A Haptic Needle Manipulation Simulator for Chinese Acupuncture Learning and Training

Pheng-Ann Heng Tien-Tsin Wong Ka-Man Leung Yim-Pan Chui Hanqiu Sun

Abstract

This paper presents a haptic needle manipulation simulator for Chinese acupuncture learning and training. Students can learn and practise acupuncture in the proposed 3D interactive virtual environment that supports force feedback interface for needle insertion. So that, students not only "see" but also "touch" the virtual patient. With the high performance computers, highly informative and flexible visualization of acupuncture points of various related meridian and collateral can be highlighted to guide the students during training. Such a virtual reality system can provide an interesting and effective learning environment for Chinese acupuncture.

1 Introduction

Chinese acupuncture is a key component of traditional Chinese medicine. It is effective in treating and preventing various diseases by stimulating acupuncture points on the body. By repeated lifting and thrusting of a needle with controlled depth and frequency at the desired acupuncture point, the point can be stimulated. Guiding the needle to the destination precisely and manipulating it properly is crucial to achieve the desired therapeutic effects and this relies on correct tissue identification from tactile experience on needle.

Conventionally, students can only practise either on real patients or unrealistic artificial mannequin. Artificial mannequin provides very limited visual and unrealistic force feedback while in-vivo practice has a high risk of invasion. Wrong identification of acupuncture points or incorrect manipulation may result in dizziness, pain, internal bleeding and even long term adverse effects.

Virtual reality (VR) based systems can solve these problems by presenting an informative and comprehensive virtual human models for the identification of different acupuncture points. As VR-based system can be used repeatedly, the availability is not a problem. Besides, educational training programme can be scheduled systematically in a computerized system without risking patient's health.

This paper presents our haptic needle manipulation simulator for Chinese acupuncture learning and training using virtual reality technology. Students can learn and practise acupuncture through a comprehensive virtual human model within the virtual environment. Our system presents user with a force feedback interface for needle insertion training. As a result, users can have a tactile feeling on the virtual patient. Our system provides informative visualization of acupuncture points of various related meridian where the collateral can be highlighted to guide the students during training.

The organization of this paper is as follows. Section 2 decribes the previous system developed and techniques used in needle insertion training. Our user inteface and system overview will be outlined in Section 3 and Section 4 respectively. Section 5 describes how we perform realistic modeling of deformable bodies while Section 6 discusses our force composition in the simulation system. Finally, the skin deformation is discussed in Section 7.

2 Related Works

In the past 5 years, following the advancement in the development of haptic rendering techniques and hardware, several computer simulation systems have been developed for needle surgeries, including epidural puncture [Brett et al. 1997], spine needle biopsy [Kwon 2001] and catheter insertion [Zorcolo et al. 2000]. The needle insertion modeling described in [DiMaio and Salcudean 2002] measures the needle forces during soft tissue puncture using a robotic manipulator. The manipulator is mounted with a epidural needle, a tissue phantom and a CCD camera. In the field of Chinese acupuncture, some devices and physical models are also developed for research and training purpose.

A research group in University of Bristol has developed an experimental tactile force simulator for uniaxial epidural insertion [Brett et al. 1997]. The force resisting progress of the needle in this system is derived from measured data on porcine samples, cadavers and recently deceased cadaver specimens. However, the force data is measured from a specific surface point on the back of the body through the epidural space. Therefore, the force simulation is only unidirectional and is valid for that specific entry point and insertion trajectory only. Force sensations resulted in wrong trajectory, for example, in contact with bone, is not modeled.

Kwon has made a spine biopsy simulator consists of a force feedback device and a dummy simulating the back of human body [Kwon 2001]. While the dummy can provide an additional degree of freedom in rotational force feedback, this also poses limitation on the possible needle insertion positions and directions. Besides, only unidirectional insertion is modeled and only the needle tip is taken into account in force computations in his simulator.

Zorcolo et al. has constructed such a simulation system [Zorcolo et al. 2000]. His system is based on patient-specific volumetric medical imaging data. The force parameters are adjusted interactively. However, the force simulation is still unidirectional due to the nature of the procedure and tends to generate less realistic feedback or even undesired vibrations when the needle contact with bone.

Long Paper

3 User Interface

An intuitive user interface design can make significant contributions to a surgery training system. It can provide a comfortable and interesting learning environment for students that help reducing the learning time of the operation. Providing a set of functions that mimic the tools assisting real operations can let the student to perform their best in the training, thus increasing the usefulness and reliability of the training results. In our simulation system, a student can easily access the three main functions via the main menu: needle practice, device calibration and model settings. These functions will be described in details in the following sections.

3.1 Needle Practice

This is the main part of the simulation system. During each training session, the name of an acupuncture point randomly chosen will be displayed and student is asked to perform lifting-thrusting by a needle at that point. Students can apply the needle on the virtual body of real size lies in the center of the workspace (Figure 1(a)). The virtual needle is controlled by the 6-degree-of-freedom PHANToM stylus. A change in the needle handle's color serves as a cue for body contact (Figure 1(b)). When the student is manipulating the needle inside the virtual body by one of his hand, he can conveniently keep the needle in the current position by pressing the space on keyboard using another hand. Besides the basic needle operations, several functions also assist students in training.

3.1.1 Moving Mode

Student can grab the virtual body and change to the desired position and orientation intuitively by a virtual handle before applying needle. Axes centered in the origin of the workspace guide the placement of the body within the physical workspace of the device.

3.1.2 Acupuncture Atlas

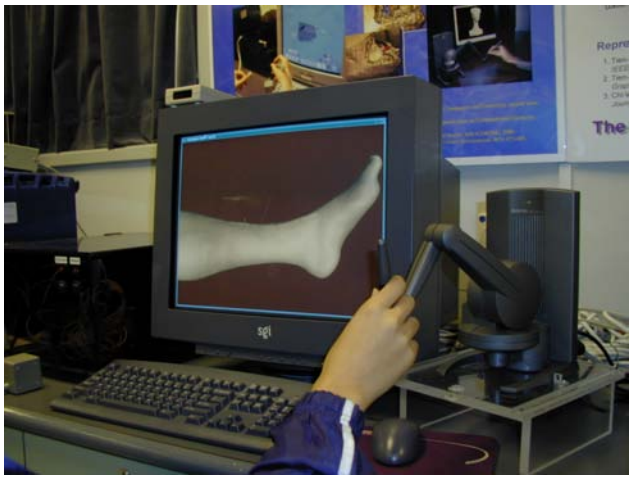
Acupuncture points can be visualized on the virtual body. Moving the needle to the acupuncture point can highlight it and check its name and corresponding meridians.

3.1.3 Training Results

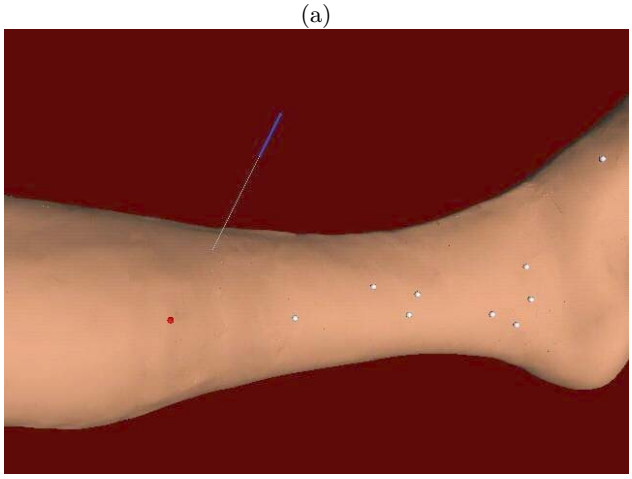
After a student remains a needle in the virtual body or takes a needle out of the body, a training session is considered to be finished. The correct acupuncture point position is shown and the student can check his performance by clicking a button. He can also choose to practice again on the same point if he thinks his previous performance is unsatisfactory.

3.2 Model Settings

Parameters controlling the haptic model can be adjusted by experienced practitioners though this interface. It is divided into eleven steps, one step for tuning one parameter. The adjustments are immediately made on the haptic model and practitioners can evaluate the result by moving the needle in the virtual body. There are detail instructions and color changes guiding through the tasks. Adjusted parameter values can be stored into data files and loaded into the simulator.



(a)



(b)

Figure 1: Overview of the visual-haptic interaction (a) The user interface (b) Sample frame rendered.

4 System Overview

A sufficiently smooth visual display requires an update rate of 30Hz while a stable haptic display requires an update rate of 1kHz. To enhance the overall performance, it is desirable to separate the haptic feedback generation system from the visualization environment. Removing the burden of force computations and haptic feedback delivery from the graphics workstation can result in an increased system performance.

The haptic feedback device, PHANToM Desktop, is connected on a remote workstation, which communicates with the graphics workstation through a network. Although an overhead in network communication is introduced, it is justified by the increased overall efficiency. Besides, careful design of algorithms and models results in loosely coupled graphics and haptic computations, which effectively minimize the communication lag. Separated renderings also enable the use of specialized computing architectures for the two interfaces and achieves separation of concerns during development.

In our simulation, a Windows NT workstation serves as the haptic server whereas a SGI Octane2 Workstation serves as the visual server. The visual server mainly updates the 3D

scene and delivers visual feedback to users while the haptic server handles force computations and communicates with the haptic device (see Figure 2). Using such architecture, our simulator achieves a visual update rate of >30Hz and haptic update rate of 1kHz.

The stylus of the PHANToM Desktop is used to simulate the Chinese acupuncture needle. The force acting on the needle is computed by integrating force components along the needle up to its tip. The whole system is supported by parallel visual and haptic rendering on two servers (see Figure 2). Implementation is based on cross-platform open-source libraries. So, together with the intuitive interface, the system is not cost-prohibitive and readily applicable in medical schools for training.

4.1 Visual Rendering Pipeline

The visual rendering part of the simulation is programmed in C++ with the aid of two libraries, Visualization Toolkit and the GIMP Toolkit. At the start of each rendering step, a request signal is sent to the haptic server asking for the haptic device's position and orientation. After the haptic server replies back the require information, it will use the information to render the virtual needle on the interface. These two messages forms the most frequent communicated data (˜30Hz) between the two servers. Since the visual server is also responsible for capturing the keyboard and mouse inputs, messages that sent to the haptic server include the notification of these commands that accept the haptic rendering.

4.2 Haptic Rendering Pipeline

The haptic rendering part is programmed in C++ with the aid of General Haptic Open Software Toolkit. Since the haptic rendering loop is running at a frequency of 1kHz, which is a lot higher than that of visual rendering, the position and orientation information of the device are not sent unless a request from the other side is received. Other messages that are sent to the visual server include the deformation of skin and performance statistics of the student.

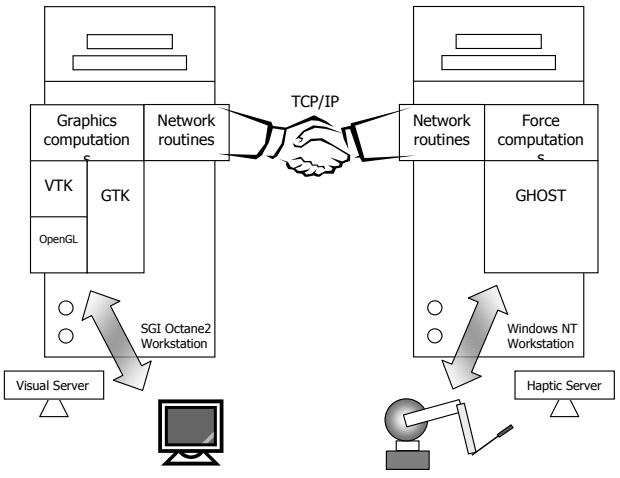


Figure 2: Overview of the visual-haptic interaction

4.3 Segmentation of Data

The virtual human body in our simulator is based on the Visible Human Data produced by the National Library of Medicine, Maryland. We are interested in the tissue information inside the Visible Man body. So, instead of directly making use of the Visible Man's raw data, we employed the segmented tissue data set HUGO from MVR Studio GmbH. Based on the HUGO classification, we make use of adaptive skeleton climbing to extract the multiple layer of interaction. We have extracted the bone, muscle and fatty tissue.

5 Haptic Model Construction

Several factors are being considered during the development of the whole needle manipulation simulator:

Realism - One of the goals in our simulation is to construct models that can demonstrate the force behaviors during the needle manipulation as close to realism as possible.

Computation - Computational complexity in simulation has to be considered. The computational requirement for haptic rendering is especially tight. To achieve a stable haptic display, an update rate of more than 1kHz is desired. Besides, the final purpose of our simulation is for acupuncture needle treatment training, having an interactive update rate is the key to improve students' performance, unlike simulation for surgery planning and scientific analysis. Since the requirements of realism and computational efficiency always conflict with each other, a trade-off has to be made between the two in the simulation design. So, like other needle simulations, in designing our haptic models, we tend to seek for simple equations that can describe the experimental results instead of computational expensive models of complex tissue behaviors.

Limitations - Haptic display interfaces are not perfectly designed. PHANToM Desktop is not an exception. The device has several limitations on force display, including maximum force limit and stiffness limit. Some artifacts such as buzzing may also results when large velocity-dependent (damping) forces or forces with bumpy direction and magnitude are displayed on the device. These have to be taken into account when designing the models, in order not to exceed its limits and cause undesirable behaviors during training.

All variables are in the following units unless stated:
Force - N (Newton); Displacement - mm; Velocity - mms^{-1}.

Notations adopted in our haptic models are as follows:
T - Set of tissue types {Air (a), Skin (k), Fat (f), Muscle (m), Bone (b)};
Ω - Set of tissue states {Pre-puncture (p), Static (s), Dynamic (d)};
x - Displacement; v - Velocity; f - Force;
$F_{\mathrm{t,s}}(x, v)$ - Force resulted from tissue $t \in T$ at state $s \in \Omega$, with, tissue deformation x and needle velocity v; e.g. $F_{\mathrm{k,p}}$ represents force resulted from skin at pre-puncture state.
k_{t} - Stiffness of tissue $t \in T$;
ρ_{t} - Viscosity (damping coefficient) of tissue $t \in T$;
Δx - Fundamental length in the incremental viscoelastic model;
H_{t} - Break limit control parameter 1 (for high velocity) of tissue $t \in T$;
L_{t} - Break limit control parameter 2 (for low velocity) of tissue $t \in T$;
S_{t} - Static friction limit of tissue $t \in T$ (per unit contact area);
D_{t} - Dynamic frictional force of tissue $t \in T$ (per unit contact area);
δ - Skin roughness [0,1);
F_{G} - Structure weight compensation force;
$F_{\mathrm{C}}(d)$ - Path constraint force;
k_{c} - Constraint spring constant;
F_{N} - Along-the-needle force;

Among the 5 principle tissue categories including air, skin, adipose tissue, muscle and bone being modeled, air is the simplest one, since no force feedback needed to be modeled

and displayed, except the compensation for device weight. The simulation of axial (along the needle) force responses for the other 4 categories will be illustrated in the sections below.

5.1 Skin

The force profile of skin when reacting with the needle can be divided into 3 states: pre-puncture, static and dynamic.

Pre-puncture State

Prior to puncture, skin exhibits a viscoelastic behavior, which can be demonstrated by a Viogt element [Findley 1976]. It consists of a linear spring with Young's modulus k and zero rest length, and a damper with viscosity ρ in parallel. A force f resulting in an extension of ϵ with velocity v can be expressed as:

$$f = k\epsilon - \rho v \tag{1}$$

However, the linear response from a Viogt element does not match with the non-linear force-deformation behavior of skin. So, the incremental viscoelastic model developed by Brett [Brett et al. 1997] is employed to simulate skin pre-puncture response. The model starts with a single Viogt element, a new Viogt element is added each time whenever the total skin deformation x exceeds a multiple of Δx, an adjustable variable. Under this model, the force required to make a deformation x ($\geq$0) on the skin before puncture is:

$$F_{\mathrm{k,p}}(x, v) = \frac{x(6\rho v\Delta x + kx^2 + 3kx\Delta x + 2k\Delta x^2)}{6\Delta x^2} \tag{2}$$

where k and ρ are the spring constant (k_{k}) and damper viscosity (ρ_{k}) respectively of skin, v is the velocity of the needle. Its profile agrees with most of the experimental studies on skin [Brett et al. 1997].

During simulation, the force to be sent to the haptic device has the same magnitude as f but in opposite direction, attempting to bring the user back to the equilibrium point (i.e. x=0). Similar rules hold in the following discussions.

The Viogt elements are not infinitely added into the model, as the skin will be ruptured (or punctured) when the force reaches a break limit. This limit is velocity-dependent according to some experiments. The faster is the needle motion, the smaller is the break limit. When the velocity tends to infinity, the break limit should tend to zero. On the other extreme, when the velocity tends to zero, the break limit should tend to infinity. The above observations led us to the modeling of the break condition as:

$$F_{\mathrm{k,p}}(x, v) \geq H_{\mathrm{k}} + \frac{L_{\mathrm{k}}}{v} \tag{3}$$

Such model is computational efficient and capable of demonstrating the above behavior. The two parameters H and L can be separately calibrated. H controls the break limit at high velocity while L controls the limit at low velocity. Setting L to zero can eliminate the velocity dependency. Now the two parameters are first set according to a few experiment results, but still rely on feedbacks from practitioners for tuning.

Overcoming the break limit triggers a change in the state of skin from "*Pre-puncture*" to "*Static*". As the needle is no longer opposed by the pre-puncture forces, but by the friction with the skin afterwards.

Static and Dynamic States

The two states are described together as both of them are closely related to friction. Friction is the result of sliding between two surfaces. In our case, the surfaces are of the needle and skin. Most interactions involving non-stiff objects demonstrate significant displacement before sliding. It is due to adhesion between the two objects and surface deformation. During pre-sliding, the needle velocity is non-zero, however, it is only deforming the tissue but not truly sliding against it. So, the two surfaces are relatively static and the tissue is said to be in static state with the needle. The tissue is in dynamic state only when the needle actually sliding against it. The force response during pre-sliding adhesion is not exactly friction, but have to be considered as part of static friction model in the simulation.

Now we first consider the dynamic state. The modeling of dynamic friction was long debated. Some models represent it as a function of velocity. However, existing haptic display devices are not good in demonstrating damping (velocity-dependent) forces, especially for large damping coefficients. Besides, the force perception resulted from these models does not appear as sliding the needle with skin but appear as moving it in viscous liquid. So we follow the Coulomb friction model, in which the frictional force always opposes the velocity and its magnitude is constant in dynamic state. This constant can be named as dynamic friction coefficient D_{k}. Skin goes into static state when the needle movement ceases.

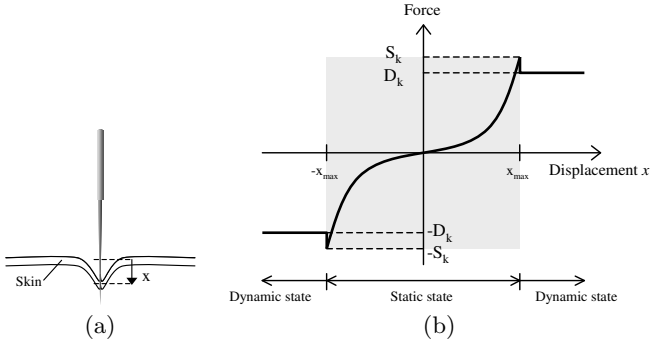


Figure 3: (a) Displacement x (b) Graphical illustration of the frictional model

The next step is to model the pre-sliding displacement in static state. Dahl was the first to study and model this behavior systematically [Dahl 1968]. Most models, including Dahl's, specify the static frictional force to be proportional to the strain, as if the two objects were connected by a spring. As pre-sliding displacement is due to the adhesion between the needle and tissue in contact, the force response should therefore follow the relationship between force and deformation of that tissue (skin in this case). So, we can again use the incremental viscoelastic model to demonstrate this phenomenon. The original equation of the model (2) derived in [Brett et al. 1997] can only handle deformation in single direction (i.e. $x \geq 0$). We extended it to handle the bi-directional needle manipulation.

Pre-sliding of the needle ends when the static frictional force exceeds a breakaway force limit (or static friction limit) S_{k} and then the skin goes into dynamic state. According to the phenomenon of stick-slip, this limit is modeled to be slightly higher than the dynamic coefficient D_{k}.

The resulting friction model is shown graphically in Figure 3. Note that the figure assumes the needle is in constant velocity and does not in contact with other tissues except the

skin of constant depth. This model displays Coulomb friction model's behavior and the stick-slip phenomenon. It can also demonstrate pre-sliding displacement with incremental viscoelastic properties.

However, the above model still has one defect. The force feedback from the model in dynamic state is overly smooth. This results in an undesired perceived experience of advancing the needle through soft plastic. Under an electron microscope, skin appears to have a jagged surface, especially for dryer skin. So in real needle manipulation, due to the needle clamping by the skin, the perceived force has a rough feature. Such feature is often smoothed out by curve fitting based on experimental results and left missing in most of the models in needle simulation. We adopt a method in which random fluctuations are added onto the original dynamic coefficient. A random quantity is generated equal likely from the range of $-D_{\mathrm{k}}\delta$ to $D_{\mathrm{k}}\delta$ with $\delta \in [0, 1)$ and then added to the dynamic friction. The roughness of skin can be adjusted by varying the parameter δ. This random function is chosen instead of others such as Gaussian distribution because it can ensure a bounded frictional force response and fast computation with satisfactory results. Mathematically, the static frictional force per unit contact area can be expressed as:

$$F_{\mathrm{k,s}}(x, v) = \frac{x(sign(x)6\rho v\Delta x + kx^2 + sign(x)3kx\Delta x + 2k\Delta x^2)}{6\Delta x^2} \quad (4)$$

with state change to dynamic when $|F_{\mathrm{k,s}}(x, v)| \geq S_{\mathrm{k}}$. The dynamic frictional force per unit contact area can be expressed as:

$$F_{\mathrm{k,d}}(x, v) = sign(v)D_{\mathrm{k}} + rand(-D_{\mathrm{k}}\delta, D_{\mathrm{k}}\delta) \quad (5)$$

with state change to static when $v = 0$. The stick-slip constraint $S_{\mathrm{k}} \geq D_{\mathrm{k}}$ always holds in the model.

5.2 Adipose Tissue

The mechanical properties of adipose tissue have been shown empirically to be almost completely viscous [Duck 1990]. A needle moving within these tissues would be opposed by a force that is very nearly proportional to its speed and contact area. Thus, the force profile of adipose tissue only has the dynamic state and the opposing force per unit contact area can be expressed as:

$$F_{\mathrm{f,d}}(x, v) = \rho_{\mathrm{f}} v \quad (6)$$

5.3 Muscle

Muscle exhibits viscoelastic behavior similar to skin. So, its force profile can be analogically divided into 3 states with models very similar to skin of different parameters. The only difference is the absence of roughness component in the dynamic friction. The model for pre-puncture state is adopted from Equation (2), while the model for static state is adopted from Equation (4). The models for dynamic state can be expressed as:

$$F_{\mathrm{m,d}}(x, v) = sign(v)D_{\mathrm{m}} \quad (7)$$

with state change to static when $v = 0$. The variable x, ρ and k refer to the corresponding parameters of muscle.

5.4 Bone

We assume the needle cannot puncture bones so we only need to consider the pre-puncture state of it. Haptic interfaces only have finite bandwidth power, which is unable to actually prevent the interface point from penetrating into the bone surface. We adopt the penalty-based method in which a linear spring with the maximum affordable stiffness is connected between the interface point and a point on the bone surface. A great force generated by the spring would attempt to repel the interface point back to the surface. A high velocity during this action may results in a large opposite force by the damping effect of other tissues in contact with the needle. This force in turn will bring the needle back into the bone and the process repeats. Finally, the device starts buzzing and become unstable. To prevent this, the viscous (damping) effect of other tissues must be disabled when the interface point is inside the bone.

Another concern is that in average the stiffness of human bone is larger than the maximum stiffness limit (3.16 N/mm) of PHANToM Desktop. We cannot reproduce the actual bone stiffness. However, since normally the stiffness of other tissues and the magnitude of the integrated force on the needle still have certain distance from the limit, that stiffness can already produce a convincing perception of striking on bone. This is approved by users of the simulator.

6 Force Composition

During passage of the needle through the various tissues, there is a combination of events, such as tissue deformation, sliding, puncture, taking place at the same time along the needle. Careful composition of forces resulting from these various events ensures the capability of simulating any possible situations in real needle manipulation. In addition to the force $\vec{F_{\mathrm{N}}}$ along the needle, there are two other components have to be considered. They are the structure weight compensation force $\vec{F_{\mathrm{G}}}$ and the path constraint force $\vec{F_{\mathrm{C}}}$ (Figure 4). The total force sending to the virtual needle tip is the sum of the three components:

$$\vec{F} = \vec{F_{\mathrm{G}}} + \vec{F_{\mathrm{C}}} + \vec{F_{\mathrm{N}}} \quad (8)$$

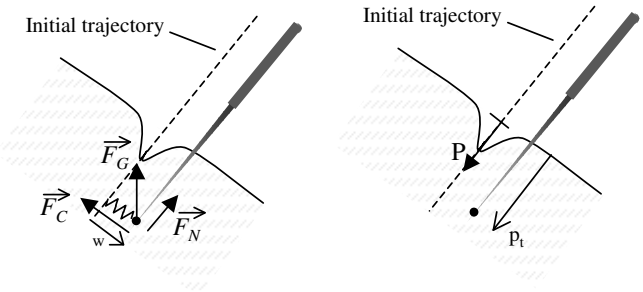


Figure 4: (a) Forces acting on the needle (b) Axis P and p_{t}

6.1 Structure Weight Compensation

The structure and actuator mass of the haptic device exert a downward force on the stylus tip even when no force is sent to the device. The virtual needle therefore seems to have an

unbalanced and overly large weight at the needle tip. The problem is even worse as the force magnitude varies with the device joint positions and orientations. The solution being adopted is to send a constant upward force $\vec{F_{\mathrm{G}}}$ to compensate the weight. Now the force is tuned to balance the weight when the stylus tip, that is the interface point, is at the center region of its workspace and the result is acceptable. This force always acts on the needle even when it has not yet been inserted into the virtual body. Further measurements will help in investigating the relationship between this weight and the tip's position.

6.2 Path Constraint Force

In real needle interactions, an inserted needle is constrained by surround tissues. We model this constraint as a linear spring connecting the needle tip to a point on the line of initial trajectory with shortest distance from the tip. The spring will bring the needle back to the initial trajectory if it is not. It can be expressed as:

$$|F_{\mathrm{C}}(\vec{w})| = -k_{\mathrm{c}} w \tag{9}$$

where k_{c} is the spring constant, w is the perpendicular displacement from the needle. This is initiated when the needle trajectory is fixed, that is, during the initial contact with the needle on the skin.

6.3 Needle Axial Force

Force along the needle $\vec{F_{\mathrm{N}}}$ starts to be computed at the moment of initial contact between the needle tip and the skin surface. This moment is identified by the collision detection of the needle with the skin surface triangular meshes. Once a collision detected, the needle's orientation and position are recorded and a line of initial trajectory is formed. The needle is assumed to follow this trajectory thereafter and the constraint force will be called into action if it deviates from the line, as mentioned before. All force computations then will be based on the tissue layers along the line of initial trajectory.

Let P be an axis along this line with origin at the original skin surface. Mathematically, the force along the needle can be computed by integrating force components from skin surface up to the needle tip:

$$|\vec{F_{\mathrm{N}}}| = \int_0^{p_{\mathrm{t}}} F_{\mathrm{t(p),s(p)}}(x(p), v)dp \tag{10}$$

where $t(p)$, $s(p)$ and $x(p)$ are the type, the state and the deformation of the tissue, p_{t} is the displacement between the needle tip and the skin surface. However, this requires interactive modification of the tissue layer depths according to their complex mutual interactions and is computational inefficient. Thus, we adopt another composition method that does not involve the adjustment of tissue layer depths but can still generate very close results.

We first define a tissue block as a group of consecutive tissue layers with the same type along the trajectory line in the HUGO dataset (Figure 5). We take advantage from tissue coherence by assuming the state and deformation of tissues within a tissue block are the same. This assumption largely simplifies the computation but does not affect the result too much.

The force computation in our simulator is performed at a frequency of 1kHz and we now describe the processes involved in a haptic iteration step that lead to the final $\vec{F_{\mathrm{N}}}$ value.

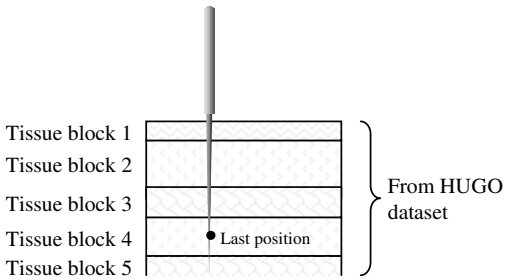


Figure 5: Tissue blocks along the needle

6.3.1 Tissue Layers Look-up

All tissue layers along the initial trajectory can be looked up from the dataset once the trajectory is determined. However, this will cause a sudden high computation load at the time of initial contact, so we average the load by performing the look-up only when necessary within each iteration step, and buffer the results in an array of tissue layers for fast retrieval. In detail, the look-up is performed only if the tissue layer at the current tip position is not yet buffered. Tissue layers in the array will not be modified once recorded. Since a voxel in the original dataset is $1\mathrm{mm}^3$ in size, the tissue layers are sampled along the line at 1mm interval. Thus, each tissue layer in the array is assumed to be 1mm thick.

6.3.2 Pre-puncture Force Computation

In our simulation, a tissue block needs to be punctured once only when the needle first encounters it, at the tissue boundary. A variable l_{p} records the index of the tissue layer that is last punctured. All tissue layers with index smaller than l_{p} have already been punctured before and will go directly into static state when contacted again by the needle. This matches with the real practice.

Besides, a needle may be simultaneously deforming more than one tissue layer before puncture. So, a pre-puncture array A is set up to record the indices of these puncturing layers. At each iteration step, new puncturing layers are first figured out and added into the array by detecting tissue boundaries in the tissue layer array, from index $\max(l_{\mathrm{t}}, l_{\mathrm{p}})$ to l_{c}, where l_{t} and l_{c} are the indices of the last and current tissue layer in which the needle tip is positioned.

Next, since we allow the needle to move backward, so some layers in the pre-puncture array may be no longer deforming the tissue. These layers should have index larger than l_{c} and is removed from the array. Then, each layer in the pre-puncture array is checked to see if it is punctured by the needle. A layer with index l of tissue type t is punctured if:

$$F_{\mathrm{t,p}}(p_{\mathrm{t}} - d_{\mathrm{l}}, v) \geq H_{\mathrm{t}} + L_{\mathrm{t}}/v \tag{11}$$

where p_{t} is the displacement of the needle tip from the skin surface, d_{l} is the accumulated depth of tissue layers with index equal to or smaller than l. Finally, the pre-puncture forces are added to the total axial force:

$$|\vec{F_{\mathrm{N}}}| = \sum_{\forall l \in A} F_{\mathrm{t(l),p}}(p_{\mathrm{t}} - d_{\mathrm{l}}, v) \tag{12}$$

where $t(l)$ is the tissue type of layer l in A.

6.3.3 Viscous Force Computation

In the current simulation, only the adipose tissue exhibits pure viscosity behavior. So, we only need to compute the depth of adipose tissue in contact with the needle from the tissue layer array. There are two cases. In the first case, the needle is not puncturing any tissue layers, so the depth of adipose tissue can be accumulated from the skin surface up to the needle tip. In another case, the needle is puncturing some tissues, although the tissue in the dataset at the needle tip position may be fat, the needle is not actually in contact with it. So, the depth of adipose tissue should be accumulated from skin surface to the first tissue layer in the pre-puncture array only. After the total depth of adipose tissue in contact with the needle d_f is found, the viscous force can be computed and added to the axial force:

$$|\vec{F_\mathrm{N}}| = |\vec{F_\mathrm{N}}| + v\rho_\mathrm{f} d_\mathrm{f} \tag{13}$$

6.3.4 Frictional Force Computation

Here, the tissue coherence assumption is employed. An array D is set up to record the indices of the first tissue layer in each tissue block under frictional (static or dynamic) effect. The deformation, depth and state of each tissue block are also stored. We define $x(b)$, $d(b)$ and $s(b)$ as the deformation, depth and state of tissue block b in contact with the needle respectively.

As described before, the condition of state change from dynamic to static is the stop of the needle motion ($v = 0$). However, the limited update rate of haptic rendering may not be able to capture this short moment, the needle may well have moved again after a pause without noticed by the program. So, instead of checking only the condition $v = 0$, we also look for a direction change. Both conditions trigger the state change.

Within each iteration step, first is to figure out new tissue blocks to include. There are two sources, one is from the just-punctured tissue blocks, and another is from checking new tissue boundaries. When a tissue block is punctured, its first tissue layer index will be added to the array D with deformation, depth and state initialized. The state of just-punctured tissue block is set as static. Another source is due to the fact that some tissue blocks may have been punctured before, so penetrating them again will not experience pre-puncture but directly go into static state. These blocks can be found out by checking tissue boundaries. There are two cases again. If $l_\mathrm{c} > l_\mathrm{p}$, we only need to check from l_t to l_p, otherwise we have to check from l_t to l_c.

Next, we check each tissue block in D for a state change from static to dynamic. A block b with tissue type t initially in static state will advance to dynamic state if the static limit is exceeded:

$$|F_{\mathrm{t,s}}(x(b), v)| > S_\mathrm{t} \text{ and } x(b)v > 0 \tag{14}$$

After the state change, tissue deformation will be reduced due to the stick-slip phenomenon. The tissue deformation thereafter is maintained by the dynamic friction, so the new deformation can be expressed as:

$$x(b) = D_\mathrm{t}/k_\mathrm{t} \tag{15}$$

Finally, if the last block in D is in dynamic state, the depth of this block has to be enlarged as the needle further advances forward into tissue of the same type. The total frictional force is than added onto the axial force:

$$|\vec{F_\mathrm{N}}| = |\vec{F_\mathrm{N}}| + \sum_{b \in D} F_{\mathrm{t}(b),\mathrm{s}(b)}(x(b), v)d(b) \tag{16}$$

where $t(b)$ is the tissue type of block b; $s(b)$ is the state of the tissue of block b. Now, the computation of axial force finished and is sent to the haptic device for display.

One thing remains unconsidered is the termination condition for the above computations. In other words, we have to determine when the needle will completely get out of the body. We have to treat it in two cases. If the needle has not punctured the skin, the needle is out of the body when $p_\mathrm{t} \leq 0$. However, if it did puncture the skin, the skin may be still clamping on the needle when $p_\mathrm{t} \leq 0$. So, we have to check whether the skin tissue block is still in the frictional array D. If not, the skin is no longer in contact with the needle and we can safely stop the computations.

Since the needle velocity varies under manual practices, it is difficult to compare the resulting force profile with standard results and perform validation. So, an offline simulator has been constructed to simulate the force profile under constant velocity and some specified tissue configuration (Figure 6).

7 Skin Deformation

Some existing deformable models such as those using finite element method can produce realistic soft tissue deformation but requires complex computation. During the needle manipulation, usually only localized skin deformation near the insertion point is involved. So, we choose not to model global body deformation but employ the local deformation model proposed by Cagatay, etc [Basdogan et al. 1998]. The original model is for deforming tissue surface by a surgical tool tip, we modified it a bit to model deformation after skin puncture also. In the model, a local region of the skin surface in the close vicinity of the intersection between the needle and skin is deformed. The deformation is achieved by translating those vertices within a certain distance (the radius of influence R) of the intersection point, along the needle direction. The magnitude of translation is determined by a second order polynomial. Let $\vec{O}$ be the intersection point, $\vec{Q}$ be a vertex on the skin surface within the radius of influence R, and $\vec{u}$ be a unit vector in the needle direction pointing towards the tip, the translated vertex $\vec{Q_\mathrm{N}}$ can be expressed as:

$$\vec{Q_\mathrm{N}} = \vec{Q} + (1 - \frac{|\vec{Q} - \vec{O}|}{R})D\vec{u} \tag{17}$$

where $D = p_\mathrm{t}$ before skin puncture while $D = x(0)$ after skin puncture, $x(0)$ is the deformation of the first tissue block in frictional array D, which always assumed to be skin.

The parameter R controls the extent and shape of deformation. This local deformation technique is computationally less expensive and produce convincing skin surface changes. Besides, the deformation model is loosely coupled with the force model, which achieves real-time rendering rates, and also facilitates the separation of two rendering pipelines. Only the variable D is required to be communicated between them.

8 Implementation and results

The resulting haptic model is capable of rendering the feedback force during the bi-directional needle movement at any positions and angles inside body. An interactive calibration interface is set up so that haptic model parameters can be tuned step-by-step simply by sliders. The generated force

profiles are physically reasonable, agree with published data and approved by practitioners (see Figure 6). After a stu-

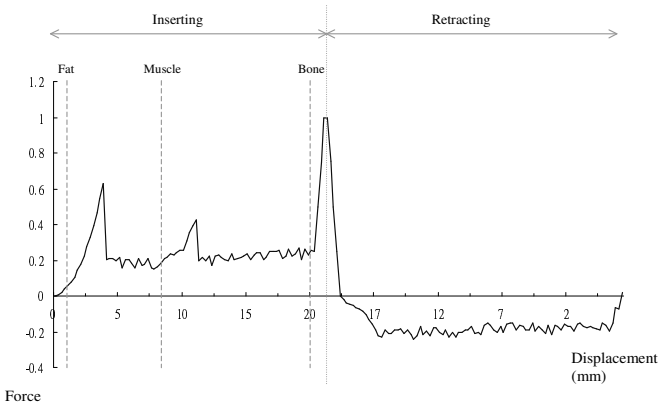


Figure 6: Force vs. displacement under our haptic model for needle insertion of constant velocity (1 mm/s)

dent has finished a training session, the precision and performance of his operation are computed and presented as a guideline of improvement.

The 3 DOF haptic response in the PhanToM Desktop does impose limitations on our system. In the current system, we cannot simulate a torque needle puncturing. In the next prototype, we shall incorporate a 6 DOF haptic device in order to construct a fully-functional simulator for Chinese acupuncture. Currently, we have adopted VHD for our simulation system. The use of the Chinese Visible Human (CVH) will fit traditional Chinese medicine study better. However, these data sets have not been segmented at the moment. Once segmented CVH results is available, we shall develop another prototype.

9 Conclusion

We have developed a haptic needle manipulating simulator for Chinese acupuncture learning and training. Our system presents user with realistic tactile feeling through a bi-directional force model.

Acknowledgment

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Central Allocation Grant Ref. No. CUHK1/00C, Earmarked Grant Ref. No. 4185/00E, and 4356/02E) .

References

Basdogan, C., Ho, C. H., Srinivasan, M. A., Small, S. D., and Dawson, S. L. 1998. Force interactions in laparoscopic simulations: Haptic rendering of soft tissues. In *Medicine Meets Virtual Reality (MMVR)*, 385–391.

Brett, P. N., Packer, T. J., Harrison, A. J., Thomas, T. A., and Carr, A. 1997. Simulation of Resistance Forces acting on Surgical Needles. In *Proc Instn Mech Engrs*, vol. 211. part H.

Dahl, P. R. 1968. A solid friction model. In *The Aerospace Corporation, El-Secundo, California*. TOR-158(3107-18).

DiMaio, S. P., and Salcudean, S. E. 2002. Needle Insertion Modelling for the Interactive Simulation of Percutaneous Procedures. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Pt. 2. 2002, Lecture Notes in Computer Science, Vol. 2489*, Springer, Tokyo, Japan, 253–260.

Duck, F. A. 1990. *Physical Properties of Tissue - A Comprehensive Reference Book*. Academic Press.

Findley, W. N. 1976. Creep and relaxation of non-linear viscoelastic materials: with an introduction to linear viscoelasticity. In *Series in Applied Mathematics and Mechanics*, vol. 18. North Holland, Amsterdam.

Hayward, V., and Armstrong, B. 1999. A new computational model of friction applied to haptic rendering. In *Preprints of ISER'99 (6th Int. Symp. On Experimental Robotics)*. Sydney, Australia.

Kwon, D. S. 2001. Realistic force reflection in a spine biopsy simulator. In *Proceedings of the IEEE International Conference on Robotics & Automation*.

Zorcolo, A., Gobbetti, E., Zanetti, G., and Tuveri, M. 2000. A Volumetric Virtual Environment for Catheter Insertion Simulation. In *6^{th} Eurographics Workshop on Virtual Environments*.