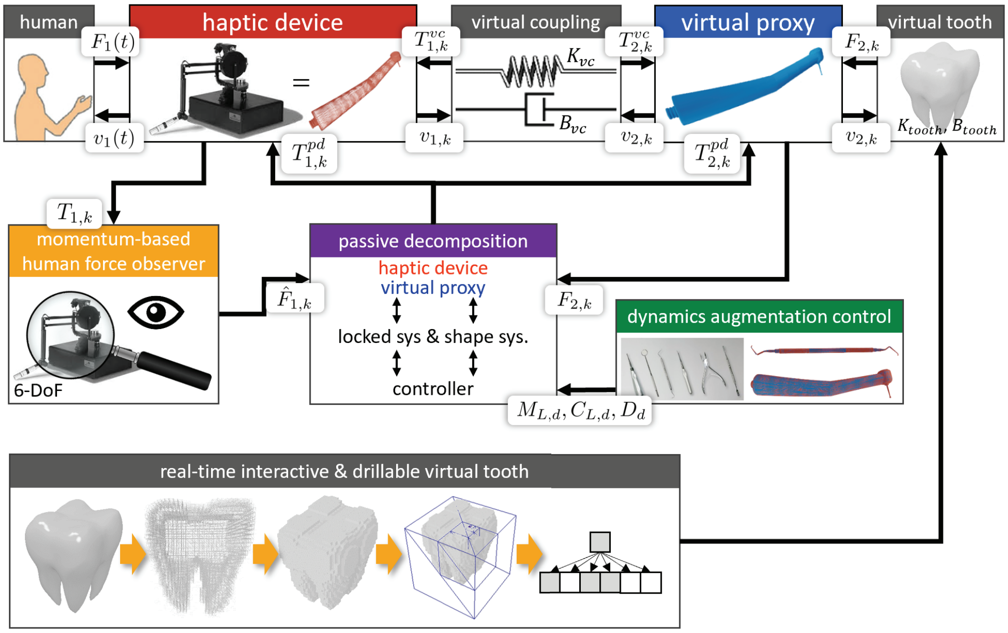


FIGURE 4.1: Block diagram of the haptic rendering framework.

4.1.1 Simulation

The simulation incorporating modeling and controls discussed in Ch. 2 - 3 are elaborated in implementation perspective.

- **Dynamics**

  The objective is to compute the control inputs to haptic device $T_{1,k}$ for haptic rendering, and the inputs to virtual proxy $T_{2,k}$ for interactive simulation. Also, the kinematics update for the position $x_{2,k}$, and the orientation $R_{2,k}$ for the virtual proxy should be computed for the simulation.

Following the PMI-based modeling in Sec. 2, we use the following dynamics for the virtual proxy.

$$\begin{aligned} m\frac{v_{k+1}-v_k}{T_k} + b_x \hat{v}_k &= T_{x,k} + F_{x,k} \\ J\frac{w_{k+1}-w_k}{T_k} + b_w \hat{w}_k &= T_{w,k} + F_{w,k}. \end{aligned} \tag{4.1}$$

(4.1) can be rewritten as

$$\left(M_2\frac{2}{T_k} + B_2\right)\hat{V}_{2,k} = M_2\frac{2}{T_k}V_{2,k} + T_{2,k} + F_{2,k}, \tag{4.2}$$

where $M_2 =: diag[m_2, J_2] \in \mathcal{R}^{6\times 6}$ is the inertia matrix, $B_2 =: diag[b_x, b_w] \in \mathcal{R}^{6\times 6}$ i s the damping matrix, $V_{2,k} =: [v_{2,k}; w_{2,k}] \in \mathcal{R}^6$ is the translational and rotational velocity vector, $\hat{V}_{2,k}$ is the representative velocity for PMI (2.2), and $T_{2,k}, F_{2,k} \in \mathcal{R}^6$ are the control inputs and external forces, respectively.

- **Virtual Coupling**

  We now give the control inputs from the virtual coupling for the haptic device $T^{vc}_{1,k}$ and the virtual proxy $T^{vc}_{2,k}$ (Sec. 2.2).

  For the haptic device,

$$T^{vc}_{1,x} = -K_{vc,x}(x_{1,k} - x_{2,k}) - B_{vc,x}(v_{1,k} - \hat{v}_{2,k-1}) \tag{4.3}$$

$$T^{vc}_{1,w} = -K_{vc,w}\left(-\frac{\phi_{12}}{2\sin(\phi_{12})}\bar{w}_{12} + \frac{T_k}{2}w_{1,k}\right) - B_{vc,w}\left(w_{1,k} - R_1^T R_2 \hat{w}_{2,k-1}\right), \tag{4.4}$$

and for the virtual proxy,

$$\begin{aligned} T^{vc}_{2,x} &= -K_{vc,x}\left(\hat{x}_{2,k} - x_{1,k}\right) - B_{vc,x}\left(\hat{v}_{2,k} - v_{1,k}\right) \\ T^{vc}_{2,w} &= -K_{vc,w}\left(-\frac{\phi_{21}}{2\sin\left(\phi_{21}\right)}\bar{w}_{21} + \frac{T_k}{2}\hat{w}_{2,k}\right) - B_{vc,w}\left(\hat{w}_{2,k} - R_2^T R_1 w_{1,k}\right). \end{aligned} \tag{4.5}$$

For the virtual proxy, combining (4.2) and (4.5) yields

$$\begin{aligned} &\left(M_2\frac{2}{T_k} + B_2 + K_{vc}\frac{T_k}{2} + B_{vc}\right)\hat{V}_{2,k} \\ &= M_2\frac{2}{T_k}V_{2,k} - K_{vc}\begin{bmatrix} x_{2,k} - x_{1,k} \\ -\frac{\phi_{12}}{2\sin\left(\phi_{21}\right)}\bar{w}_{21} \end{bmatrix} + B_{vc}\begin{bmatrix} v_{1,k} \\ R_2^T R_1 w_{1,k} \end{bmatrix}, \end{aligned} \tag{4.6}$$

where $K_{vc} =: diag[K_{vc,x}, K_{vc,w}] \in \mathcal{R}^{6\times 6}$ is the virtual coupling stiffness matrix and $B_{vc} =: diag[B_{vc,x}, B_{vc,w}] \in \mathcal{R}^{6\times 6}$ is the virtual coupling damping matrix.

- **Passive Decomposition**

  In addition to virtual coupling, we compute the control inputs from passive decomposition for increasing stiffness in haptic rendering beyond the device's physical damping (Sec. 3.1). Also, we incorporate dynamics augmentation control discussed in Ch. 4.

  We assume the user operates the haptic device with relatively slow velocity that we can ignore the Coriolis terms. Also, we use the estimation $\hat{F}_{1,k}$ of the actual human force $F_{1,k}$ from the momentum-based human force observer for passive decomposition. Then, the control inputs from passive decomposition with dynamics augmentation in (3.6) becomes

$$\begin{aligned} T^{pd}_{1,k} &= (I - \phi^T)\left(-M_L M_d^{-1}(D_d V_L - F_L)\right) - \hat{F}_{1,k} \\ T^{pd}_{2,k} &= E\phi^T\left(-M_L M_d^{-1}(D_d V_L - F_L)\right) - F_{2,k}. \end{aligned} \tag{4.7}$$

Then, we add $T^{pd}_{2,k}$ control terms to (4.6) as

$$\begin{aligned} &\left(M_2\frac{2}{T_k} + B_2 + K_{vc}\frac{T_k}{2} + B_{vc}\right)\hat{V}_{2,k} \\ &= M_2\frac{2}{T_k}V_{2,k} - K_{vc}\begin{bmatrix} x_{2,k} - x_{1,k} \\ -\frac{\phi_{12}}{2\sin(\phi_{21})}\bar{w}_{21} \end{bmatrix} + B_{vc}\begin{bmatrix} v_{1,k} \\ R_2^T R_1 w_{1,k} \end{bmatrix} + T^{pd}_{2,k}. \end{aligned} \tag{4.8}$$

- **Momentum-Based Human Force Observer**

From Sec. 3.2, the momentum-based human force observer estimates the human force acting on the device in generalized coordinates $f_{1,k}$. For implementation, the following equations are used.

$$\begin{aligned} \delta\hat{p}_k &= C^T(q)\delta q + \tau_{1,k} + \hat{f}_{1,k} \\ \hat{p}_{k+1} &= \hat{p}_k + \delta\hat{p}_k T_k \\ \hat{f}_{1,k+1} &= L\left(M(q)\delta q - \hat{p}_{k+1}\right), \end{aligned}$$

where $q, \delta q \in \mathcal{R}^6$ are the joint position and velocity of the device, respectively, and $L \in \mathcal{R}^{6\times 6}$ is a diagonal gain matrix.

The human force estimates $\hat{f}_{1,k}$ in generalized coordinates is converted into maximal coordinates $\hat{F}_{1,k}$, and is used in (4.7).

- **Penalty-Based Contact Normal Force**

  The contact force is part of the external force from the virtual environment $F_{2,k}$ on the virtual proxy. We add the PMI-based contact force model from (2.15) and (2.19) to (4.8).

$$\begin{aligned}
&\Bigg(M_2\frac{2}{T_k} + B_2 + K_{vc}\frac{T_k}{2} + B_{vc} \\
&\quad + \sum\Big(K_{tooth}\frac{T_k}{2}\begin{bmatrix} n_k n_k^T & -n_k n_k^T R_k S(r_k) \\ n_k n_k^T R_k^T S(r_k) & -R_k^T S(r_k) n_k n_k^T R_k S(r_k)\end{bmatrix} \\
&\quad + B_{tooth}\begin{bmatrix} I & -R_k S(r_k) \\ R_k^T S(r_k) & -R_k^T S(r_k) R_k S(r_k)\end{bmatrix}\Big)\Bigg)\hat{V}_{2,k} \\
&= M_2\frac{2}{T_k}V_{2,k} - K_{vc}\begin{bmatrix} x_{2,k}-x_{1,k} \\ -\frac{\phi_{12}}{2\sin(\phi_{21})}\bar{w}_{21}\end{bmatrix} + B_{vc}\begin{bmatrix} v_{1,k} \\ R_2^T R_1 w_{1,k}\end{bmatrix} + T_{2,k}^{pd} \\
&\quad + \sum\left(K_{tooth}\begin{bmatrix} I \\ R_k^T S(r_k)\end{bmatrix}(x_o - x_c)\right).
\end{aligned} \tag{4.9}$$

- **Constraint-Based Friction Force**

Similarly, the friction force terms $F_{fric,k} \in \mathcal{R}^6$ from Sec. 2.5 are added to $F_{2,k}$ for the virtual proxy.

$$F_{fric,k} = \begin{bmatrix} I \\ R_k^T S(r_k) \end{bmatrix} f_{fric,x} \tag{4.10}$$

Finally, this friction force $F_{fric,k}$ is added to the left-hand side of (4.9).

4.1.2 Hardware

Figure 4.2: HTC VIVE$^{\text{TM}}$ Virtual Reality Headset.

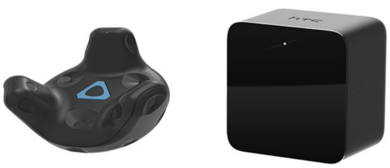


Figure 4.3: HTC VIVE$^{\text{TM}}$ Tracker (left) and Base Station (right).

For virtual reality application in dental simulation, the head-mounted display (HTC VIVE$^{\text{TM}}$ Virtual Reality Headset (Fig. 4.2)) along with global position tracker (HTC VIVE$^{\text{TM}}$ Tracker and Base Station (Fig. 4.3)) is incorporated into the system.

Figure 4.4: KaVo Super-Torque 625C handpiece (left),
and 3D Systems PhantomPremium 1.5 / 6DoF (right).

Also, for more realistic operation, a dental handpiece (KaVo Super-Torque 625C (Fig. 4.4)) with its drill part removed is attached to the handle of the haptic device (3D Systems PhantomPremium 1.5/6DoF (Fig. 4.4)) using a 3D-printed connector, as in Fig. 4.5.

FIGURE 4.5: A dental handpiece attached to the handle of the hapic device using a 3D-printed connector.

The dimensions and the representation of the connector part is give in Fig. 4.6.

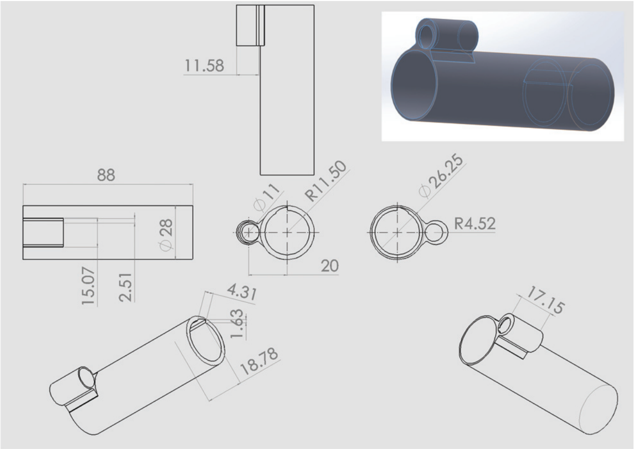


FIGURE 4.6: 3D-printed connector between handpiece and device handle.

Lastly, the palm rest (Fig. 4.7) is designed and 3D-printed to enhance dexterity and user experience. Its size is similar to that of a real human head facing up

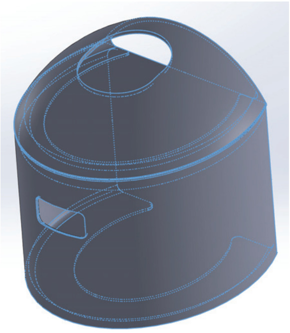


FIGURE 4.7: 3D-printed palm rest representing patient's head facing up.

with mouth open for a dental operation.

Fig. 4.8 is the final setup for our virtual reality dental simulator.

FIGURE 4.8: Virtual reality dental simulator in operation using haptic device.

4.2 Results

Recall our objective is to render the virtual tooth as stiff as it is in real world, and allow human operators to feel it realistically stiff using haptic devices.

To assess the displayed stiffness, we use Z-width computed as the ratio of total force feedback to the maximum penetration depth. By definition, Z-width is the dynamic range of impedance that can be rendered by haptic device [6]; increasing the maximum impedance is a way of increasing Z-width of haptic rendering. Larger Z-width will, in general, render *better* feeling of virtual environment, thus the *better* quality of haptic rendering [6]. For this reason, we focus on increasing Z-width of haptic rendering for our framework.

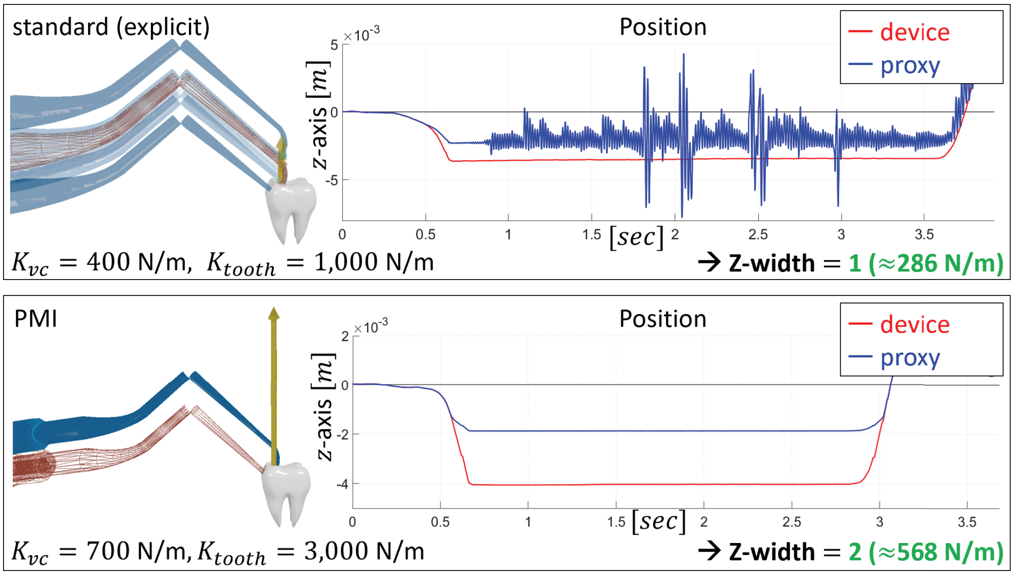


FIGURE 4.9: Z-width comparison between standard and PMI-based method.

- **Z-width Increase from PMI**

  We compare the increase of Z-width between the standard (i.e., explicit Euler integration) and PMI-based virtual coupling.

  As seen in Fig. 4.9, the PMI-based method displays a stable behavior for larger stiffness K_{vc}, K_{tooth}; whereas, the standard method is unstable even for lower gains. Z-width is computed at around 568 [N/m] for the PMI-based method, whereas below 286 [N/m] for the standard method. This implies our framework has approximately doubled Z-width by exploiting on PMI.

- **Z-width Increase from Passive Decomposition**

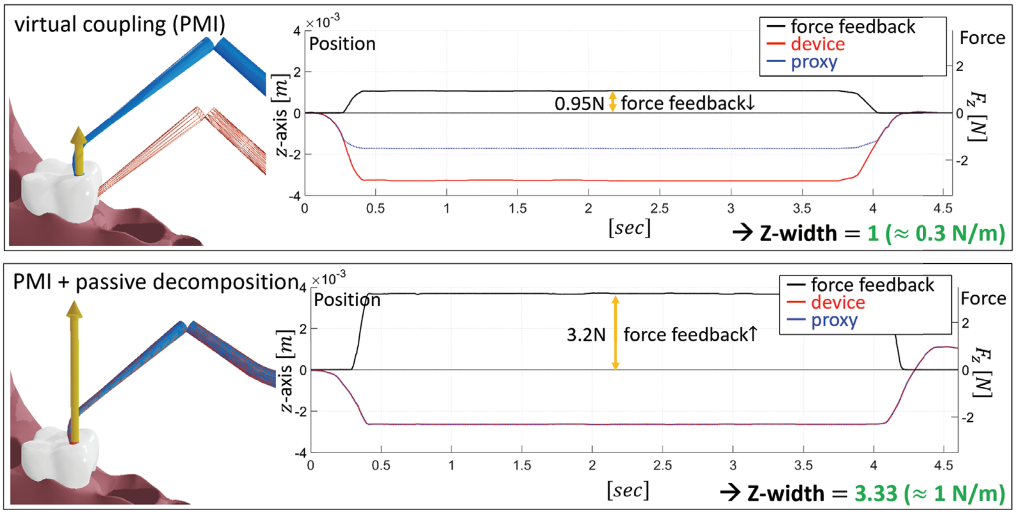


FIGURE 4.10: Z-width comparison between virtual coupling and with passive decomposition.

Next, we compare Z-width of the PMI-based virtual coupling without and with passive decomposition. As seen in Fig. 4.10, the Z-width ratio between without and with passive decomposition is 1:3.33.

- **Overall Z-width Increase**

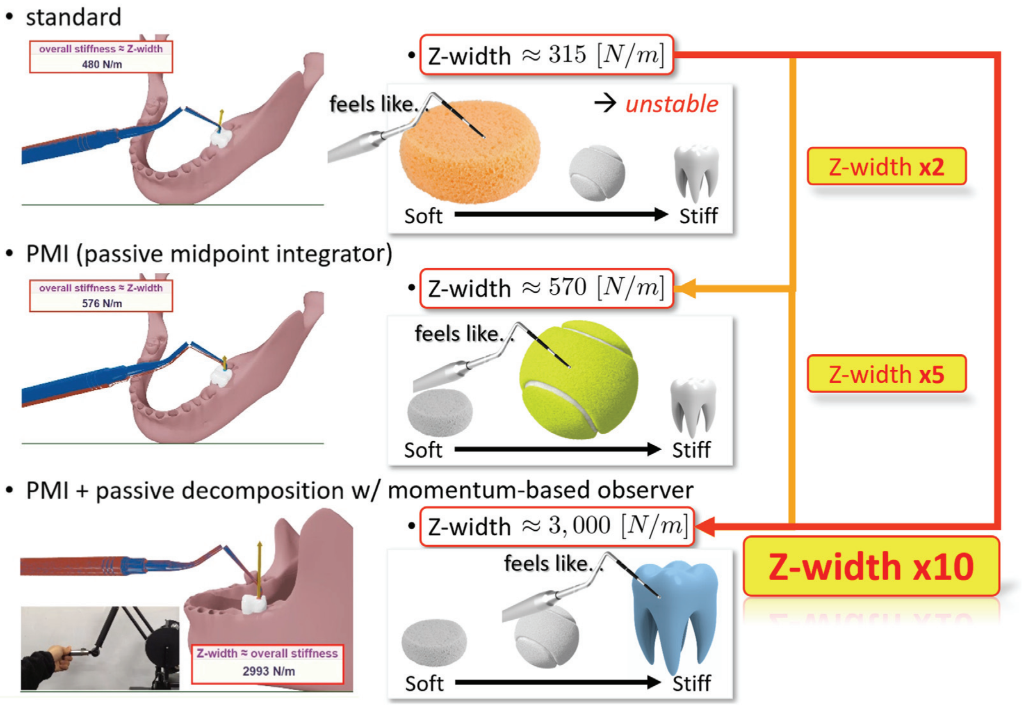


FIGURE 4.11: Summary of increase in Z-width.

Overall, the maximum achievable stiffness, yet unstable, using the standard virtual coupling (i.e., explicit Euler integration) are $K_{vc} = 400\ [N/m]$ and $K_{tooth} = 1,500\ [N/m]$, which yields Z-width below 315 $[N/m]$. On the other hand, the maximum achievable stiffness, with stability, using the PMI-based method with passive decomposition $K_{vc} = 700\ [N/m],\ K_{tooth} \rightarrow \infty$, but we set $K_{tooth} = 3,000\ [N/m]$ to match with the average stiffness of a real tooth. This yields Z-width at around $3,000\ [N/m]$, which implies that our framework has increased Z-width around ten times compared to the standard method (Fig. 4.11).

Chapter 5

Conclusion and Future Work

Conclusion

We present a novel haptic rendering framework for dental simulators that can (1) render very stiff haptic force feedback during interactions with virtual tooth, and (2) be used with general haptic devices with no force sensors and significant physical damping.

We assess the displayed stiffness by measuring the maximum achievable impedance (i.e., Z-width), and observe through experiments our framework can display around ten times larger Z-width of haptic rendering, compared to the standard explicit method (e.g., explicit Euler integration).

Large Z-width is achieved by having utilized non-iterative passive mid-point integration (PMI) for kinematics, virtual coupling, and contact forces, which allows increasing spring stiffness while maintaining system stability.

Spring stiffness of virtual coupling is upper limited due to the intrinsic and uncontrollable physical damping of haptic device, which limits further increasing the Z-width in the spring-damper virtual coupling. For this, we utilize passive decomposition, and generate stiffer haptic force feedback using feedforward control. As a result, human operators can feel the stiffness of virtual environment directly, increasing Z-width of haptic rendering significantly.

Using passive decomposition, however, requires measuring of human force applied to the device. Instead of attaching additional force sensors to the device, we derive the momentum-based human force observer to estimate the human force using generalized momentum. This observer is applicable to other multi-DoF haptic devices, allowing our framework to be used with othe general haptic devices with no force sensors.

Moreover, to render different tool dynamics for each virtual tool, we incorporate dynamics augmentation control into passive decomposition. As a result, desired dynamics can be displayed to human operators for different virtual tools.

Lastly, to maintain high update rate (i.e., above 1kHz) for haptic rendering, we exploit on full octree-based voxel rendering to simulate real-time collision detection and drilling of the virtual tooth. Small time-steps are desired for almost any kind of simulations since it makes the discrete-time virtual environment behave similar to the continuous-time real environment, which in turn allows use of large spring stiffness.

Through experiments, we observer our framework can display around ten times larger Z-width of haptic rendering, compared to the standard method. This implies the virtual tooth feels ten times more realistically stiff, making it desirable for dental simulators.

In conclusion, our haptic rendering framework for dental simulators displays increased Z-width of haptic rendering. Moreover, due to the momentum-based human force observer in generalized coordinates, our framework can be used with other multi-DoF haptic devices with no force sensors and significant physical damping.

Future Work

For practical applications, the resolution of voxels for the virtual tooth needs to be increased to improve the visualization and interactions with the tooth. This is an inherent drawback of voxel rendering (i.e., approximated surface). Also, more realistic drilling mechanism is needed for practical applications. Although vibration and friction effects have been incorporated already, improvements in contact force modeling and drilling mechanism would convey more realistic feeling of drilling.

Bibliography

[1] Wikipedia contributors. Volume rendering, 2007. URL `https://en.wikipedia.org/wiki/Volume_rendering`. File: `Figures/VolumeRendering.png`.

[2] Wikipedia contributors. Octree, 2007. URL https://en.wikipedia.org/wiki/Octree. File: Figures/Octree.png.

[3] Wikipedia contributors. Voxel, 2006. URL https://en.wikipedia.org/wiki/Voxel. File: Figures/VoxelGrid.png.

[4] Wikipedia contributors. Flood fill, year =.

[5] Gilad Ben Gal, Ervin I Weiss, Naomi Gafni, and Amitai Ziv. Preliminary assessment of faculty and student perception of a haptic virtual reality simulator for training dental manual dexterity. *Journal of dental education*, 75(4):496--504, 2011.

[6] DW Weir and JE Colgate. Stability of haptic displays. *Haptic Rendering: Foundations, Algorithms and Applications*, pages 123--156, 2008.

[7] Myungsin Kim, Yongjun Lee, Yongseok Lee, and Dongjun Lee. Haptic rendering and interactive simulation using passive midpoint integration. *The International Journal of Robotics Research*, 36(12):1341--1362, 2017.

[8] Myungsin Kim and Dongjun Lee. Improving transparency of virtual coupling for haptic interaction with human force observer. *Robotica*, 35(2):354--369, 2017.

[9] Min Jun Kim, Young Jin Park, and Wan Kyun Chung. Design of a momentum-based disturbance observer for rigid and flexible joint robots. *Intelligent Service Robotics*, 8(1):57--65, 2015.

[10] Aaron Knoll. A survey of octree volume rendering methods. In *GI, the Gesellschaft für Informatik, publishes this series in order• to make available to a broad public recent findings in informatics (ie computer science and information systems)• to document conferences that are organized in cooperation with GI and*, page 87, 2006.

[11] Andreas Petersik, Bernhard Pflesser, Ulf Tiede, Karl Heinz Höhne, and Rudolf Leuwer. Haptic volume interaction with

anatomic models at sub-voxel resolution. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, pages 66--72. IEEE, 2002.

[12] Emanuele Ruffaldi, Dan Morris, Federico Barbagli, Ken Salisbury, and Massimo Bergamasco. Voxel-based haptic rendering using implicit sphere trees. In *Haptic interfaces for virtual environment and teleoperator systems, 2008. haptics 2008. symposium on*, pages 319--325. IEEE, 2008.

[13] Online voxelizer. http://drububu.com.

[14] Wei-Wei Yu, Fei He, and Ping Xi. A rapid 3d seed-filling algorithm based on scan slice. *Computers & Graphics*, 34(4): 449--459, 2010.

요약

본 논문에서는 가상현실 치과시술 시뮬레이터의 Z-width 향상을 위한 새로운 햅틱 렌더링 프레임워크를 제시한다. 본 프레임워크는 가상 치아의 햅틱 강성도를 실제와 유사하게 렌더링 하는 것이 목표이며, 결과적으로 일반적인 햅틱 렌더링 방식보다 10배 가량 향상된 Z-width를 실험을 통해 확인할 수 있다. Z-width를 증가시키기 위해 시스템의 안정성을 향상시킬 수 있는 passive midpoint integration (PMI)를 활용하며, 햅틱 장비의 물리적 댐핑의 한계를 극복하기 위해 passive decomposition을 활용한 transparent virtual coupling 프레임워크를 활용한다. 또한, 이 때 요구되는 사람의 힘을 힘 센서 없이 예측하기 위해 momentum-based human force observer를 유도 및 사용하며, 이는 본 프레임워크가 힘 센서가 없는 여타 다자유도 햅틱 장비와 사용될 수 있게하는 핵심적인 요소이다. 이 외에도 다양한 치과시술 도구의 다이나믹스를 구현하기 위해 dynamics augmentation control을 passive decomposition에 추가하여 가상 장비들의 각각 다른 다이나믹스를 느낄 수 있게 한다. 마지막으로, 실시간 시뮬레이션을 위해 full octree 기반의 복셀 렌더링을 활용하여 가상 치아 탐색 및 치석 제거를 실시간으로 구현한다.

주요어: Z-width, Haptic rendering, Interactive simulation, Dental simulator, Passive midpoint integration, Passive decomposition, Momentum-based human force observer, Full octree-based voxel rendering, Dynamics augmentation control, Virtual reality.

학번: *2017-24482*