Artificial Intelligence in Medicine (2004) 32, 51—69

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

An efficient and scalable deformable model for virtual reality-based medical applications

Kup-Sze Choi*, **Hanqiu Sun**, **Pheng-Ann Heng**

Department of Computer Science and Engineering, Chinese University of Hong Kong, New Territories, Hong Kong, China

Received 1 April 2003; received in revised form 26 October 2003; accepted 17 January 2004

KEYWORDS
Virtual reality;
Deformable simulation;
Heuristic optimization;
Simulated annealing;
Haptic rendering;
Medical simulations

Summary Modeling of tissue deformation is of great importance to virtual reality (VR)-based medical simulations. Considerable effort has been dedicated to the development of interactively deformable virtual tissues. In this paper, an efficient and scalable deformable model is presented for virtual-reality-based medical applications. It considers deformation as a localized force transmittal process which is governed by algorithms based on breadth-first search (BFS). The computational speed is scalable to facilitate real-time interaction by adjusting the penetration depth. Simulated annealing (SA) algorithms are developed to optimize the model parameters by using the reference data generated with the linear static finite element method (FEM). The mechanical behavior and timing performance of the model have been evaluated. The model has been applied to simulate the typical behavior of living tissues and anisotropic materials. Integration with a haptic device has also been achieved on a generic personal computer (PC) platform. The proposed technique provides a feasible solution for VR-based medical simulations and has the potential for multi-user collaborative work in virtual environment.

1. Introduction

Despite the rapid technological advance in medicine, the effectiveness and safety of conventional medical practices have been questioned. Patients may undergo potentially hazardous medical procedures or become living subjects of interest for medical learning. For example, diagnosis using optical endoscopy may cause perforations, or even bleeding due to injury to blood vessels, or lead to complications associated with medications such as sedatives. In medical education, it has been controversial that surgical training performed on patients is risky and any errors may jeopardize their health. Substitutes such as animals or cadavers, in some occasions, may replace human body but their properties are by no means equivalent to a living human, not to mention other issues concerning cost effectiveness, availability of cadavers and animal rights. The dependence of the training on patients may also hinder the learning efficiency. Availability and variety of the cases directly influence the accumulation of hands-on experience. It takes many years for a novice to gain enough exposure before becoming experienced, or for a medical doctor to become a specialist. In addition, the learning process relies on experienced physicians to teach novices by demonstrating the craft of medical procedures, and in the meantime, conveying important details and issues verbally. This approach is indispensable in medical training but if more self-learning opportunities are also provided, the novices can

* Corresponding author. Tel.: +852-2609-8440; fax: +852-2603-5302.
E-mail address: kschoi1@cse.cuhk.edu.hk (K.-S. Choi).

0933-3657/$ — see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2004.01.013

speed up the learning curve and enhance their confidence and the ability to make independent judgement and decision.

Virtual reality (VR)-based medical simulations has emerged as a promising methodology to further improve the conventional medical practices. The goal is to allow medical procedures, diagnosis or planning to be performed in cyberspace as if carrying out in reality. VR has been widely applied to endoscopic simulations. For instance, virtual colonoscopy has enabled the interactive visualization of interior colon and polyp detection in virtual environment using computed tomography (CT) data [1,2]. Potential risk of hypoxia due to severe airway constriction during clinical bronchoscopic diagnosis can be avoided by virtual bronchoscopy [3]. Cost effective surgical training involving such operations as cutting, suturing and injection in gynecological laparoscopy has been achieved with a VR-based system [4]. Virtual training of bile duct exploration for the search of gallstones during endoscopic cholecystectomy has also been realized [5]. Performance of novices in laryngoscopic procedures has been evaluated quantitatively and compared with expert skills, thereby providing guidance for improvement [6]. In other medical applications, training of direct rectal examination for prostate cancer diagnosis has been performed by using virtual finger and prostate model [7]. In Chinese medicine, VR-based acupuncture training system has been developed recently to simulate the needle manipulation procedures with realistic force feedback [8]. Further examples also include the modeling of eyeball for surgical simulation [9,10], and the reconstruction of skull and face for planning craniofacial surgery [11,12]. Results of all these research work demonstrate the potential of applying VR technologies to improve and transform traditional practices in medicine.

Nevertheless, the requirements associated with the realization of realistic and immersive virtual environments are demanding. A major difficulty is to satisfy the two conflicting requirements—virtual realism and real-time interactions. First, virtual tissues have to respond to user's manipulations in a physically realistic manner as if possessing the true mechanical behavior of real tissues. Physical realism is determined by the accuracy of the underlying deformable model. Because of the complicated tissue mechanics, comprehensive modeling is necessarily computation-expensive. On the other hand, realistic feedbacks should be provided during the human-computer interactions in virtual environments. Visual and haptic feedback are two primary perception channels, requiring high refresh rates of 30 and 1000 Hz, respectively, for producing natural feelings [13]. Computationally efficient deformable models are thus needed to generate such feedbacks in real-time. Furthermore, hardware requirements such as visual displays for creating stereoscopic vision, force feedback devices for generating realistic sense of touch, motion tracking sensors for registering body movement, and user interfaces for mimicking real medical instruments are all important to immersive virtual environments.

The purpose of this paper is two-fold. First, an efficient physically-based deformable model is presented for the development of VR-based medical simulation system. The idea is to consider deformation microscopically as a process of force transmittal among the constituting mass points. Search algorithms are developed to determine the order of propagation and to control the scalability of the simulation. Haptic feedback is incorporated into the model with a generic personal computer (PC). Second, an artificial intelligence (AI) technique is developed to identify the parameters of the proposed deformable model. It is achieved by optimizing the parameters such that the resulting deformation approximates the reference data generated by the linear static finite element method (FEM).

1.1. Deformable models

An overview of the common deformable simulation techniques is presented in this section. It includes the geometric free-form deformation and the two widely used physics-based techniques—the mass-spring model (MSM) and the finite element method.

1.1.1. Free-form deformation

Early efforts in deformable modeling employ purely geometric techniques without considering the physical properties of materials. A typical example of this type of non-physics-based technique is the free-from deformation (FFD). It embeds deformable objects in a parallelepiped lattice of control points which are manipulated to deform the surrounding space and induce deformations on the embedded objects [14]. FFD is a tedious approach since explicit specifications of the desired deformations are always needed. Knowledge about the correlation between movements of control points and the corresponding changes in the shape of the embedded objects is required. Hence, FFD is a not suitable for VR-based medical applications.

1.1.2. Mass-spring model

In the mass-spring model, deformable objects are discretized into a system of mass points connected by springs. Deformation is simulated by the dynamics of the mass points and springs. This approach is easy to implement but involves the formulation of stiffness matrices. For an objected modeled by a mass-spring system containing n nodes, the displacement $\boldsymbol{u}^{t+h}$ and velocity $\boldsymbol{v}^{t+h}$ of the nodes at the next time-step $t+h$ are evaluated by numerical integration using the backward Euler method as follows [15]:

$$\begin{pmatrix} \boldsymbol{u}^{t+h} \\ \boldsymbol{v}^{t+h} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}^{t} + \Delta\boldsymbol{u}^{t+h} \\ \boldsymbol{v}^{t} + \Delta\boldsymbol{v}^{t+h} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}^{t} + h\boldsymbol{v}^{t+h} \\ \boldsymbol{v}^{t} + h\boldsymbol{M}^{-1}\boldsymbol{f}^{t+h} \end{pmatrix} \tag{1}$$

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_n \end{pmatrix}, \quad \boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_n \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_n \end{pmatrix},$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 & & & 0 \\ & \boldsymbol{M}_2 & & \\ & & \ddots & \\ 0 & & & \boldsymbol{M}_n \end{bmatrix}, \quad \boldsymbol{M}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \tag{2}$$

where h is the simulation time-step, the position, velocity, force and mass of the node i are denoted by $\boldsymbol{u}_i, \boldsymbol{v}_i, \boldsymbol{f}_i$ and m_i, respectively. The vectors $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{f}$ are of size $3n \times 1$, and the mass matrix of the system $\boldsymbol{M}$ is a $3n \times 3n$ matrix. The simulated shape of the deformed object is thus represented by the updated nodal displacement $\boldsymbol{u}^{t+h}$. It is evident from Eq. (1) that the determination of $\boldsymbol{u}^{t+h}$ is reduced to the evaluation of the nodal velocity change $\Delta\boldsymbol{v}^{t+h} = h\boldsymbol{M}^{-1}\boldsymbol{f}^{t+h}$, where the force vector at the next time-step $\boldsymbol{f}^{t+h}$ is approximated by using the Taylor series expansion [15]. Hence, to simulate the deformation of the system, it is necessary to solve the matrix equation for $\Delta\boldsymbol{v}^{t+h}$ first. It involves the calculation of the inverse matrix, which is usually computation-intensive and calls for numerical techniques such as the conjugate gradient method to reduce the number of operations involved. On the other hand, the system of differential equations that govern the model dynamics should satisfy some criterions [4] in order to prevent from numerical instability.

1.1.3. Finite element method

The idea of applying the finite element method for deformable simulation is to model the mechanical properties, namely the stress—strain relationship, of an object by decomposing it into a number of elements. The deformation of the entire object is then represented by the displacement of the nodes in each individual element. Interpolation of the nodal displacements using shape functions is performed to model the continuous variation in displacement within each element. While being relatively more accurate, FEM is computationally intensive and therefore not suitable for interactive applications although some mathematical techniques such as condensation [16] and modal analysis [17] have been proposed to accelerate numerical computations. Further discussion on the finite element will be given later in Section 3.2.2.

In this paper, a mass-spring-based deformable simulation system is presented. The system is based on the *force transmittal mechanism* which considers deformation as a force propagation process among the mass points in an ordered manner [18]. It avoids the laborious matrix formulations in conventional MSM and the computation is scalable simply by controlling the extent of force propagation. A similar approach has been presented by Brown et al. [19], in which a quasi-static algorithm has been also presented to simplify the computations when the deformation process is so slow that the system can be assumed to be in static equilibrium at each instant. Whenever this assumption is invalid, it reverts to the classical numerical integration techniques.

1.2. AI in VR-based medical applications

The development of VR-based medical application has been benefited from the artificial intelligence techniques. In a recent work by Radetzky and Nürnberger [20], a patient-specific surgical simulator has been developed. The virtual tissue models are generated from the magnetic resonance imaging (MRI) datasets of patients and the tissue mechanics are simulated by a viscoelastic deformable model. AI comes into play in the identification of the parameters of the deformable model. An artificial neural network model is utilized to approximate the underlying mass-spring model for real tissues. A neuro-fuzzy system has been developed to enable the definition of the model parameters based on a linguistic description to the tissue characteristics. The developed learning algorithms also enable the system to learn the deformation behavior from the measured data of real tissues or reference data generated by a physically more accurate deformable model.

Besides, a novel VR-based surgical planning system has been proposed by Sørenson et al. [21]. It enables pre-operative planning of cardiac interventions to be performed in a highly immersive virtual environment with the aid of electromagnetic tracking devices and stereoscopic displays. To create three-dimensional heart models, MRI datasets are segmented using a two-phase segmentation technique with balloons and snakes. A contour-matching algorithm is then applied to search for the similar contours from the neighboring slices, followed by the minimal weight triangulation (MWT) algorithm.

In our work, the force transmittal mechanism employs the breadth-first search (BFS) algorithm to identify the nodes to be involved in the deformable simulation, and also to determine the order of force transmittal among the nodes. Furthermore, a simulated annealing (SA) algorithm has been developed to identify the parameters of the deformable model by heuristic optimization.

2. Modeling of soft tissues

Realism of deformable tissue simulation depends on the accuracy of the underlying tissue models. Ideally, it should enable the computer-generated organs to reproduce the biomechanical behaviors of real tissues. Unfortunately, the complexity of tissues makes accurate modeling prohibitive. Microscopically, human tissues are composed of various biomaterials with different compositions and arranged into different structures, thereby exhibiting complicated mechanical characteristics when studied from a macroscopic perspective. Given the intricacies, attention is focused on linear elasticity which is the dominant characteristic of tissues when subjected to small deformations.

2.1. Mechanic properties of human tissues

Human tissues are viscoelastic in general, exhibiting both elastic and viscous behavior when being deformed, although their mechanical properties may be further complicated by non-linearity, anisotropy, plasticity, compressibility or inhomogeneity due to compositional and structural variations. While purely elastic material demonstrates simple linear stress—strain relationship (also known as the constitutive law), mechanical properties of viscoelastic material is time-dependent, resulting in three phenomena, namely *creep*, *relaxation* and *hysteresis* [22]. Creep is the increase in strain with time under constant stress, relaxation is the decrease in stress under constant strain, and hysteresis refers to the scenario when load is applied to and then removed from a material, the stress—strain curve follows two different paths and forms a loop. Nevertheless, human tissues can be approximated as linear elastic material during small deformation.

Elastin and collagen are two important constituents in living tissues. Elastin is a dominant component of ligamentum nuchae and ligamentum flavum. It is also found in arteries, veins, aorta, lung and skin. Elastin is almost a purely linear elastic material except the presence of small hysteresis loop in the stress—strain curve. Collagen is a fibrous connective tissue that is present as a basic structural element in human tissues and organs. It dominates the mechanical behaviors of tendon and fascia, and also exists in bone, cartilage and skin. Young's modulus of collagen is about 10^3 times greater than that of elastin, and thus less extensible [22].

2.2. Physical models

Rheological models have been used to depict the time-dependent viscoelastic behaviors of tissues. In these models, discrete mechanical devices such as springs, dampers and sliders serve as ''proxy'' to describe the behaviors. To model linear elasticity, different combinations of linear springs and viscous dampers are employed. The linear springs obey Hooke's law, whereas the viscous dampers generate a resistance force proportional to velocity. Fig. 1 shows three conventional spring-damper configurations—Maxwell, Voigt and Kelvin model. In practice, tissue is discretized appropriately into a network of mass points (nodes) linked by springs and dampers. These models are essentially discrete, which reduce a three-dimensional (3D) model to a network of mass points connected with one-dimensional (1D) linear springs and dampers. An obvious advantage of this type of model is that the mathematical formulation is relatively simple, despite being less accurate.

A superior approach for tissue modeling is to employ continuum mechanics which considers tissues as a continuum entity in the stress—strain analysis. For a linear elastic material, the constitutive

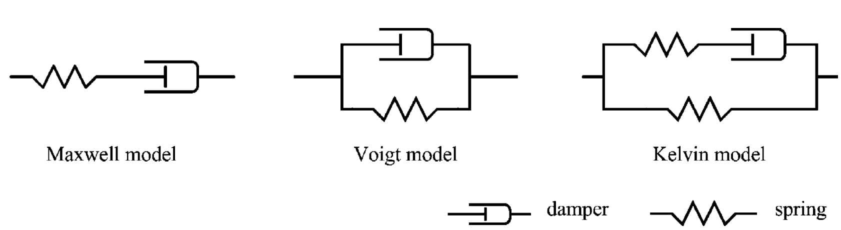

Figure 1 Three conventional rheological models for analyzing viscoelasticity.

law can be expressed using the generalized 3D Hooke's law in tensor form [23].

$$\sigma_{ij} = \sum_{k,l} C_{ijkl}\varepsilon_{kl} \tag{3}$$

In this equation, σ_{ij} and ε_{kl} are the stress and strain tensors of rank two, where subscript i, j, k and l take the values of 1, 2 or 3, denoting x, y and z axis, respectively. The stress tensor $\boldsymbol{\sigma}$ contains six elements, representing tensile and shear stress components associated with the three axes. Similarly, the six elements in strain tensor ε are the corresponding tensile and shear strain components. Material stiffness is defined by the fourth-rank elasticity tensor C_{ijkl} which contains 36 elements, but for linear isotropic material, it depends on two parameters only, namely Young's modulus and Poisson's ratio. Based on Eq. (3), deformation can be calculated numerically using FEM.

Modeling of the viscoelastic behavior is mathematically more involved. Owing to the history dependence of the mechanical effects, the stress—strain relationship of viscoelastic material is described by a time-dependent tensor equation. By the Boltzmann superposition principle [24], the constitutive law for a linear viscoelastic tissue is expressed as

$$\sigma_{ij}(t) = \int_{-\infty}^{t} C_{ijkl}(t-\tau)\frac{\mathrm{d}\varepsilon_{kl}(\tau)}{\mathrm{d}\tau}\,\mathrm{d}\tau \tag{4}$$

However, in order to reduce computation complexity in real-time interactive simulations, it is generally accepted, as a first-order approximation, to regard viscoelastic tissues as linear elastic solids for the rationale that linear elasticity dominates when tissues are subjected to quasi-static loading or undergo small deformation.

2.3. Geometrical models

Geometrical models of virtual anatomical parts are obtainable in several ways. One approach is to create patient-specific virtual 3D models of the tissues or organs from volumetric CT or MRI datasets [25]. Surface rendering techniques such as the classical marching cubes method [26], or the alternative adaptive surface climbing algorithm [27] are applied to construct *iso*-surfaces from the raw volume data. Fig. 2 shows a surface model of bronchia, which is generated from the volumetric MRI lung data using adaptive surface climbing.

Another way is to derive from the digitized axial CT/MRI datasets of human cadaver, e.g. the visible human dataset [28]. Virtual anatomical models may also be created directly by 3D computer-aided design tools in the absence of medical data.

3. Scalable deformable simulation

3.1. The force transmittal mechanism

The idea of the force transmittal mechanism is to model deformation as a process where forces are transmitted to individual mass points in a mass-spring network. Force exerted on a node first propagates via springs to its nearest neighbors, then induce secondary forces to the next nearest neighbors, and so on until all nodes in the mass-spring network are visited. Mass points are displaced successively during the process of force transmittal, which results in global deformation. The transmittal mechanism for small deformation is however localized to the neighborhood of the node in direct contact with external force, whereas the effect on nodes farther away is insignificant and thus negligible.

3.1.1. Formulations of mass-spring interactions

Consider a deformable object modeled as a network of linked mass nodes. Based on the Voigt's model, each link comprises of a spring and a damper connected in parallel. The displacement $\boldsymbol{u}_i$ of a node i

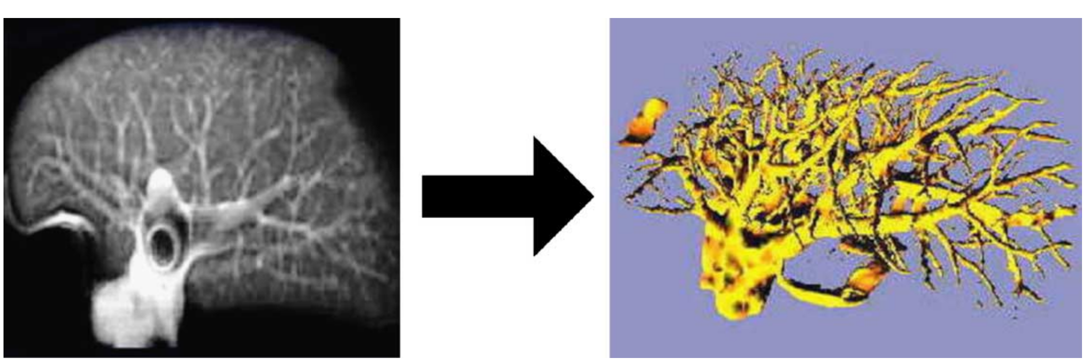

Figure 2 Generation of surface model of bronchia from MRI dataset using adaptive surface climbing.

due to its neighboring nodes j and the external force $\boldsymbol{F}_i$ is governed by the Newtonian mechanics with the following equation:

$$m_i\ddot{\boldsymbol{u}}_i + d_i\dot{\boldsymbol{u}}_i + \sum_j \boldsymbol{f}_{ij} = \boldsymbol{F}_i \tag{5}$$

with

$$\boldsymbol{f}_{ij} = k_{ij}(|\boldsymbol{r}_{ij}| - l_{ij})\frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|} \tag{6}$$

where m_i and d_i, are, respectively, the mass and damping constant of node i, $\boldsymbol{r}_{ij}$ and $\boldsymbol{f}_{ij}$ are the vector distance and the spring force between node i and j, l_{ij} and k_{ij} are, respectively, the natural length and stiffness of spring connecting node i and j. Eq. (5) can be solved numerically using the central difference method using Eqs. (7) and (8). As time advances by a time-step h, the displacement of node i at time $t+h$, $\boldsymbol{u}_i^{t+h}$ is readily obtained from its current and previous displacements $\boldsymbol{u}_i^t$ and $\boldsymbol{u}_i^{t-h}$ as follows:

$$\dot{\boldsymbol{u}}_i^t \approx \frac{\boldsymbol{u}_i^{t+h} - \boldsymbol{u}_i^{t-h}}{2h} \tag{7}$$

$$\ddot{\boldsymbol{u}}_i^t \approx \frac{\boldsymbol{u}_i^{t+h} - 2\boldsymbol{u}_i^t + \boldsymbol{u}_i^{t-h}}{h^2} \tag{8}$$

3.1.2. BFS-assisted force propagation

To realize the proposed force transmittal mechanism, algorithms based on the breadth-first search have been developed to determine the correct order of force transmittal among the mass points in localized deformation regions. When only one mass point is subjected to external force, the order is similar to that of the breadth-first traversal, which begins from the contact point to all its immediate neighbors, followed by the neighbors of neighbors, until the maximum depth of traversal (i.e. penetration depth) that limits the size of the deformation region is reached. For example, consider a two-dimensional (2D) mass-spring network with the topology shown in Fig. 3, each node is connected with six neighboring nodes. For deformation localized by a maximum depth of traversal of two layers, the order of force transmittal given by breadth-first search is illustrated in Fig. 3. During the searching process, a node that has been visited previously will not be re-visited. Deformable simulation is carried out by first evaluating the updated position of the contact point, followed by the position update of the six immediate neighbors at a topological distance $d = 1$ in layer L_1, and finally the twelve nodes at $d = 2$ in layer L_2. Note that while the discussion is based on 2D mass-spring network, this approach is equally applicable to 3D volumetric networks.

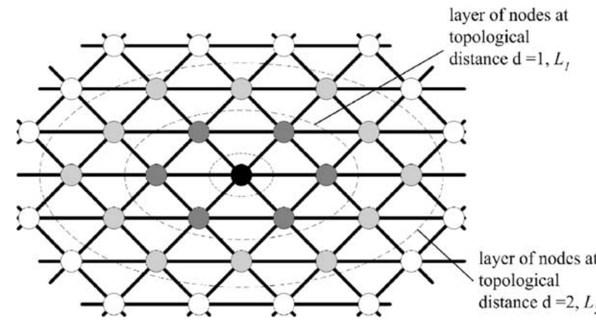

Figure 3 Order of force transmittal in a 2D mass-spring network. Force applied to the black node at the center is transmitted radially outwards to neighboring nodes via springs.

3.1.3. Optimum penetration depth

The penetration depth in the deformable model controls the number of nodes involved in the deformable simulation. The choice of the optimum penetration depth is a trade-off between visual realism and computation complexity. While a small penetration depth would lead to over-localized deformation and unnatural visual effects, a deep penetration would also result in intensive computation. To estimate the optimum depth, simulations are performed at various penetration depths. By monitoring the incremental changes in the shapes of the deformed object with respect to the increasing penetration depth, it is anticipated that no noticeable changes in shape can be detected as the penetration depth is further increased beyond a certain value, which is the optimum depth. The estimation process is automated by setting a threshold such that, when the changes in shape due to an increase in the penetration depth by one layer exceeds the threshold, the deformable simulation is repeated by further increasing in the penetration by one more layer. The process continues until the change is below the threshold, i.e. the incremental change is hardly noticeable. The penetration depth at which the calibration process ceases is the optimum depth. The experiments designed for the estimation of the optimum penetration depth will be described in Section 4.1.1.

3.1.4. Intra-layer propagation order

In Section 3.1.2, it has been assumed that the order of nodes at the same topological distance, i.e. the intra-layer propagation order, has negligible effect on deformation. In principle, the forces due to the nodes in one layer would exert on all the nodes in the next layer simultaneously. However, this would revert to the matrix formulations in conventional

MSM. The simultaneous propagation process is therefore modeled as a process where the forces propagate successively to each of the nodes in the same layer. The intra-layer propagation order is however disregarded. It simply follows the order of node traversal in BFS. Nevertheless, even if the forces propagate among the nodes in the same layer in an arbitrary order, as long as the whole process takes place in one simulation time-step, it can be regarded as being occurred in a simultaneous manner. This hypothesis has been verified by simulating the deformation with different intra-layer propagation orders. The experimental results have indicated that the resulting differences in the nodal positions of the deformed object are insignificant, suggesting the order of propagation for the nodes in the same layer only plays a minor role in the force transmittal mechanism, and can be neglected in the deformable simulation.

3.1.5. Scalability

As discussed above, the order of force transmittal is determined by the order of node traversal in BFS. The number of nodes that the traversal would cover, i.e. the number of nodes involved in deformable simulation, is dependent on the mesh topology. Consider an unbounded 2D mesh with the topology as shown in Fig. 3, the number of nodes N due to a single external force is given by

$$N = 3n^2 + 3n + 1 \tag{9}$$

where n is the penetration depth. If a force is applied to a surface node of a 3D lattice, the number of nodes is increased by an order of magnitude, i.e.

$$N = \frac{1}{3}(n + 1)(2n^2 + 4n + 3) \tag{10}$$

The simulation is thus scalable simply by controlling the penetration depth to limit the amount of mass points involved. When real-time interactions are required, the computation time can be reduced effectively with a smaller penetration.

To facilitate the simulation process, connectivity information is stored in the data structure of each node. The immediate neighbors of each node are identified instantly by accessing the data structure during node traversal. Furthermore, each node is allowed to connect to any number of nodes that are required for the geometric modeling of both volumetric and surface models.

3.1.6. Multiple external forces

When external forces are applied to a number of mass points, multiple localized deformation regions are created in the mass-spring network. The algorithm presented in Section 3.1.2 is extended such that the force transmittal mechanism is carried out at each of these regions in a synchronous manner. This is illustrated schematically in Fig. 4. The overall order of force transmittal is synchronized in such a way that nodes in the different localized regions ω_i but at the same topological distance d from their respective origin of traversal N_i are assigned with the same priority. As shown in Fig. 4, the neighboring nodes belonging to different regions are grouped together as common layers according to their topological distances. For a mass-spring network with m mass points subjected to external forces, and the maximum penetration depth in all localized regions is n layers, deformation is simulated by sequential evaluation of updated nodal positions starting from mass points in the layer L_0 to L_n, in an increasing order of depth. The pseudo-code for the algorithm is shown in Fig. 5. Furthermore, since the algorithm is depth-synchronized for all localized regions and each node is only visited once during the search of neighbors, when two regions overlap, the corresponding layers of neighboring nodes located at the same depth would merge together, forming a common layer shared by the two regions.

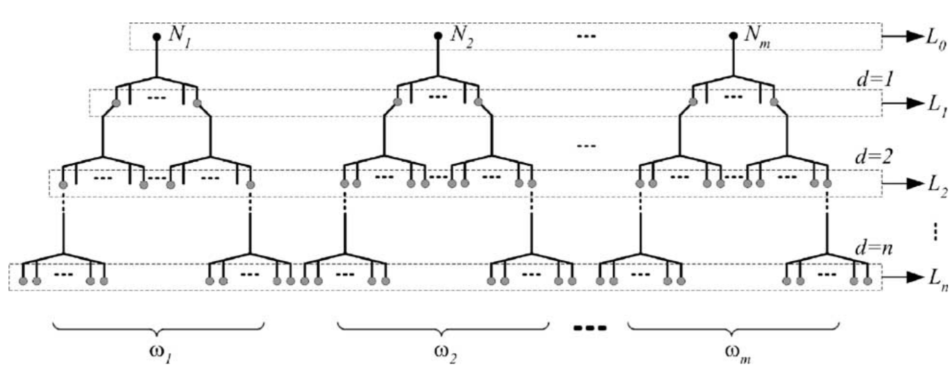

Figure 4 Schematic diagram showing the order of force transmittal when m nodes (N_1–N_m) in a mass-spring network are subjected to external forces, forming m localized deformation regions (ω_1–ω_m). In each region, forces are allowed to penetrate n layers.

```
Force_Transmittal_Order (V, n):
Input:    A set of nodes V = {v_i : i = 1...m } in a mass-spring network;
          The maximum penetration depth n;
Output:   A vector of n+1 containers.
Create an empty container L_0.
for each node v in V do {
    Set v to visited.
    Store v in L_0
}
Set i = 0
while i is not equal to n {
    Create an empty container L_{i+1}
    for each node j in L_i do {
        for each neighbor k of node j do {
      if      k is not been visited then
                Set k to visited
                Store k in L_{i+1}
      }
    }
    increase i by 1
}
The order is given by the vector {L_0, L_1, ..., L_n}
```

Figure 5 The breadth-first search algorithm for the determination of propagation order.

3.2. Identification of model parameters

A major issue with mass-spring-based deformable models is the specification of physically meaningful model parameters to reproduce the behavior of real tissues, which has received relatively little attention in the literature. The determination of such parameters as spring stiffness and nodal damping is resorted to estimations through interactive adjustments in a trial-and-error manner, by comparing to the experimentally measured tissue responses, or by surgeons who are experienced with the mechanical behavior of real tissues. The interactive tuning process iterates until the simulated responses are seemingly realistic, matching the real tissue responses.

Attempts have been made to automate the parameter identification process by using optimization techniques. Louchet et al. [29] have proposed an evolutionary minimization algorithm to identify the parameters of a mass-spring cloth model by optimizing against the synthetic cloth animations or real cloth images. The nodal positions of the model are tuned on a per-frame basis. Joukhadar et al. [30] have applied genetic algorithms (GA) to a parameter identification problem concerning the realistic simulation of the complex contact interactions between a robot and a mass-spring-based object. The behavior of the object is optimized by using GA with a set of constraints such as the maximum deformation, temporal position and velocity of the object. Deussen et al. [31] have employed simulated annealing to identify the model parameters of a 2D mass-spring array, using the analytic solution to the static deformation as a reference. However, to obtain the static deformation analytically, only a simple square plate is used and the test configurations are limited to shear and tensile stress. On the other hand, a representative work related to medical applications is the ingenious approach proposed by Radetzky and Nürnberger [20]. This approach has utilized an artificial neural network model to approximate the underlying tissue model. A neuro-fuzzy system has been developed to enable the definition of model parameters based on a linguistic description to the tissue characteristics. The fuzzy rules and membership functions are designed with medical experts. The values of the fuzzy input and output terms are described by linguistic terms, which are then mapped to the physical parameters of the model by the fuzzy system.

In this paper, we present a parameter identification approach which is capable of identifying the parameters of a 3D mass-spring model by using the static FEM deformation as the reference. The advantage of utilizing FEM is the flexibility that allows the deformation of an object with non-trivial geometry, subjected to arbitrarily applied forces, to be evaluated numerically. The optimization is based on the simulated annealing heuristic. The optimization process along with the generation of the reference data and the simulated annealing process are presented in the following sections.

3.2.1. Heuristic optimization

As described earlier, the basic idea of parameter identification is centered on the extraction of the unknown parameters from a set of reference data which are obtained by experimental measurements or imposing behavioral constraints, or calculated theoretically based on a physically more rigorous model. To obtain the unknown parameters for a 3D mass-spring system, the linear static FEM along with continuum mechanics are used to generate the reference deformation data. It takes advantage of the availability of the linear FEM parameters—Young's modulus and Poisson ratio, which can be measured experimentally for different materials. The parameters for some living tissues are also obtainable from the literature. The unknown mass-spring model parameters are then adjusted until it deforms in a way that approaches the deformation as represented by the reference data. That is, the optimization process involves the deformable simulation of an object using two methods. On the one hand, deformable simulation of the object is carried out with the mass-spring-based force transmittal mechanism (MSM-FTM), while on the other hand, the simulation is repeated for the same object under the same conditions using the linear static FEM, which serves as the benchmark.

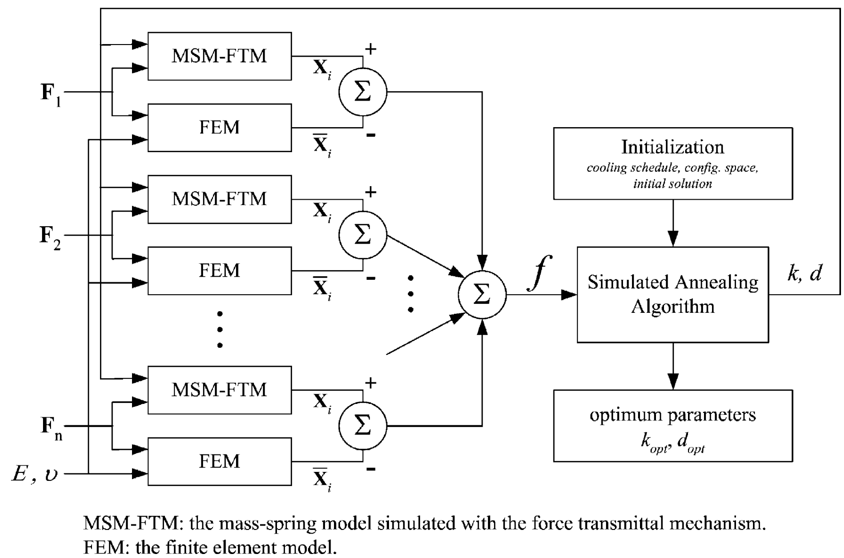

Figure 6 The nodal positions of the mass-spring model (MSM-FTM) are optimized against that of the FEM. Optimization is carried out with both models in equilibrium state. The resulting differences in nodal positions are summed and passed to the simulated annealing algorithm as the value of the cost function f.

The positions of the nodes in the deformed objects as simulated by these two methods are then compared. The distances between the nodes are regarded as errors, which is to be minimized to yield the optimum parameters for the mass-spring model. The framework for the optimization is illustrated in Fig. 6.

To ensure the comparison is made on equal ground, the following conditions are enforced. Since the optimization is performed by benchmarking against the deformation simulated by the static FEM, the mass-spring simulation is allowed to proceed for a sufficient amount of time so that it would reach the equilibrium state. Furthermore, both models are constrained by the same boundary conditions and subjected to the same external forces. It should be emphasized here that although linear FEM may be more accurate, it does not necessarily model the real world exactly. As discussed in Section 2, real tissues are very complicated, which cannot be simulated simply by linear FEM. The use of linear FEM to generate the reference data is primarily due to the availability of measurable FEM parameters that correspond to specific materials.

3.2.2. Reference data

The reference data are generated by using continuum mechanics and the finite element method. Consider a 3D object with volume V and composed of linear isotropic material, the stress vector $\boldsymbol{\sigma}$ and strain vector ε are related by the Hooke's law as

$$\boldsymbol{\sigma} = \boldsymbol{D}\varepsilon \tag{11}$$

where

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix}, \qquad \varepsilon = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \end{bmatrix},$$

$$\boldsymbol{D} = \frac{E}{(1+\upsilon)(1-2\upsilon)} \begin{bmatrix} (1-\upsilon) & \upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & (1-\upsilon) & v & 0 & 0 & 0 \\ \upsilon & \upsilon & (1-\upsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-\upsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-\upsilon) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-\upsilon) \end{bmatrix} \tag{12}$$

In Eq. (12), $\boldsymbol{D}$ is the material matrix, E and υ are the Young's modulus and Poisson's ratio, respectively. The strain vector ε is further expressed in terms of the displacement vector $\boldsymbol{u}$ as $\varepsilon = \boldsymbol{Bu}$, where $\boldsymbol{B}$ is a 6×3 matrix of differential operators and $\boldsymbol{u}$ is a 3×1 vector [16,32]. For static deformation, the displacement $\boldsymbol{u}$ due to an external force $\boldsymbol{f}$ is given by

$$\boldsymbol{f} = \boldsymbol{Ku} \tag{13}$$

where $\boldsymbol{K}$ is the stiffness matrix

$$\boldsymbol{K} = \int\int_V\int \boldsymbol{B}^{\mathrm{T}}\boldsymbol{DB}\,\mathrm{d}V \tag{14}$$

Hence, it is necessary to compute the stiffness matrix $\boldsymbol{K}$ in order to determine the resulting deformation of the object.

The problem is then formulated using FEM which decomposes the object into a number of elements [32]. In our work, the basic shape employed for the finite elements is a hexahedron, containing eight corner nodes with three degrees of freedom for each node. The choice of hexahedron is because it has the same topology as the mass-spring lattice to be optimized. Thus, the global stiffness matrix $\boldsymbol{K}_{\mathrm{g}}$ in Eq. (14) of the entire object is now calculated indirectly by first evaluating the stiffness matrices for each hexahedron element $\boldsymbol{K}_{\mathrm{e}}$, which are assembled subsequently to give the global $\boldsymbol{K}_{\mathrm{g}}$.

To calculate $\boldsymbol{K}_{\mathrm{e}}$, each hexahedron element is transformed from the Cartesian coordinate system (x, y, z) to the natural coordinated system (r, s, t). The displacement within each element is obtained by interpolating the displacement of the eight corner nodes using the tri-linear Lagrange shape function

$$N_i = \tfrac{1}{8}(1 + rr_i)(1 + ss_i)(1 + tt_i) \tag{15}$$

where $i = 1$–8, corresponding to the eight corners with r_i, s_i and t_i equal to either $+1$ or -1. The element stiffness matrix is then evaluated as

$$\boldsymbol{K}_{\mathrm{e}} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{DB}\,\det(\boldsymbol{J})\,\mathrm{d}r\,\mathrm{d}s\,\mathrm{d}t \tag{16}$$

where $\boldsymbol{J}$ is the Jacobian of the coordinate transformation. Once the $\boldsymbol{K}_{\mathrm{e}}$'s of all the hexahedrons are computed, the global stiffness $\boldsymbol{K}_{\mathrm{g}}$ of the object is obtained by the assemblage of the $\boldsymbol{K}_{\mathrm{e}}$'s, and the static deformation can be evaluated by Eq. (13). Further details can be found in [32,33].

3.2.3. Cost function

The objective of the optimization problem is to minimize the difference in nodal positions between the force transmittal model and the finite element model. Referring to Fig. 6, suppose the spring stiffness k and damping constant d are the parameters to be tuned during optimization, the position vectors of the ith node in the force transmittal model and the FEM are $\boldsymbol{X}_i$and $\bar{\boldsymbol{X}}_i$, respectively, then the cost function f is given by the sum of the Euclidean distances between all nodes in the two models, that is

$$f(k, d) = \sum_{\boldsymbol{F}}\sum_{i} |\boldsymbol{X}_i(k, d, \boldsymbol{F}) - \bar{\boldsymbol{X}}_i(\boldsymbol{F})| \tag{17}$$

where $\boldsymbol{F}$ the set of force configurations, each configuration defines the forces applied to the nodes in two models. Both models are subjected to identical loading and boundary conditions. Optimization is achieved by minimizing the distances between the nodes in the two models so that the optimum values of spring stiffness and damping constant k_{opt} and d_{opt} are obtained as f is minimized.

3.2.4. Simulated annealing

Originated in statistical thermodynamics, simulated annealing is a flexible optimization technique that simulates the thermal annealing process in metallurgy for the removal of internal stress and structural defects [34]. It is a flexible optimization method and widely applied in many combinatorial optimization problems. An important difference between SA and other heuristic optimization techniques is the random evaluations of the cost function which allows an intermediate solution to escape from local minimum by making a probabilistic uphill move. Hence, it offers a higher chance to reach the global minimum even if many local minima exist in the cost function. The probability Pr of an uphill move, i.e. the metropolis acceptance criterion, is given by

$$\mathrm{Pr} = \begin{cases} 1 & \Delta f \leq 0 \\ \mathrm{e}^{-\Delta f / k_{\mathrm{B}} T} & \Delta f > 0 \end{cases} \tag{18}$$

where Δf is the change in function value for such a transition, k_{B} is the Boltzman constant and T is the temperature.

Cooling schedule plays an important role in simulated annealing. The temperature decrement function, the initial and the final temperature would considerably affect the optimization process. It is evident from Eq. (18) that since Pr increases with T, annealing at a high temperature would result in numerous random uphill moves without significant minimization of the cost function. The frequent update of the system status also slow down the optimization process. On the contrary, low temperature precludes uphill moves, reducing SA to a slow descent algorithm. A pre-processing stage has

```
Initialize:
Define configuration space:
    S={(k, d) | k∈K, d∈D}, where K={k | kmin ≤ k ≤ kmax}, D={ d | dmin ≤ d ≤ dmax}.
Set cooling schedule parameters
Select an initial solution (kopt, dopt)  S at random.
Compute the cost function fmin = f(kopt, dopt) with Eq.(17).
Set system temperature T = initialTemperature;
Simulated Annealing:
While T is larger than finalTempearture do
{
    Set i = 0;
    While i is not equal to numIterations do {
        Select a random solution (k, d)  S in the neighborhood of (kopt, dopt).
        Compute the cost function f = f(k, d) with Eq.(17).
        CalculateΔf = f − fmin ;
          IfΔf ≤ 0
              (kopt, dopt) ←(k, d); fmin ← f;
          else
              generate random variable R  (0, 1)
              if R ≤ exp(-Δf /T)
                  (kopt, dopt) ← (k, d) ; fmin ← f;
        increase i by 1
    }
    Reduce T by one temperature step
}
```

Figure 7 The simulated annealing algorithm developed to identify the parameters of MSM-FTM.

been carried out to locate the suitable temperature range before simulated annealing proceeds. The temperature reduction function $\alpha(T) = T/(1+\beta T)$ proposed by Lundy and Mees [35] is employed for the cooling process, where β is a suitably small value. System temperature is reduced very slowly but only one iteration is performed at each temperature-step. The system is frozen when the temperature is below the final temperature where no more transition is possible. The SA algorithm developed for optimizing the force transmittal model against the FEM deformation is shown in Fig. 7.

3.3. Incorporation of force feedback

Force feedback is not only involved in medical simulation to enhance the realism of virtual environments, but also to provide diagnostic information from the sense of touch. For example, palpation is an important technique in cancer diagnosis that size, location and tenderness of abnormal tissues are detected by hand. In Chinese medicine, the state of balance of the whole body is determined by pulse diagnosis and touching local areas. Force feedback is incorporated into the force transmittal model by using the six degrees of freedom haptic device PHANToM Desktop from the SensAble Technologies Inc. [36]. User interacts with deformable objects in the virtual environment by manipulating the stylus of PHANToM. Point-based haptic rendering technique [37] is used in which the stylus endpoint of PHANToMs (i.e. haptic interface point, HIP) is modeled as the probing object. During the course of haptic simulation, collision detection algorithm checks continuously if the HIP collides with a virtual

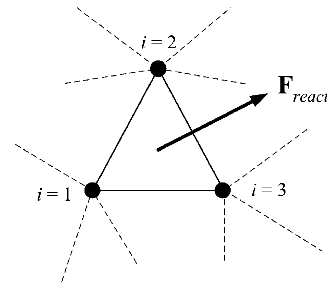

Figure 8 Calculation of reaction force by the vector sum of forces at the three vertices of a collided polygon.

object. For an object modeled with triangulated meshes, the three vertices of a collided polygon serve as the starting points for the breadth-first traversal to determine the order of force transmittal in subsequent deformable simulation. To constrain the HIP on the object surface, the collided polygon is translated with a displacement as defined by the HIPs penetration into the object. The reaction force $\boldsymbol{F}_{\text{react}}$ resulting from such displacement is evaluated by the vector sum of spring forces at the three vertices (see Fig. 8), that is

$$\boldsymbol{F}_{\text{react}} = -\sum_{i=1}^{3}\sum_{j=1}^{n_i}\boldsymbol{f}_{ij} \quad (19)$$

where n_i is the number of neighbors of the ith vertex of collided polygon and f_{ij} is the spring force on the ith vertex due to its jth neighbor. Deformable simulation then proceeds with forces transmitting successively to the neighboring nodes on a node-by-node basis according to force transmittal mechanism. During simulation, reaction forces are computed to drive the haptic device and the nodal positions are updated to refresh graphics.

3.3.1. Efficient haptic rendering

High haptic refresh rate of 1 kHz is required by PHANToM to generate natural force feedback. If computationally intensive deformable model is employed, it is only resorted to multi-processor environment or distributed computing [38]. In contrast, the proposed approach is suitable for realizing haptic rendering on a standalone and generic PC with single processor. As computational speed decreases with increasing number of nodes involved, the simulation can be expedited simply by reducing the extent of localization. However, when large deformation is necessitated, computational time extends and the disparity between haptic refresh rate and simulation speed becomes more serious. The rate at which reaction forces are calculated cannot cope with the 1 kHz haptics simulation, which causes discontinuity in force feedback.

This is alleviated by generating additional force data through the extrapolation of the two most updated and consecutive reaction force data [39].

4. Performance and evaluation

In this section, the deformable model based on the force transmittal mechanism is evaluated comprehensively in three aspects. First, the mechanical behavior and computation efficiency are discussed, followed by the extraction of model parameters by optimizing against linear static FEM, and finally the realization of a VR-based medical simulation system with the incorporation of force feedback device.

4.1. Mechanical behavior

4.1.1. Estimation of optimum penetration depth

The estimation of optimum penetration depth has been illustrated by using an isotropic mass-spring lattice. A constant point load is applied to a node on the top of the lattice. Simulation is carried out with the penetration depth increased incrementally so that the localized deformation region, i.e. the bounding volume as defined by the penetration depth, expands in a stepwise manner to include more nodes for the simulation. On the other hand, the nodes falling outside the bounding volume do not involve in the simulation and remain fixed during the deformation process. The simulated shapes of the lattice, represented by the profiles, are recorded at each penetration depth and plotted in Fig. 9. It is evident that further increase in penetration beyond four layers does not produce noticeable change in the profile, which suggests that the optimum depth in four layers.

It is noteworthy to point out that the optimum penetration depth is dependent on the stiffness of the mass-spring model. The dependence has been demonstrated by conducting the estimation process for isotropic lattices having different spring stiffness. Three isotropic lattices with spring stiffness $k = 0.5$, 2.0 and 5.0 are deformed by the same external force. The simulated profiles at each penetration depth increment are recorded. As indicated by the deformation results shown in Fig. 10, the optimum penetration depths are 4, 6 and 8 layers, respectively, for the three lattices, suggesting that the greater the stiffness, the deeper the required penetration depth. This can be explained by the fact that, under the same external force, it would be more difficult to deform a hard object and only result in small deformation in a very localized region. Hence, a small penetration depth would be sufficient to simulate the deformation.

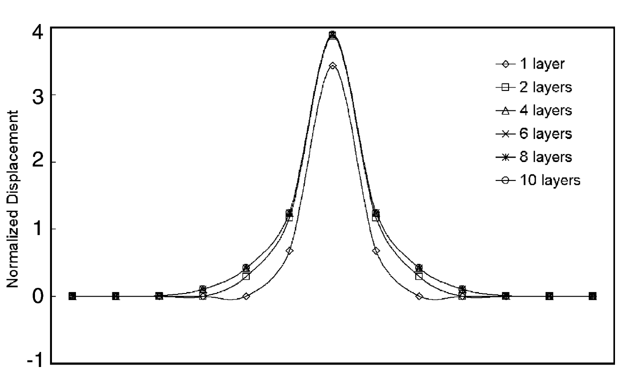

Figure 9 Deformed profiles simulated with different penetration depths.

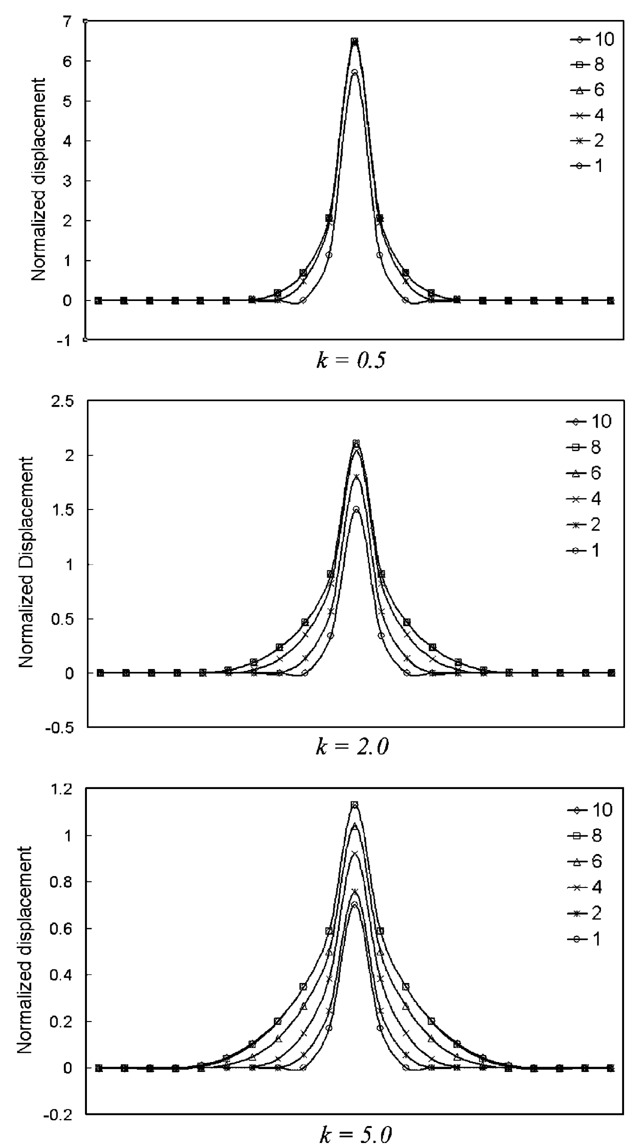

Figure 10 Simulated profiles of a deformed lattice with spring stiffness $k = 0.5$, 2.0 and 5.0.

4.1.2. Tissue responses

To access the ability of the force transmittal mechanism to model real tissues, a set of experiments have been conducted to test if the approach could simulate the typical tissue behavior. It includes the two common features of viscoelasticity—strain creeping and hysteresis, as well as the nonlinear load-deformation relationship [22,40]. In the mass-spring-based viscoelastic model, the resistant force exerted on each mass point is responsible for the simulation of the viscous behavior. The force, modeled by using a damper, is proportional to the negative of velocity and the proportionality constant, i.e. the damping constant, determines the viscosity of the system. In the following experiments, the damping constant of the mass-spring system is varied to study the simulated mechanical properties of materials with different viscosity. The penetration depth is set in such a way that the simulation covers the greatest amount of nodes in the mass-spring system.

In the creeping test, a constant force is applied to a mass-spring lattice and the resulting strain is recorded as a function of time. Deformation of three materials with different viscosity is modeled by varying the damping constant. In Fig. 11(a), the variation in strain against time shows that, under a constant stress, the strain builds up asymptotically toward a maximum value. Among the three viscoelastic materials, the one with the least viscosity has the highest strain rate. The simulated behavior is similar to the creeping measurements obtained from real tissues [22,40]. Next, cyclic loading is simulated to study the effect of hysteresis. A mass-spring lattices is subjected to an external force which is ramped up at a constant rate from zero to a maximum value and then down to zero again. The plot of strain against force in Fig. 11(b) shows that the loading and unloading processes follow two distinct paths, demonstrating the phenomenon of hysteresis as observed in real tissues [22]. Also shown in the experiment is that the width of the hysteresis loop increases with the viscosity of material, which demonstrates that the effect hysteresis is more significant for a material having a higher viscosity. In the load-deformation test, the force applied to the lattice is increased at constant rate and the resulting strain is measured. The curves plotted in Fig. 11(c) shows that the deformation varies nonlinearly with the applied force, which resembles the measurements obtained from living tissues [40]. The results of the above experiments indicate that the force transmittal mechanism can be used to model the typical behavior of real living tissues. Deformable simulation for tissues with different viscosity can be performed by adjusting by the damping constant of the mass-spring system.

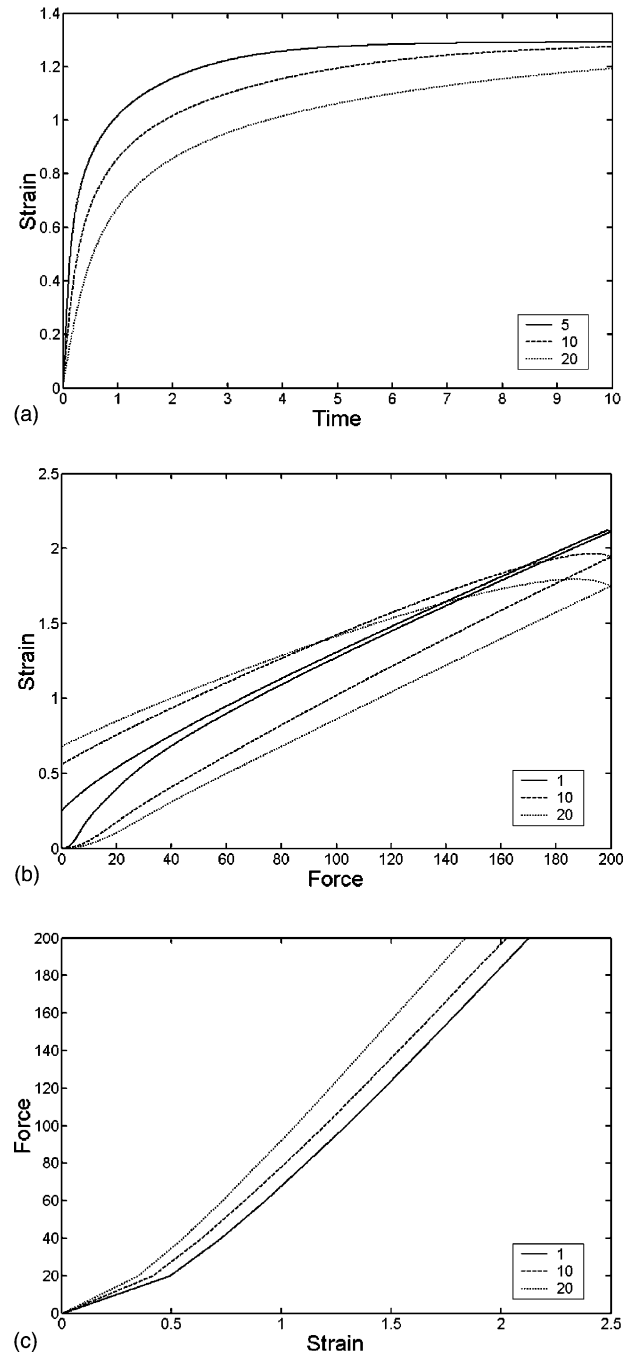

Figure 11 Typical tissue behavior simulated by the force transmittal mechanism: (a) creeping; (b) hysteresis; and (c) nonlinear load-deformation relationship.

It should be pointed out that all the above experiments were conducted with the same integration time-step because, in addition to the physical parameters of the model, the resulting deformation is also dependent on the time-step. As in most mass-spring simulations, the time-step is often set to be small enough to maintain numerical stability, whereas a large time-step would lead to oscillations or the collapse of the mass-spring system.

4.1.3. Anisotropy

The ability of the force transmittal mechanism to simulate the anisotropic behavior of materials has been evaluated by setting the spring stiffness along specific directions to different values. In Fig. 12, two transversely isotropic models are created by assigning the vertical (k_y) and horizontal springs (k_x and k_z) with different stiffness. In model A, the stiffness of the horizontal spring is 10 times smaller than that of the vertical springs, whereas the reverse is applied to model B. An isotropic model is also created as a control. To study the anisotropic behavior, a force of constant magnitude is applied vertically to a node on the top, and horizontally to a node on one side of the models. The shapes of the deformed models, as depicted by its profiles, under these two scenarios are compared. As expected, the simulated profiles for the isotropic model due to the vertical and horizontal force are identical, regardless of the direction of the applied force. For model A, since the horizontal stiffness is smaller, it experiences greater deformation when the force is applied horizontally, as indicated by the rounded corners in the deformed profile. In contrast, as the stiffness of the horizontal springs is larger in model B, it appears harder when being deformed horizontally and the resulting deformation is smaller when compared to the deformation due to a vertical force. The direction-dependent deformation illustrated in this experiment shows that the proposed force transmittal mechanism can be used to simulate the deformation of anisotropic materials.

4.2. Deformable simulation with medical data

Deformable simulation based on the force transmittal mechanism has been applied to medical imaging data. The work is divided into two parts—the visualization of volumetric medical data and the generation of deformable models.

4.2.1. 3D visualization

The visualization of volumetric medical data is achieved by using the technique of *direct volume rendering*. Hardware-supported 3D texture mapping is utilized to store the volume data in solid texture memory. The volume is sliced with a set of planes orthogonal to the viewing direction. Each plane is rendered and shaded using the *Z*-buffer algorithm, and is alpha-blended using the opacity values of the 3D volume texture. With hardware

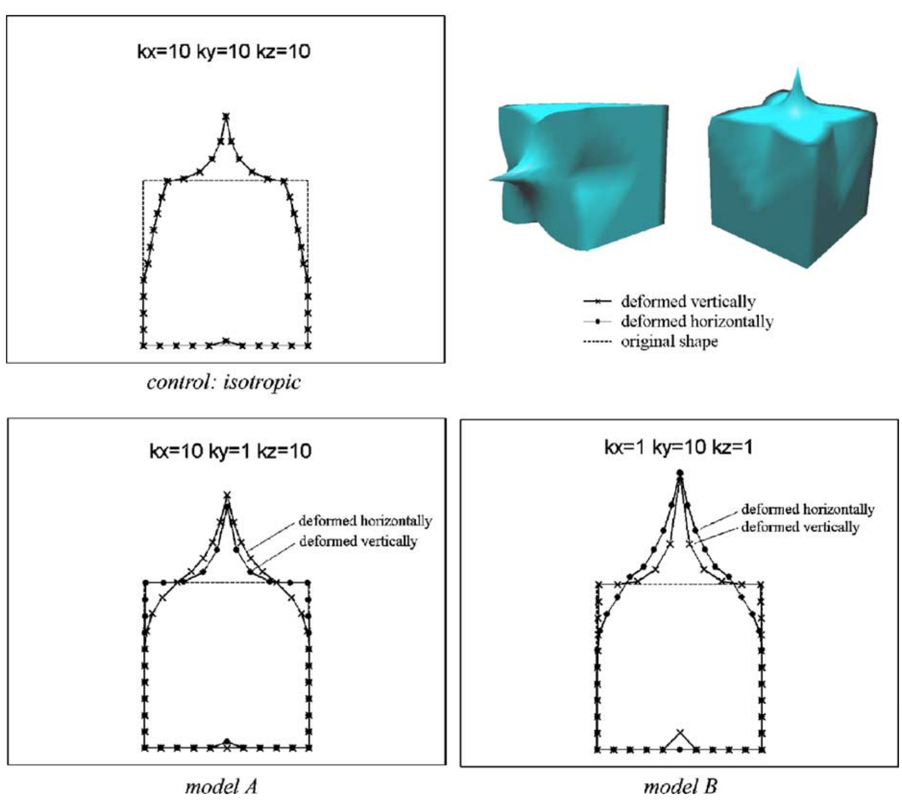

Figure 12 Simulated deformation of transversely isotropic lattices having different vertical and horizontal stiffness.

acceleration, this method can reach a real-time frame rate. However, the large size of texture memory becomes the main concern. This method is thus adopted by mapping the volume data into a number of rectangular planes aligned in the *Z* direction using 3D texture mapping. The blending function uses the alpha value in the data source or assigned by the gray-scale value proportional to opacity. It is intuitive that a large gray-scale value usually corresponds to bones or other solid tissues, whereas a small gray-scale value corresponds to fat or flesh. The planes are drawn in the back-to-front order. With sufficient data, the visual effect is satisfactory.

4.2.2. Mass-spring controlled free-form deformation

Deformable simulation for medical data is achieved by incorporating the physically-based force transmittal mechanism into the purely geometric free-form deformation. In conventional FFD, knowledge of the correlation between the movements of control points and the resulting changes in shape is required to achieve the desired deformation. With the aid of the force transmittal mechanism, the deformation of the control mesh is now governed by the dynamics of the mass points. The modeling process is as follows. A set of volumetric medical data is first embedded in a mass-spring network which serves as a control mesh. When an external force is applied to a node in the control mesh, the neighboring nodes are displaced according the force transmittal mechanism, thereby resulting in localized mesh deformation. As the surrounding space of those displaced nodes is also deformed, it induces localized deformation to the embedded volume data.

Deformable simulation has been performed using MRI knee data. The data contains 256^2 by 127 slices. It is embedded in a mass-spring control mesh of $10 \times 10 \times 10$ nodes. Nine slices are inserted between each pair of nodes by tri-linear interpolation so that there are totally 100 slices for alpha-blending and final image composition. Fig. 13 shows the deformation of the MRI knee data by using the force transmittal mechanism. Further illustrations using the MRI head data are also shown in Fig. 13.

Notice that for medical data containing hard tissues that are non-deformable, some of the nodes in the mass-spring mesh should be fixed during deformable simulation if they are surrounded the hard tissues. This is taken into account by constraining the movement of the nodes that correspond to the non-deformable hard tissues so that the nodes remain fixed throughout force transmittal process. Whether a node is movable or fixed is determined according to the density value of the tissue surrounding the node.

4.3. Parameter extraction

The simulated annealing algorithm developed for model parameter extraction has been experimented to obtain the suitable spring stiffness and damping constant that correspond to the mechanical properties of elastin. A cubic elastin sample of size $0.1\,\mathrm{m} \times 0.1\,\mathrm{m} \times 0.1\,\mathrm{m}$ is modeled with 1000 nodes and 729 hexahedrons. Forces are applied to four central nodes located on one face of the sample while the opposite face is fixed. Reference deformation data of elastin is first simulated using the linear static FEM. The characterizing parameters of elastin employed in FEM, including mass density, Young's modulus and Poisson's ratio, are $1040\,\mathrm{kg\,m^{-3}}$, 0.6 and 0.45 MPa, respectively. The spring stiffness and damping constant in the force transmittal model are then tuned by using simulated annealing such that the resulting deformation approaches the FEM deformation. The optimization parameters and results are given in Table 1, the corresponding errors are shown in Table 2. Since some of the nodes in the two models have the same positions, the minimum nodal position error is zero. The discrepancy between the optimized model and the FEM reference data is evaluated by the difference in nodal positions, normalized to the natural length of spring. Under

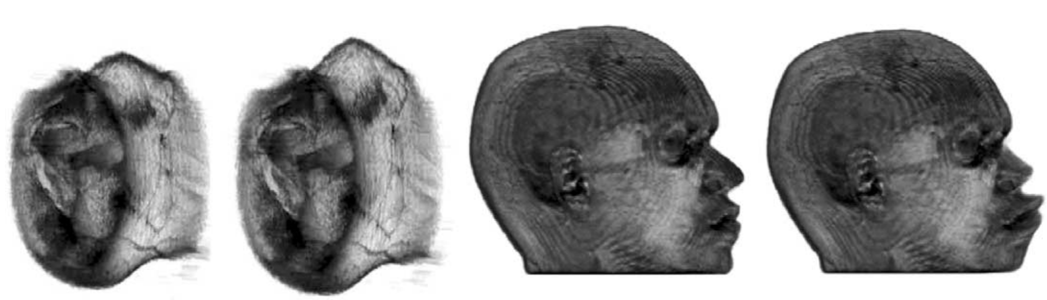

Figure 13 Application of the force transmittal mechanism to deform the MRI datasets of knee and head.

Table 1 Model parameters extraction by simulated annealing

	Elastin
FEM	
Young's modulus E (MPa)	0.6
Poisson's ratio σ	0.45
Density ρ (kg m^{-3})	1040
External force range (N)	10—40
Simulated annealing	
Initial temperature	0.1—1.0
β in Lundy and Mee's cooling function	0.4—0.5
Simulation based on force transmittal mechanism	
Penetration depth n (layers)	10
Optimization results	
Spring stiffness k_{opt}	2575
Nodal damping d_{opt}	2149

Table 2 Optimization error at each applied force

Force (N)	Average error (%)	Maximum nodal position error (%)
10	2.7	9
20	5.4	17.1
40	10.9	27

the specified range of external forces, the average discrepancy in nodal position between the force transmittal model and FEM is about 6% of the natural length of spring, which only produces minor visual differences. When the smallest specified force is applied, the average discrepancy is only 2.7%. However, the experimental results show that the discrepancy increases with the magnitude of external forces. This is expected due to the intrinsic differences in the principle of the two models. While a close matching can be achieved for small deformation, it becomes difficult to approach the FEM results when deformation is very large.

4.4. Haptic simulation

A multi-threaded platform has been developed to incorporate haptic simulation into the proposed deformable model. Fig. 14 shows the system framework. Three threads are created, respectively, for dynamics simulation, haptic rendering and visualization. Because of the high bandwidth requirement, the haptics loop is assigned with the highest priority in order to maintain fast refresh rates. It provides the dynamics simulator with the position of the virtual probe and generates feedback forces to the user. The dynamics simulator performs collision detection and carries out deformable simulation if collision occurs. The simulator computes updated nodal positions and passes them to the visualization loop for graphics refresh. It also computes the reaction forces required by the haptics loop to generate force feedback.

Real-time interactive deformation of virtual organs using the PHANToM Desktop has been achieved. In Fig. 15, surface-based liver and stomach model are deformed interactively using a virtual probe which represents the end effector of the PHANToM arm. Once collisions between PHANToMs HIP and virtual organ are detected, vertices of collided polygon are passed to the deformation

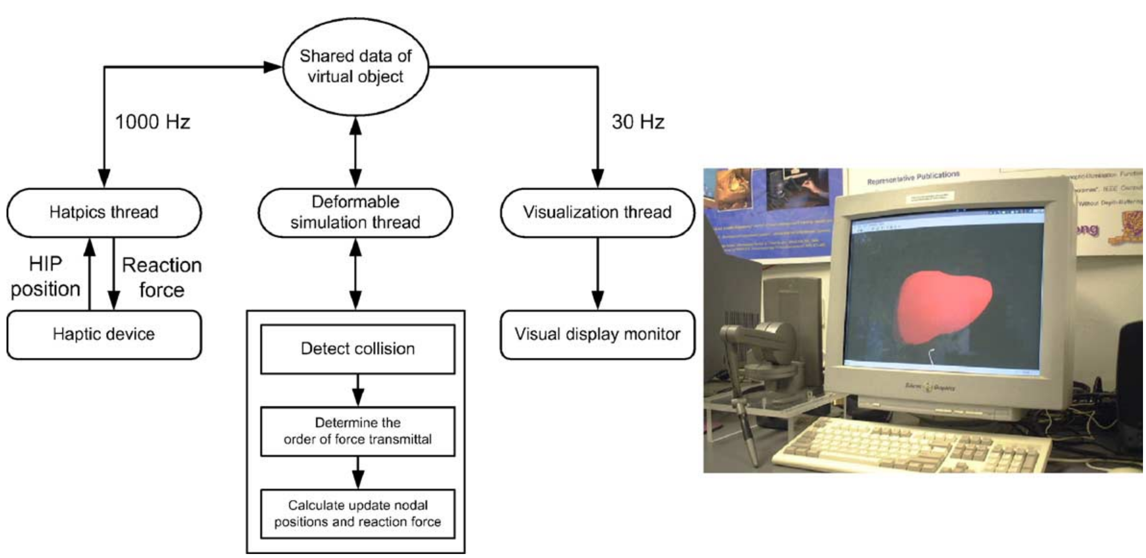

Figure 14 System framework for deformable simulation with haptic feedback (left). Setup for interactive deformable simulation on generic PC platform with PHANToM Desktop (right).

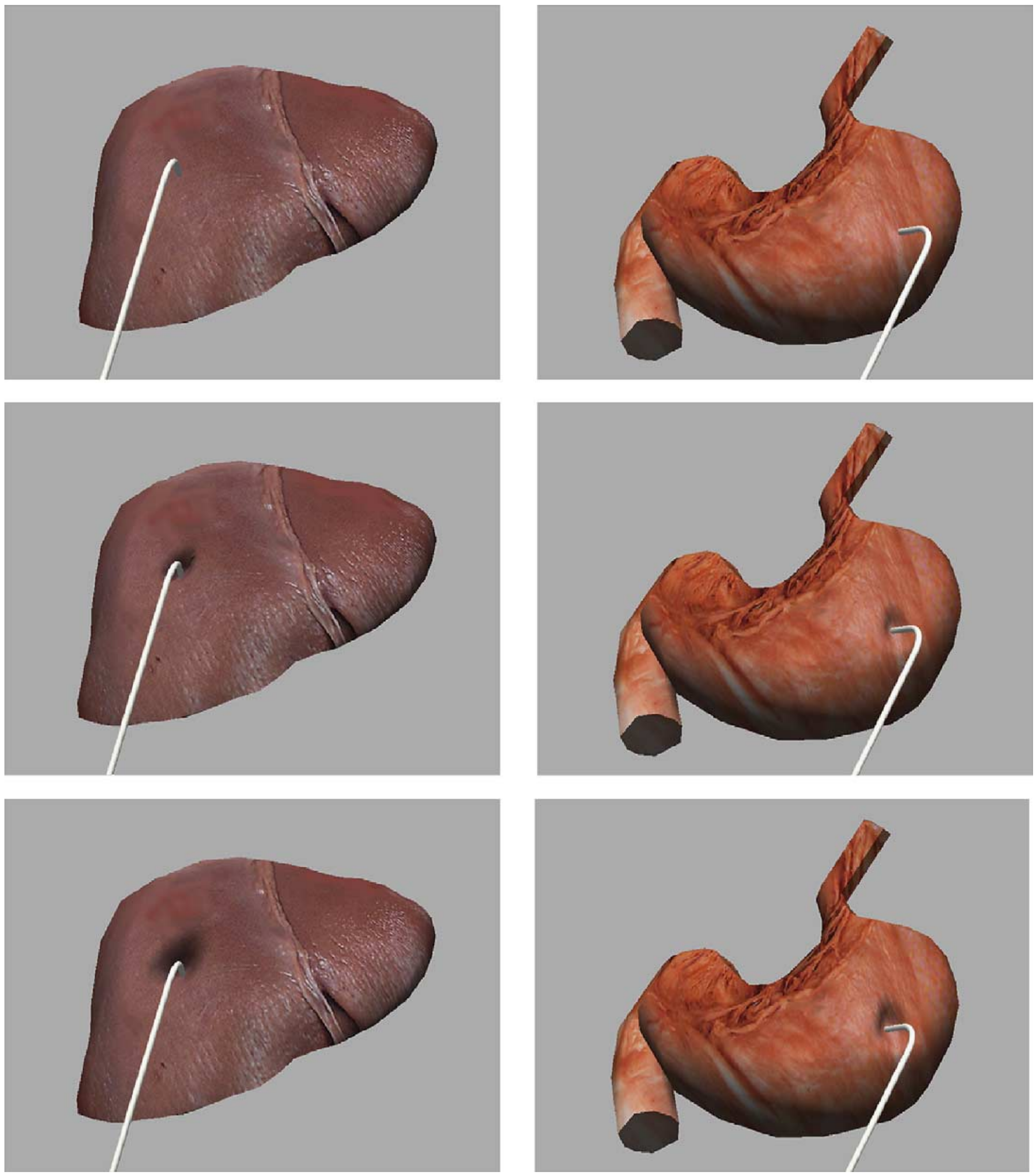

Figure 15 Interactive deformation of 3D models of lung (1769 nodes, 3534 polygons) and stomach (512 nodes, 1020 polygons). Deformation is simulated based on the proposed force transmittal mechanism.

simulator to determine the order of force transmittal. Updated positions of the nodes in the neighborhood of the vertices are then calculated and used to refresh graphics. In the meantime, the reaction forces are computed and perceived by user through PHANToM.

4.5. Computation efficiency

Timing experiment has been conducted to evaluate computation efficiency with an isotropic 2D lattice with the same topology as the one shown in Fig. 3. The lattice contains 50×50 nodes and a constant force is applied to the central node. Time elapsed for searching neighboring nodes and calculating updated node positions during deformable simulation are measured at different penetration depths. For a penetration of n layers, the number of nodes involved in the simulation at each time-step is given by Eq. (9). Simulation is carried out on a generic Intel Pentium III 800-MHz PC without graphics acceleration. Refer to Table 3, as penetration depth increases from 1 to 20 layers, it takes 0.14—31 ms to update nodal positions for one time-step, corresponding to refresh rates of 32 Hz to over 7 kHz. Hence, by limiting the extent of deformation appropriately, computation can be expedited to a level that facilitates real-time interactive applications.

Table 3 Variation in computation time against penetration depth

Penetration (layers)	Number of nodes	Computation time (ms)	
		Neighbor search	Deformable simulation
1	7	0.02	0.14
2	19	0.07	0.38
3	37	0.1	0.49
4	61	0.27	0.97
5	91	0.44	1.42
10	331	2.56	5.87
15	721	10.57	14.79
20	1261	28.79	30.97

5. Conclusion

An efficient and scalable deformable model has been presented in this paper. The simulation is based on the force transmittal mechanism in which breadth-first search algorithms have been used to determine the propagation order. The computation speed is adjustable simply by varying the penetration depth. The model has demonstrated typical tissue responses and anisotropic behavior.

To deal with the common problem of parameter identification in mass-spring system, an attempt has been made to extract the parameters through the optimization against the reference data generated by using linear static FEM. Simulated annealing algorithms have been developed to obtain the spring stiffness and damping constant of a 3D mass-spring lattice. However, the simulated deformation is indeed a multivariable function of the physical parameters of the model (including mass, spring stiffness, damping constant and penetration depth) as well as the time-step in numerical integration. The numerous variables further complicate the identification process, making it difficult to tune the simulated deformation to the desired tissue responses.

The deformable model has been applied to deform volumetric medical data, and has been effectively integrated with a commercial haptic device on a generic PC platform. It provides a feasible solution for the development of VR-based medical applications, e.g. medical training system pertaining to anatomy, palpation diagnosis or surgery.

By taking advantages of computation efficiency and scalability, the proposed deformable modeling technique will be explored for the development of online medical applications. It has the potential for the development of web-based surgical training involving deformable simulation [41] since fast computation and high refresh rate can be achieved even with a generic PC. Simulation speed can be adjusted dynamically by varying penetration depth to adapt to network traffic. Furthermore, the bandwidth requirement is low as the information required for transfer only includes the mass points subjected to external forces and the associated force vectors. In particular, it is promising to develop collaborative virtual environments built on client-server architecture, which allows the planning and training of medical procedures involving multiple user interactions. Research is underway to build a VR-based system containing virtual organs with local variations in stiffness, which will be used for training palpation during medical examination. Model parameters corresponding to different tissue stiffness will be determined by using the simulated annealing algorithm. Ongoing work also includes the simulation of tissue cutting to enhance the features and realism of the model.

Acknowledgements

This work was supported by the Research Grants Council of the HKSAR (Nos. 4356/02E, CUHK 1/00C) and CUHK direct grant No. 4450002. The authors would like to thank Prof. J.C.Y. Cheng and Prof. K.M. Chan of the Department of Orthopaedics and Traumatolgy for their expertise in surgical procedures, and L. Bai for the assistance in processing the MRI datasets. Thanks are extended to the anonymous reviewers for helpful discussions and editorial comments.

References

[1] Acar B, Napel S, Paik D, Göktürk B, Tomasi C, Beaulieu CF. Using optical flow fields for polyp detection in virtual colonoscopy. In: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention; 2001. p. 637—44.
[2] Kiss G, Van Cleynenbreugel J, Thomeer M, Suetens P, Marchal G. Computer aided diagnosis for virtual colonography. In: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention; 2001. p. 621—8.
[3] Heng PA, Fung PF, Wong TT, Leung KS, Sun H. Virtual bronchoscopy. Int J Virtual Reality 2000;4(4):10—20.
[4] Kühnapfel U, Çakmak HK, Maaβ H. Endoscopic surgery training using virtual reality and deformable tissue simulation. Comput Graphics 2000;24(3):671—82.
[5] Basdogan C, Ho CH, Srinivasan MA. Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration. IEEE/ASME Trans Mechatron 2001;6(3):269—85.
[6] Delson NJ, Koussa N, Hastings RH, Weinger MB. Quantifying expert vs. novice skill in vivo for development of a

laryngoscopy simulator. In: Proceedings of Medicine Meets Virtual Reality; 2003. p. 45—51.
[7] Burdea GC, Patounakis G, Popescu V, Weiss RE. Virtual reality-based training for the diagnosis of prostate cancer. IEEE Trans Biomed Eng 1999;46(10):1253—60.
[8] Leung KM, Heng PA, Sun H, Wong TT. A haptic needle manipulation simulator for Chinese acupuncture. In: Proceedings of Medicine Meets Virtual Reality; 2003. p. 187—9.
[9] Cai Y, Chui CK, Wang Y, Wang Z, Anderson JH. Parametric eyeball model for interactive simulation of ophthalmologic surgery. In: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention; 2001. p. 465—72.
[10] Sagar MA, Bullivant D, Mallinson GD, Hunter PJ. A virtual environment and model of the eye for surgical simulation. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques; 1994. p. 205—12.
[11] Keeve E, Girod S, Girod B. Craniofacial surgical simulation. In: Proceedings of the 4th International Conference on Visualization in Biomedical Computing; 1996. p. 541—6.
[12] Lee TY, Sum YN, Lin YC, Lin L, Lee C. Three-dimensional facial model reconstruction and plastic surgery simulation. IEEE Trans Inform Technol Biomed 1999;3(3):214—20.
[13] Burdea GC. Haptic issues in virtual environment. In: Proceedings of the Computer Graphics International; 2000, p. 295—302.
[14] Sederberg TW, Perry SR. Free-form deformation of solid geometric models. In: Proceedings of SIGGRAPH; 1986, p. 151—60.
[15] Baraff D, Witkin A. Large steps in cloth simulations. In: Proceedings of the SIGGRAPH; 1998. p. 43—54.
[16] Bro-Nielsen M. Finite element modeling in surgery simulation. Proc. IEEE 1998;86:490—503.
[17] Pentland A, Williams J. Good vibrations: modal dynamics for graphics and animation. In: Proceedings of the SIGGRAPH; 1989. p. 215—22.
[18] Choi KS, Sun H, Heng PA, Cheng JCY. A scalable force propagation approach for web-based deformable simulation of soft tissues. In: Proceedings of the 7th International Conference on 3D Web Technology; 2002, p. 185—93.
[19] Brown J, Sorkin S, Bruyns C, Latombe JC, Montgomery K, Stephanides M. Real-time simulation of deformable objects: tools and application. In: Proceedings of Computer Animation; November 2001. p. 228—36.
[20] Radetzky A, Nürnberger A. Visualization and simulation techniques for surgical simulators using actual patient's data. Artificial Intelligence Med 2002;26(3):255—79.
[21] Sørenson TS, Therkildsen SV, Makowski P, Knudsen JL, Pedersen EM. A virtual reality approach for planning cardiac interventions. Artificial Intelligence Med 2001;22(3): 192—214.
[22] Fung YC. Biomechanics: mechanical properties of living tissues. 2nd ed. New York: Springer-Verlag; 1993.
[23] Landau LD, Lifshitz EM. Theory of elasticity. 3rd ed. New York: Pergamon Press; 1986.
[24] Maurel W, Wu Y, Thalmann NM, Thalmann D. Biomechanical models for soft tissue simulation. Berlin: Springer-Verlag; 1998.
[25] Senger S. Visualizing and segmenting large volumetric data sets. IEEE Comput Graphics Appl 1999;19(3):32—7.
[26] Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques; 1987. p. 163—9.
[27] Poston T, Wong TT, Heng PA. Multi-resolution isosurface extraction with adaptive skeleton climbing. In: Proceeding of Eurographics; 1998. p. 137—48.
[28] Ackerman MJ. The visible human project. Proc IEEE 1988;86(3):504—11.
[29] Louchet J, Provot X, Crochemore D. Evolutionary identification of cloth animation models. In: Proceedings of the Eurographics Workshop on Computer Animation and Simulation; 1995. p. 44—54.
[30] Joukhadar A, Garat F, Laugier Ch. Parameter identification for dynamic simulation. In: Proceedings of the IEEE International Conference on Robotics and Automation; 1997. p. 1928—33.
[31] Deussen O, Kobbelt L, Tucke P. Using simulated annealing to obtain good nodal approximations of deformable bodies. In: Proceedings of the Eurographics Workshop on Computer Animation and Simulation; 1995. p. 30—43.
[32] Rao SS. The finite element method in engineering. 3rd ed. Boston, MA: Butterworth; 1999.
[33] Zienkiewicz OC, Taylor RL. The finite element method. 5th ed. Boston, MA: Butterworth; 2000.
[34] Laarhovan PJM, Aarts EHL. Simulated annealing: theory and applications. Dordrecht: Reidel; 1987.
[35] Lundy M, Mees A. Convergence of an annealing algorithm. Math Programming 1986;34(1):111—24.
[36] Salisbury K, Brock D, Massie T, Swarup N, Zilles C. Haptic rendering: programming touch interaction with virtual objects. In: Proceedings of the 1995 Symposium on Interactive 3D Graphics; 1995. p. 123—30.
[37] Zilles CB, Salisbury JK. A constraint-based god-object method for haptic display. In: Proceedings of the IEEE International Conference on Intelligent Robots Systems, Human Robot Interaction, and Co-Operational Robots, vol. 3; 1995. p. 146—51.
[38] Cotin S, Delingette H. Real-time surgery simulation with haptic feedback using finite elements. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2000. p. 3739—44.
[39] Picinbono P, Lombardo JC, Delingette H, Ayache N. Anisotropic elasticity and force extrapolation to improve realism of surgery simulation. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2000. p. 596—602.
[40] Kenedi RM, Gibson T, Evans JH, Barbenel JC. Tissues mechanics. Phys Med Biol 1975;20(5):699—717.
[41] El-Khalili NH, Brodlie KW. Surgical training on the web. Future Generation Comput Syst 2000;17:147—58.