Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Haptic Rendering of A Puncture Task with 4-Legged 6 DOF Parallel Haptic Device

Hyung Wook Kim
Department of Electronics Engineering,
Hanyang University
Ansan, Korea
Email: hwkim@incorl.hanyang.ac.kr

Il Hong Suh and Byung-Ju Yi
School of Electrical Engineering and Computer Science
Hanyang University
Ansan, Korea
Email: {ihsuh, bj}@hanyang.ac.kr

***Abstract*— A haptic rendering system is proposed for a puncture task of a virtual vertebra model. To build a mesh model from medical images, Delaunay triangulation is applied and physical models are based on elasticity theory. Also, a redundant actuated 6 DOF parallel type haptic device is designed to display large force and to resolve the singularity problem of parallel type mechanisms. Haptic feeling of puncture task and the performance of the proposed haptic device are tested by two puncture task experiments.**

I. INTRODUCTION

Percutaneous therapies have been investigated as one effort of robotic applications to surgical procedures. Modeling of percutaneous tasks such as needle insertion, bone drill, screw, and trocar is understood crucial to provide physicians more information for successful minimally invasive surgery using many advanced information technologies such as medical images, sensors, and robot. This effort could be finally integrated into the form of haptic simulators to train surgeons. However, few puncture tasks have been investigated due to the difficulty in modeling the task and displaying large forces [9]–[14]. In [10], modeling and analysis of needle insertion forces for robot-assisted percutaneous therapy was performed. To render a puncture task haptically, following issues are considered; how to build a deformable model, how to render reflection forces, how to display visual reality, and how to design a haptic device appropriate for displaying large puncture forces.

Deformable modeling is an active research topic in virtual reality and physics-based models are necessary in many applications such as medical training systems, and surgery simulation to achieve virtual realism. Deformable models can be classified into nonphysics-based(free-form deformation) and physics-based(Finite Element Model) approaches. Free-form deformation uses control points that are manipulated to deform virtual objects [6]. Since the correlation information between movements of control points and the changes of object is required, this technique is not intuitive and not suitable for applications where physical realism is important. Finite element method is a common physics-based technique and has been used in surgical simulation to acquire a more accurate analysis of tissues and organs. Although this technique is accurate, these physics-based models are computationally intensive and not suitable for virtual reality applications that require high refresh rates for haptic rendering [7].

In this paper, Delaunay triangulation is used to generate 2D meshes and "scalable force propagation" [5] is used to reduce the computation load by confining the deformation in pre-specified regions. Instead of considering deformable objects globally, this model effectively focuses computation resources on the interacting nodes in the localized regions. Along with this force rendering algorithm, to give a visual reality for puncture task, meshes around puncture tool tip are regenerated locally by using Delaunay triangulation.

To display puncture forces, a haptic device should have the ability to exhibit force of 20N or more continuously for some time interval. Although various types of haptic devices [8], [15]–[17] have been developed, the force reflection capability of those haptic devices is confined to three or less DOF, which is not enough to mimic real puncture tasks.

Parallel mechanisms have advantages over serial mechanisms because of high structural stiffness, low inertia, and high force bandwidth. However, parallel haptic devices are bulky, have relatively small workspace, and have multiple forward kinematic solutions. Also they require large power due to floating actuators [8], [18], [19]. To overcome these difficulties, new parallel mechanisms are proposed in [20], [22]–[25], which are light by employing non–floating actuators and have relatively large workspace as compared to previous parallel mechanisms. Although parallel mechanisms have many advantages as a haptic device, it is known that parallel mechanisms have many singularities that degrade the force reflection performance in haptic applications. At singular points, most of parallel type haptic systems cannot generate the desired reflection force completely, and moreover, torque saturation happens. In this paper, to resolve the singularity problem, a new design including redundant actuation will be employed.

This paper is organized as follows; Section II shows the process of virtual model generation by using Delaunay triangulation, and the kinematics of redundant actuated 6 DOF parallel haptic device is discussed in section III. In section IV, the haptic device is employed to the puncture task of a virtual vertebra. The force reflecting experimentation was performed using the haptic simulator. Finally, we draw conclusions in section V.

0-7803-8463-6/04/$20.00 ©2004 IEEE

II. VIRTUAL MODEL GENERATION FOR PUNCTURE TASK

A. Creation of a virtual model

In order to generate a virtual model for medical use, CT image data are used. They are 24-bit color images of size 512 by 512 pixels, and are taken at 1mm interval. Fig.1 shows a high resolution image of an anatomical slice of the vertebra.

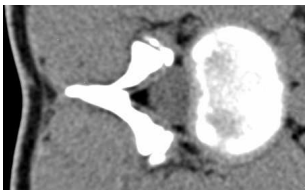


Fig. 1. CT Image of human vertebra

To extract the shape of vertebra from those images, histogram equalization technique is applied to give a better contrast between the vertebra and the surrounding organs. After contrast enhancement, a simple thresholding technique is used to get a binary image and an edge detection algorithm is applied to extract the contours of the image(Fig.2)

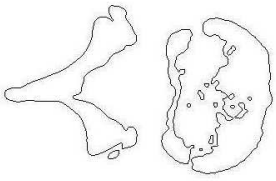


Fig. 2. The result of edge detection

Since extracted contours have too much resolution, that is, points on contours representing boundaries of the object are too close together for mesh generation, vertices and edges of contours need to be reduced. Here, the Douglas-Peucker(DP) algirithm [1] is applied to reduce the number of vertices on the contour.

The DP algorithm is widely used in the field of computer graphics and geographic information systems. It connects the two extreme endpoints of a given polyline by a straight line as the initial approximation of the polyline(Fig.3(a)). Then, all intermediate vertices are checked by computing the distances from those to that line segment. If all distances are less than a given tolerance ϵ, then the endpoints are maintained and other intermediate vertices are deleted. If any of those distances exceeds the tolerance, the point that is furthest away from the line segment is chosen as a new vertex that devides the original line into two sub-lines(Fig.3(b)). This procedure is repeated recursively on the sub-lines until all of the intermediate distances are less than ϵ as illustrated in Fig.3(c) and (d). Then all the intermediate vertices are deleted. Fig.4 shows a resulting contour when the DP algorithm is applied.

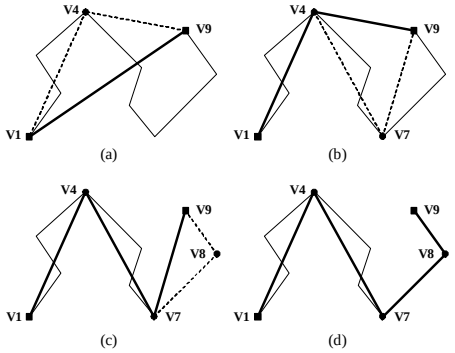


Fig. 3. Successive stages of DP algorithm

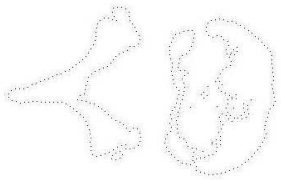


Fig. 4. Reduced points on the contour

B. 2D mesh generation using Delaunay triangulation

In this paper, Leonidas's Delaunay algorithm [2] and Ruppert's Delaunay refinement algorithm [3] are applied to generate a 2D triangular mesh. Delaunay triangulation is defined as follows: "A Delaunay triangulation of a point set is a triangulation of the point set with the property that no point in the point set falls in the interior of the circumcircle of any triangle in the triangulation. [4] "

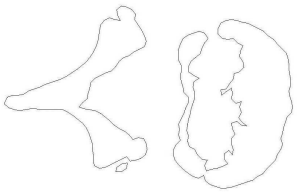


Fig. 5. PSLG of vertebra

A Planar Straight Line Graph(PSLG), which is a collection of vertices and segments is the input to the Delaunay triangulation. Basically, PSLG includes holes, concavities, and unconnected vertices. The direction of outer boundary is defined as counterclock wise and the direction of inside holes is defined as clockwise. Fig.5 shows a PSLG that defines one slice of vertebra.

The first step of Delaunay triangulation is to find Delaunay triangles of the input PSLG. At this stage, all triangles that satisfy the Delaunay criteria are found by using Leonidas's $O(n^2)$ algorithm as shown in Fig.6 and no new vertices are inserted.

The second step of algorithm is to remove unwanted triangles and to conform Delaunay triangulation before refining the triangular mesh. Unwanted triangles mean the triangles which are located outside of outer boundary or inside hall. To remove the unwanted triangles, "triangle-

eating virus [3]" is planted to infect all the unwanted triangles, starting from the triangle that are located directly to the right of the outer boundary or inside hole. Since the triangle-eating virus can penetrate into only the non-boundary edges, unwanted triangles will be queued and removed. Fig.7 illustrates a result that unwanted triangles are removed.

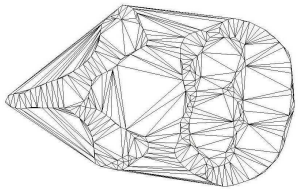


Fig. 6. Delaunay triangulation

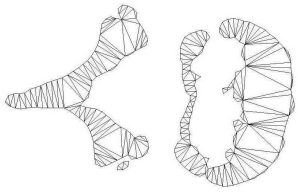


Fig. 7. Unwanted triangles are removed

To confirm Delaunay triangulation, the encroach test is performed on every PSLG segment. A segment is said to be encroached if a vertex lies in its diametral circle, which is the smallest circle containing the segment(see Fig.8(a)). An encroached segment is split by inserting a new vertex at its midpoint(see Fig.8(b)) and this process continues until all the subdevided segments pass the encroach test. The encroach test guarantees that all triangles near boundaries have circumcenters that do not cross the boundaries. A circumcenter is the center of circumcircle that is a unique circle that passes through all three vertices of the triangle.

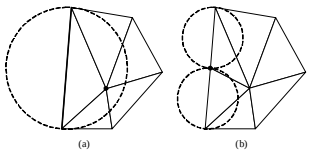


Fig. 8. Encroach test

At the third step, existing meshes are refined by inserting additional vertices into the mesh until the angle and area constraints are met. A triangle is said to be bad if it has smaller angle or larger area than the given constraints. If a triangle is bad, a new vertex is inserted at the circumcenter of the triangle and neighboring mesh are regenerated. The result of refinement process is shown in Fig.9 when the angle and area constraints are given as 30° and 200, respectively. And Fig.10 shows the result of mesh generation with 35 CT images of vertebra.

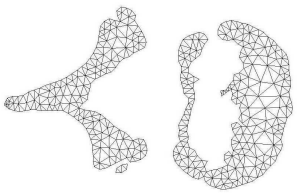


Fig. 9. Conforming Delaunay triangulation with 30° minimum angle and 200 maximum area

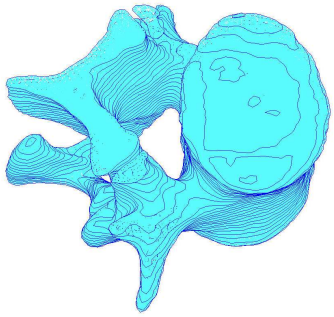


Fig. 10. The result of mesh generation with 35 vertebra slices

C. Force rendering and local mesh regeneration for puncture task

The elastic deformation model using the Hooke's law given by

$$\mathbf{F} = K_{obj} \triangle \mathbf{x} \tag{1}$$

is widely used in VR haptic feedback, because this linear equation is well suited for fast real-time force generation while allowing the modeling of object stiffness at the same time. In case of a puncture task, deformation and reflection force can be related as shown in Fig.11. While the magnitude of reflection force is less than the threshold force(F_{th}), (1)reflection force increases according to the deformation and when the reflection force exceeds F_{th}, (2)the magnitudes of reflection force and deformation go to the initial states.

Fig.12 shows the process of a puncture task. Fig.12(a) depicts the states that the tool tip approaches the surface, After the tool tip contacts the surface, surface deformation is occurred as in Fig.12(b). When the reflection force exceeds F_{th}, a hole is made and the deformed surface goes to the initial state(Fig.12(c)).

To give a visual reality for puncture task, a hole is made as soon as the force exceeds F_{th}. In Fig.13(a), a hexagon is a simplified model of tool tip for puncture. When the force exceeds F_{th}, all meshes around hexagon are regenerated locally by following procedure. The first step is to search all meshes that are across the hexagon by intersection test and all vertices inside the hexagon by encircle test. These meshes and vertices are deleted. The second step is to construct a local PSLG by using both vertices of hexagon for the tool tip(see (i) in Fig.13(a)) and vertices of edges between deleted triangles and neighboring triangles of tool tip location(see (ii) in Fig.13(a)). Then, meshes are locally regenerated by using Delaunay triangulation as shown in

Fig.13(b).

The mass-spring model is applied to render the reflection force and to model the surface deformation. In a mass-spring model, objects are discretized into mass nodes connected by springs. Although this model is easy to implement, mass-spring modeling involves laborious formulation of stiffness matrices. In this paper, a force propagation approach [5] is employed to reduce the computation load by confining the deformation in pre-specified local regions. This algorithm fits our application because the affected region of puncture task is local.

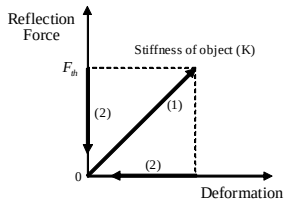


Fig. 11. The state transition of puncture task

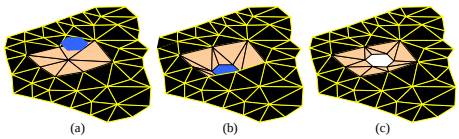


Fig. 12. Process of puncture task

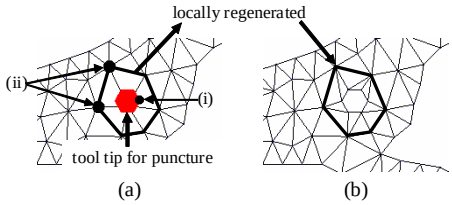


Fig. 13. Local mesh regeneration

III. 4-LEGGED 6 DOF PARALLEL HAPTIC DEVICE

Although various types of haptic device have been developed as described in Section I, the force reflection capability is confined to three or less DOF, which is not enough to display the reality in VR environment and hard contact feeling.

Parallel mechanisms have advantages over nonparallel mechanisms such as high structural stiffness and wide force bandwidth. These characteristics make parallel mechanisms fit for systems that need to display large force such as virtual spine surgery training system. Though parallel mechanisms have these advantages as a haptic device, it is known that many singularities existing in such parallel structures degrade force reflection performance. In this paper, a 4-legged parallel mechanism with redundant actuators is proposed as a means to resolve the singularity problem occurring in haptic operation and to display large reflection force [25].

A. Geometric Description

Fig.14 shows the kinematic structure of the developed 4-legged 6 DOF parallel haptic device. It consists of a top plate, eight actuators under the base, and four parallel chains connecting the top plate to the eight actuators. To avoid the interference between the motor bodies and the links, the upper actuators(M1) are placed under the base plate and the actuating torques of the upper actuators are transmitted to the first joint through a bevel gear set.

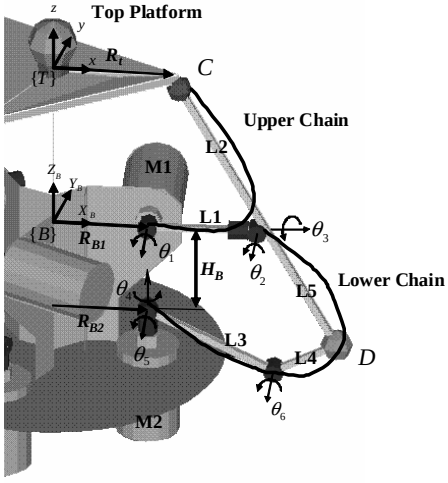


Fig. 14. Kinematic structure

Let $\{\mathrm{B}\}$ and $\{\mathrm{T}\}$ be the base frame fixed to the ground with its origin at the center of the base and the local frame fixed to the top plate with its origin at the center of the top frame, respectively. Each of the four ball-socket joints(${}^m\mathbf{C}$, m=1∼4) of the top plate is placed on the circle of radius(R_T) with 90° apart from each other. Similarly, four pairs of actuators are placed on the ground with 90° apart from each other. Each of the four actuator pairs consists of the upper actuator(M1) that is placed on the circle of radius(R_{B1}) and the lower actuator(M2) that is placed on the circle of radius(R_{B2}) vertical to the ground. And, H_B denotes the distance from the upper actuator to the lower actuator in the z-direction.

Each leg consists of a upper closed-chain and a lower closed-chain as shown in Fig. 14. The upper chains connect the upper actuators(M1) to the top plate through the ball-socket joints(${}^m\mathbf{C}$, m=1∼4). The lower chains connect the lower actuators(M2) to the upper actuators(M1) through the lower ball-socket joints(${}^m\mathbf{D}$, m=1∼4). Thus, the upper actuators support gravity loads and generate the Z directional motion and the lower actuators generate the X and Y directional motions, respectively. Interaction of the four chains generates the rotational motion of the upper plate.

The position vector of the end-effector is defined as

$$\mathbf{u} = (x_t, y_t, z_t, \theta_x, \theta_y, \theta_z)^T, \tag{2}$$

where (x_t, y_t, z_t) represents the position of the top plate's origin, and (θ_x, θ_y, θ_z) denotes $\hat{x}_t$–$\hat{y}_t$–$\hat{z}_t$ Euler angles equivalent to $[{}^T_B\mathbf{R}]$ expressed by

$$\left[{}^{T}_{B}\mathbf{R}\right] = [Rot(\hat{x}_t, \theta_x)][Rot(\hat{y}_t, \theta_y)][Rot(\hat{z}_t, \theta_z)]. \quad (3)$$

B. Forward/Inverse Kinematics

In general, parallel mechanisms have many forward kinematic solutions and our 6 DOF parallel haptic device has 8th order polynomial. Two additional encoders are placed at the passive joints of each upper chain(θ_2, θ_3) to obtain a unique forward solution.

The inverse kinematic solution of each upper chain can be calculated from the position vectors of the upper ball-socket joints(${}^{m}\mathbf{C}$, m=1∼4), which are given from the position and orientation of the top plate. In the same manner, the inverse kinematic solution of each lower chain can be obtained from position vectors of the lower ball-socket joints(${}^{m}\mathbf{D}$, m=1∼4), which are obtained from the forward kinematics of the upper chain.

C. First-order Kinematic Modeling

In the following, the first-order KIC(Kinematic Influence Coefficient) that relates the operational velocity($\dot{\mathbf{u}}$) to the active joint velocity($\dot{\theta}_A$) is described. The position vector of the four upper ball-socket joints is denoted as

$$\mathbf{C} = \left({}^{1}\mathbf{C}^T,\ {}^{2}\mathbf{C}^T,\ {}^{3}\mathbf{C}^T,\ {}^{4}\mathbf{C}^T\right)^T \quad (4)$$

with respect to the global reference frame, where ${}^{m}\mathbf{C} = ({}^{m}x_c,\ {}^{m}y_c,\ {}^{m}z_c)^T$. The position vector of the m–th upper contact point is expressed as

$${}^{m}\mathbf{C} = \mathbf{u}_t + {}^{m}\mathbf{r}_c, \quad (5)$$

where ${}^{m}\mathbf{r}_c = \left[{}^{T}_{B}\mathbf{R}\right] {}^{m}\mathbf{r}_c^{(\text{top plate})}$. Differentiating ${}^{m}\mathbf{C}$ with respect to time results in

$${}^{m}\dot{\mathbf{C}} = \dot{\mathbf{u}}_t + \omega \times {}^{m}_{c}\mathbf{r}, \quad (6)$$

where $\dot{\mathbf{u}}_t = (\dot{x}_t,\ \dot{y}_t,\ \dot{z}_t)^T$, $\omega = (\omega_x,\ \omega_y,\ \omega_z)^T$, and ${}^{m}\mathbf{r}_c = ({}^{m}r_{cx},\ {}^{m}r_{cy},\ {}^{m}r_{cz})^T$. Eq.(6) can be written in a matrix form as

$${}^{m}\dot{\mathbf{C}} = \left[{}^{m}\mathbf{G}^{c}_{u}\right]\ \dot{\mathbf{u}}, \quad (7)$$

where

$$\left[{}^{m}\mathbf{G}^{c}_{u}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & {}^{m}r_{cz} & -{}^{m}r_{cy} \\ 0 & 1 & 0 & -{}^{m}r_{cz} & 0 & {}^{m}r_{cx} \\ 0 & 0 & 1 & {}^{m}r_{cy} & -{}^{m}r_{cx} & 0 \end{bmatrix},$$

and $\dot{\mathbf{u}} = (\dot{x}_t,\ \dot{y}_t,\ \dot{z}_t,\ \omega_x,\ \omega_y,\ \omega_z)^T$.

Then, the relationship between $\dot{\mathbf{u}}$ and $\dot{\mathbf{C}}$ can be described as

$$\dot{\mathbf{C}} = \left[\mathbf{G}^{c}_{u}\right]\ \dot{\mathbf{u}}, \quad (8)$$

where

$$\left[\mathbf{G}^{c}_{u}\right] = \begin{bmatrix} \left[{}^{1}\mathbf{G}^{c}_{u}\right]^T & \left[{}^{2}\mathbf{G}^{c}_{u}\right]^T & \left[{}^{3}\mathbf{G}^{c}_{u}\right]^T & \left[{}^{4}\mathbf{G}^{c}_{u}\right]^T \end{bmatrix}^T.$$

The open chain kinematics of the m–th upper chain is described as

$${}^{m}\dot{\mathbf{C}} = \left[{}^{m}\mathbf{G}^{c}_{u\theta}\right]\ {}^{m}\dot{\theta}_u, \quad (9)$$

where ${}^{m}\dot{\theta}_u$ is the joint velocity vector of the m–th upper chain. Assuming that no singularity exists in $\left[{}^{m}\mathbf{G}^{c}_{u\theta}\right]$, the first–order inverse kinematics of (9) is obtained as

$${}^{m}\dot{\theta}_u = \left[{}^{m}\mathbf{G}^{u\theta}_{c}\right]\ {}^{m}\dot{\mathbf{C}}, \quad (10)$$

where $\left[{}^{m}\mathbf{G}^{u\theta}_{c}\right] = \left[{}^{m}\mathbf{G}^{c}_{u\theta}\right]^{-1}$.

Congregating m from 1 to 4 yields

$$\dot{\theta}_u = \left[\mathbf{G}^{u\theta}_{c}\right]\ \dot{\mathbf{C}}, \quad (11)$$

where $\dot{\theta}_u = \left([{}^{1}\dot{\theta}_u]^T\ [{}^{2}\dot{\theta}_u]^T\ [{}^{3}\dot{\theta}_u]^T\ [{}^{4}\dot{\theta}_u]^T\right)^T$ and

$$\left[\mathbf{G}^{u\theta}_{c}\right] = \begin{bmatrix} \left[{}^{1}\mathbf{G}^{u\theta}_{c}\right] & 0 & 0 & 0 \\ 0 & \left[{}^{2}\mathbf{G}^{u\theta}_{c}\right] & 0 & 0 \\ 0 & 0 & \left[{}^{3}\mathbf{G}^{u\theta}_{c}\right] & 0 \\ 0 & 0 & 0 & \left[{}^{4}\mathbf{G}^{u\theta}_{c}\right] \end{bmatrix}.$$

The velocity vector of the m–th lower ball–socket joints is described as

$${}^{m}\dot{\mathbf{D}} = \left[{}^{m}\mathbf{G}^{D}_{u\theta}\right]\ {}^{m}\dot{\theta}_u = \left[{}^{m}\mathbf{G}^{D}_{l\theta}\right]\ {}^{m}\dot{\theta}_l, \quad (12)$$

where ${}^{m}\dot{\theta}_l$ is the joint velocity vector of the m–th lower chain, and $\left[{}^{m}\mathbf{G}^{D}_{u\theta}\right]$ and $\left[{}^{m}\mathbf{G}^{D}_{l\theta}\right]$ represent the first–order KICs for the upper and the lower branch, respectively.

From (12), the first–order kinematic relationship between the upper and lower chains can be obtained as follows:

$${}^{m}\dot{\theta}_l = \left[{}^{m}\mathbf{G}^{D}_{l\theta}\right]^{-1}\left[{}^{m}\mathbf{G}^{D}_{u\theta}\right]\ {}^{m}\dot{\theta}_u = \left[{}^{m}\mathbf{G}^{l\theta}_{u\theta}\right]\ {}^{m}\dot{\theta}_u. \quad (13)$$

Note that the two active joints of the m–th chain are ${}^{m}\dot{\theta}_1$ located at the upper chain and ${}^{m}\dot{\theta}_4$ located at the lower chain. By using a row–column selection of (13), the first–order relationship between the active joint velocity(${}^{m}\dot{\theta}_A$: ${}^{m}\dot{\theta}_1$, ${}^{m}\dot{\theta}_4$, $m = 1 \sim 4$) and the joint velocity($\dot{\theta}_u$) of the upper chains can be expressed as

$$\dot{\theta}_A = \left[\mathbf{G}^{A}_{u\theta}\right]\ \dot{\theta}_u, \quad (14)$$

where

$$\left[\mathbf{G}^{A}_{u\theta}\right] = \begin{bmatrix} 1\ 0\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \left[{}^{1}\mathbf{G}^{l\theta}_{u\theta}\right]_{1;} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1\ 0\ 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \left[{}^{2}\mathbf{G}^{l\theta}_{u\theta}\right]_{1;} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1\ 0\ 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \left[{}^{3}\mathbf{G}^{l\theta}_{u\theta}\right]_{1;} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1\ 0\ 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \left[{}^{4}\mathbf{G}^{l\theta}_{u\theta}\right]_{1;} \end{bmatrix},$$

and

$$\dot{\theta}_A = \left[{}^{1}\theta_1\ {}^{1}\theta_4\ {}^{2}\theta_1\ {}^{2}\theta_4\ {}^{3}\theta_1\ {}^{3}\theta_4\ {}^{4}\theta_1\ {}^{4}\theta_4\ \right]^T.$$

Substituting (10) into (14) yields the following relationship between the velocity of upper contact points and the velocity of active joints

$$\dot{\theta_A} = \left[\mathbf{G}_c^A\right] \dot{\mathbf{C}}, \tag{15}$$

where $[\mathbf{G}_c^A] = [\mathbf{G}_{u\theta}^A][\mathbf{G}_c^{u\theta}]$.

By substituting (8) into (15), the relationship between the velocity of end–effector and the velocity of active joint is obtained as

$$\dot{\theta}_A = \left[\mathbf{G}_u^A\right] \dot{\mathbf{u}}, \tag{16}$$

where $[\mathbf{G}_u^A] = [\mathbf{G}_c^A][\mathbf{G}_u^c]$ and $[\mathbf{G}_u^A] \in \mathbf{R}^{8\times 6}$.

By the duality relation, the relationship between the operational forces and the joint torques is described as

$$\mathbf{T}_u = \left[\mathbf{G}_u^A\right]^T \mathbf{T}_A, \tag{17}$$

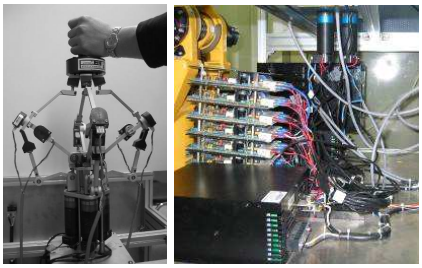
where $\mathbf{T}_u \in \mathbf{R}^6$ and $\mathbf{T}_A \in \mathbf{R}^8$ are the output force vector at the end-effector and the input torque vector of active joints, respectively. The general solution of (17) is expressed as

$$\mathbf{T}_A = \left([\mathbf{G}_u^A]^T\right)^+ \mathbf{T}_u + \left(\mathbf{I} - \left([\mathbf{G}_u^A]^T\right)^+ [\mathbf{G}_u^A]^T\right)\epsilon \tag{18}$$

where $\left([\mathbf{G}_u^A]^T\right)^+ \mathbf{T}_u$ denotes the pseudo-inverse solution of (17).

IV. Experimental Results

A haptic simulator for the puncture task of a virtual vertebra has been developed. The 6 DOF haptic device will be employed to the puncture task of a virtual vertebra. The proposed 6 DOF parallel haptic device and PC-based control system are shown in Fig.15(a) and Fig.15(b), respectively. The haptic device is driven by eight DC motors equipped with encoders. The link parameters of haptic device are as follows : $l_i(i = 1 \sim 5)$=(0.094, 0.131, 0.075, 0.075, 0.045)m. To implement a haptic device controller, a Pentium III(800MHz) computer is used. The operating system is Window NT and the control program is coded by C++ language. Control output and sensor measurement data are updated every 2msec.



(a) (b)

Fig. 15. 4 legged haptic device and control system

Fig.15(a) shows the experimental setup. A 6 DOF force sensor is mounted on the top of the haptic device and a gripper is attached on it. When the mesh generation is completed, the operator moves the gripper of the haptic device down to puncture virtual model of vertebra. The first experiment is to puncture one slice of vertebra and the second one is to puncture three successive slices of vertebra. Once the virtual tool tip contacts the slice and continues moving down, a reflection force is generated and measured by force sensor. In this paper, the stiffness of generated virtual model and F_{th} are given 5k[N/m] and 35[N], respectively. If the reflection force exceeds 35[N], a hole is made on that slice and the neighboring meshes are regenerated.

Fig.16 and Fig.17 show the result that one slice is punctured. The position of the slice is 12[cm] and the tool tip moves down from 14.8[cm]. Fig.18 and Fig.19 show the results that three slices are punctured. The positions of slices are 19[cm], 14[cm], and 10[cm], respectively. The tool tip moves down from 20.2[cm]. When the position of tool tip passes slices, the corresponding forces are generated as long as they are less than F_{th}. The reflection force is generated by the haptic device according to the particular solution in the first term of (17). If the force is over F_{th} a hole is made and the reaction force is zero. Fig.20 shows the results of three punctuated slices.

V. Conclusion

This paper presents a haptic rendering system of the puncture task, specifically for a virtual vertebra model. Some image processing techniques and Delaunay triangulation method are applied to generate virtual model. And the reflection force is modeled by using Hooke's law that is simple but efficient for real-time applications. To render large forces that are difficult for serial mechanisms to display and to resolve the singularity problem of parallel mechanism, a redundantly actuated 6 DOF parallel mechanism is proposed as a haptic device. And the performance of our haptic device that displays large force is verified by two experiments of puncture task.

Acknowledgment

This study was supported by a grant (02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, Republic of Korea. Thanks to Professor Allison Okamura for helpful comments and suggestion.

References

[1] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," The Canadian Cartographer 10(2), pp.112–122.

[2] L. J. Guibas and J. Stolfi, "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," ACM Transactions on Graphics 4(2):74–123, April 1985.

[3] J. Ruppert, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation," Journal of Algorithms 18(3):548–585, May 1995.

[4] J. R. Shewchuk, "Robust Adaptive Floating Point Geometric Predicates," Proc. of the 12th Annual Symposium on Computational Geometry, pp.141–150, May 1996.

[5] K. S. Choi, H. Sun, P. A. Heng, and J. C. Y. Cheng, "A Scalable Force Propagation Approach for Web-Based Deformable Simulation of Soft Tissues," Proc. of the 7th Int. Conf. on 3D Web technology, pp.185–193, 2002.

[6] T. W. Sederberg and R. S. Perry, "Free form deformation of solid geometric models," Annual Conference Series, Proc. of SIGGRAPH 86, pp.151–160, 1986

[7] S. Cotin and H. Delingette, "Real-time surgery simulation with haptic feedback using finite elements," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.3739–3744, 1988.

[8] R. E. Ellis, O. M. Ismaeil, and M. G. Lipsett, “Design and Evaluation of High–Performance Haptic Interface,” Robotica, Vol. 4, 1996.
[9] P. N. Brett, *et al.*, “Simulation of resistance forces acting on surgical needles,” Proc. of the Institution of Mechanical Engineers, Part H, Vol.211(H4), pp.335–347, 1997.
[10] C. Simone and A. M. Okamura, “Modeling of Needle Insertion Forces for Robot-Assisted Percutaneous Therapy,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.2085–2091, 2002.
[11] P. N. Brett and R. S. W. Stone, “A technique for measuring contact force distribution in minimally invasive surgical procedures,” Proc. of the Institution of Mechanical Engineers, Part H, Vol.211(H4), pp.309–316, 1997.
[12] V. Hayward and B. Armstrong, “A new computational model of friction applied to haptic rendering,” Lecture Notes in Control and Information Sciences, Vol.250, Springer-Verlag, pp.403–412, 2000.
[13] C. Richard, M. R. Cutkosky, and K. MacLean, “Friction Identification for Haptic Display,” Proc. of the American Society of Mechanical Engineers, Dynamic Systems and Control Division, Vol.67, pp.327–334.
[14] P. Gorman, *et al.*, “A Prototype Haptic Lumbar Puncture Simulator,” Proc. of Medicine Meets Virtual Reality 2000, pp.106–109.
[15] T. Massie and K. Salisbury, “PHANToM Haptic Interface : A Device for Probing Virtual Objects,” ASME Journal of Dynamic System and Control, pp.295–299, 1994.
[16] P. J. Berkelman, R. L. Hollis, and S. E. Salcudean, “Interacting with Virtual Environments using a Magnetic Levitation Haptic Interface,” Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp.117–122, 1995.
[17] M. Bergamasco, *et al.*, “An Arm Exoskeleton System for Teleoperation and Virtual Environments Applications,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1449–1454, 1994.
[18] M. Ishii and M. Sato, “A 3D Spatial Interface Device Using Tensed Strings,” Presence–Teleoperators and Virtual Environments, MIT Press, Vol.3, No.1, pp.81–86, 1994.
[19] J. M. Hollerbarch, “Some Current Issues in Haptic Research,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.757–762, 2000.
[20] J. H. Lee, K. S. Eom, B. –J. Yi, and I. H. Suh, “Design of A New 6–DOF Parallel Haptic Device,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.886–891, 2001.
[21] H. W. Kim, K. S. Eom, I. H. Suh, and B. –J. Yi, “A Transparency-optimized Control for a New 6–DOF Parallel-structured Haptic Device,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.2331–2336, 2001.
[22] C. M. Gosselin, J. F. Allan, and T. Lalibert, “A New Architecture for a High–performance 6-DOF Parallel Mechanism,” Proc. of 10th World Congress on the Theory of Machine and Mechanisms, pp.1140–1145, 1999.
[23] F. Pierrot, F. Marquet, C. Olivier, and T. Gil, “H4 Parallel Robot : Modeling, Design and Preliminary Experiments,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.3256–3261, 2001.
[24] L. –W. Tsai, G. C. Walsh, and R. E. Stamper, “Kinematics of a Novel Three DOF Translational Platform,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.3446–3451, 1996.
[25] H. W. Kim, J. H. Lee, B. –J. Yi, and I. H. Suh, “Singularity-free Load Distribution Algorithms for a 6 DOF Parallel Haptic Device,” Proc. of IEEE Int. Conf. on Robotics and Automation, pp.298–304, 2004.

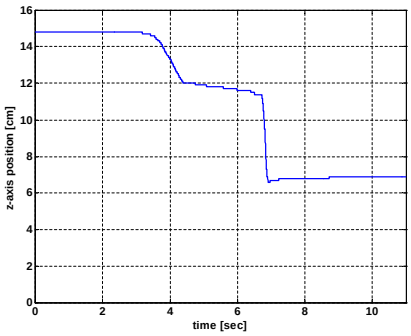


Fig. 16. Position : single slice

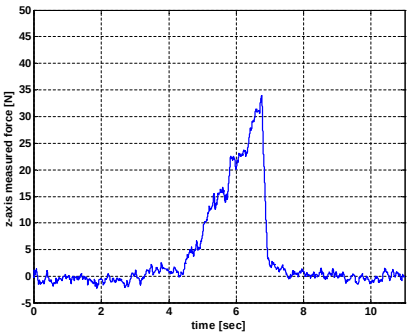


Fig. 17. Force response for single slice

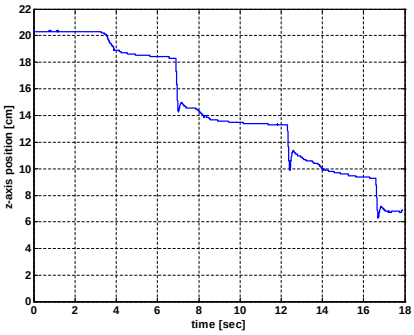


Fig. 18. Position : three slices

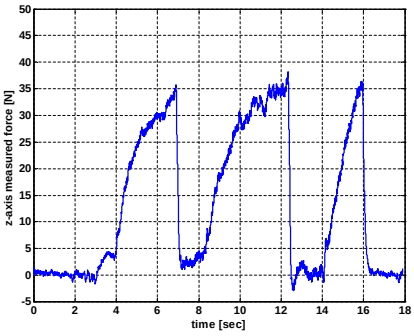


Fig. 19. Force responses for three slices

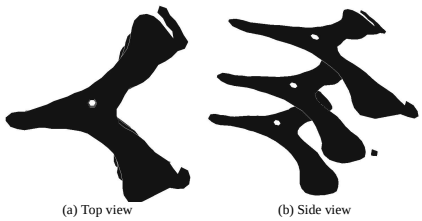


Fig. 20. Result of three punctuated slices