

Chapter 2

Heterogeneous Deformable Modeling of Bio-Tissues and Haptic Force Rendering for Bio-Object Modeling

Shiyong Lin, Yuan-Shin Lee, and Roger J. Narayan

Abstract This paper presents a novel technique for modeling soft biological tissues as well as the development of an innovative interface for bio-manufacturing and medical applications. Heterogeneous deformable models may be used to represent the actual internal structures of deformable biological objects, which possess multiple components and nonuniform material properties. Both heterogeneous deformable object modeling and accurate haptic rendering can greatly enhance the realism and fidelity of virtual reality environments. In this paper, a tri-ray node snapping algorithm is proposed to generate a volumetric heterogeneous deformable model from a set of object interface surfaces between different materials. A constrained local static integration method is presented for simulating deformation and accurate force feedback based on the material properties of a heterogeneous structure. Biological soft tissue modeling is used as an example to demonstrate the proposed techniques. By integrating the heterogeneous deformable model into a virtual environment, users can both observe different materials inside a deformable object as well as interact with it by touching the deformable object using a haptic device. The presented techniques can be used for surgical simulation, bio-product design, bio-manufacturing, and medical applications.

2.1 Introduction

The techniques of modeling of biological tissues and deformable objects for bio-manufacturing and medical applications have not been successfully dealt with using conventional processes. Many biological tissues possess multiple components

S. Lin and Y-S. Lee (✉)

Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, USA
e-mail: yslee@ncsu.edu

R.J. Narayan

Department of Biomedical Engineering, University of North Carolina, Chapel Hill, NC 27599, USA

and complex geometric information. Figure 2.1 shows the boundary surfaces and interface surfaces of the human anterior abdominal wall. As seen in this figure, adipose tissue, skin, and muscle tissues form a multilayered structure in this region of the body. It is difficult to model these structures using conventional approaches. Physically based deformable models are also of growing interest in virtual reality (VR) applications, including surgery, entertainment, design, and manufacturing, since most deformable objects consist of heterogeneous materials with complex internal structures [1–4]. Deformable models must incorporate appropriate internal structures and material properties for situations in which high-fidelity virtual environments are required [5].

Heterogeneous deformable model representation is critical in modeling internal structures of deformable biological objects that possess multiple components with different material properties [6,7]. In previous studies, nonlinear elasticity of deformable objects has been modeled using nonlinear strain tensors or nonlinear spring coefficients [8,9]. Bourguignon [10] and Picinbono [11] reported the modeling of anisotropic behaviors of deformable objects. However, these approaches are only capable of dealing with deformable objects that consist of single material type.

Haptic interfaces have attracted a lot of research interests in the recent years. Haptic interface is an electromechanical device that can provide force feedback to users [12,13]. Haptic force-feedback interface has been used for conceptual design, collaborative design, virtual prototyping, and sculpting in VR environments [14–16]. Heterogeneous deformable object models combined with accurate haptic rendering techniques based on these models provide great potential to enhance interactions with deformable objects. For example, in medicine and surgery it is often necessary to differentiate healthy soft tissues from unhealthy ones [5,17]. However, most current practices assume homogeneous material properties and ignore internal structure variations, which greatly limited their use [18]. In VR environments, new deformable models often need to be generated efficiently. In our earlier work presented in [4,19], haptic-based force rendering techniques have been

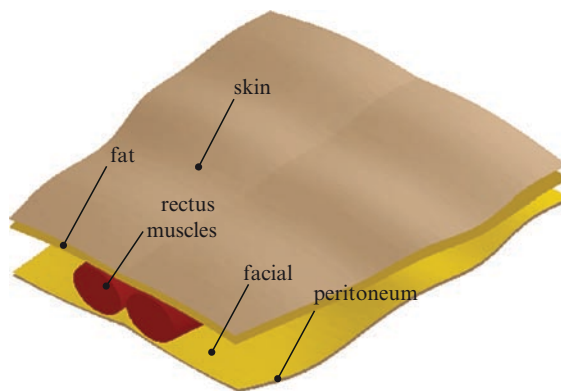


Fig. 2.1 Bio-tissue example and boundary surfaces of anterior abdominal wall

developed for medical surgical applications. For example, patient-specific models of soft tissues may be required for the presurgical rehearsal and postsurgical analysis. However, conventional virtual simulation systems lack appropriate flexibility for creating deformable models, especially heterogeneous deformable models.

In this paper, detailed techniques and modeling algorithms are proposed to generate heterogeneous deformable models for bio-manufacturing and medical applications. A constrained local static integration method is proposed for rapid and robust deformation simulation and a force accumulation model is presented to calculate spring forces for accurate haptic rendering. Parts of human body such as anterior abdominal wall, leg, and thigh are modeled using the proposed techniques for demonstration. Other heterogeneous deformable objects can also be modeled using these techniques.

The remainder of this paper is organized as follows. Section 2.2 provides a review of heterogeneous modeling of deformable objects. Section 2.3 presents the detailed node snapping algorithm for modeling deformable heterogeneous objects. Section 2.4 discusses the system modeling of spring force rendering and the detailed techniques of solving the deformation equilibrium. Section 2.5 presents the computer implementation and practical examples of the proposed methods, followed by the concluding remarks in Sect. 2.6.

2.2 Modeling of Deformable Biological Soft Tissues

Biological soft tissues are viscoelastic, anisotropic, and heterogeneous deformable objects [20]. Deformable model generation is a process of discretizing volumetric objects into mass points and springs in a Mass Spring Model or small elements in a Finite Element Model. Most simulation systems use the mesh generation approaches of Finite Element Analysis (FEA) to generate small elements such as tetrahedra or hexahedra for a Finite Element Model [21]. Many Mass Spring models can also be generated from these tetrahedra or hexahedra by taking their vertices as mass points and edges as springs [22]. However, homogeneous deformable models are generated using these approaches in many cases.

A heterogeneous deformable object may include several materials, as shown in Fig. 2.2. For example, a human leg consists of skin, adipose tissue, muscles, bone, tendons, and other tissues. To generate such a heterogeneous model, the geometry of each material inside the object is required, such as its interface surfaces. Most interface surfaces between materials can be acquired from computed tomography, magnetic resonance imaging, and other techniques, although it remains as a challenge to differentiate some similar tissues.

In some applications like Computer Aided Design (CAD), smooth variation between different materials is traditionally applied by interpolation [23]. For heterogeneous deformable modeling of biological soft tissues, a meaningful interface layer usually exists between two different materials, as shown in Fig. 2.2. For instance, muscles are often enclosed by the muscle sheath and bones are enclosed

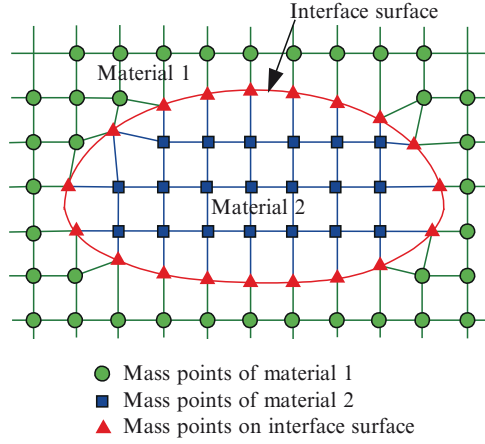


Fig. 2.2 Modeling of heterogeneous deformable bio-objects with interface surface between tissues

by the periosteum. These interface surfaces provide appropriate visual clue to differentiate soft tissues. They can also be used for specifying different material properties such as mass, texture, and stiffness.

Heterogeneous modeling and simulation require more computational power than homogeneous modeling since more topological constraints and material properties are involved. For interactive VR applications (e.g., surgical simulation), real-time computation has the highest priority. As a result, algorithms used in heterogeneous modeling must be as efficient as possible. In addition, the balance between the level of details in heterogeneous modeling and computational efficiency needs to be carefully taken into account.

2.3 Node Snapping Algorithm for Constructing Heterogeneous Models

In this paper, we present a technique of constructing heterogeneous deformable models. The key point of heterogeneous model generation is to retain a single mass spring layer for the outmost surface of a whole object and also individual mass spring layers for each internal interface surface. For the remainder of the deformable model, a uniform mass spring network is created to connect these interface mass spring layers. For a better illustration of the idea of heterogeneous deformable model generation, two definitions are provided below:

Definition 1: Mass points of the same material are defined as homogeneous mass points; otherwise they are heterogeneous mass points.

Definition 2: Springs connecting homogeneous mass points are defined as homogeneous springs; springs connecting heterogeneous mass points are defined as heterogeneous springs.

In this paper, the heterogeneous deformable model generation algorithm is implemented in the following sequence: (1) mass point generation by a tri-ray node snapping algorithm, (2) springs connection and interface marching technique, and (3) physical parameter specification.

From the boundary and interface surfaces, mass points are generated by a tri-ray node snapping algorithm similar to the Ray Casting algorithm in Computer Graphics. On the basis of the concept of the tri-dexel volumetric model developed in our earlier work [12], parallel rays are cast respectively three orthogonal directions to intersect the boundary surface and interface surfaces, to accurately preserve the shapes of these surfaces. Figure 2.3 shows the procedure of using tri-ray tracing

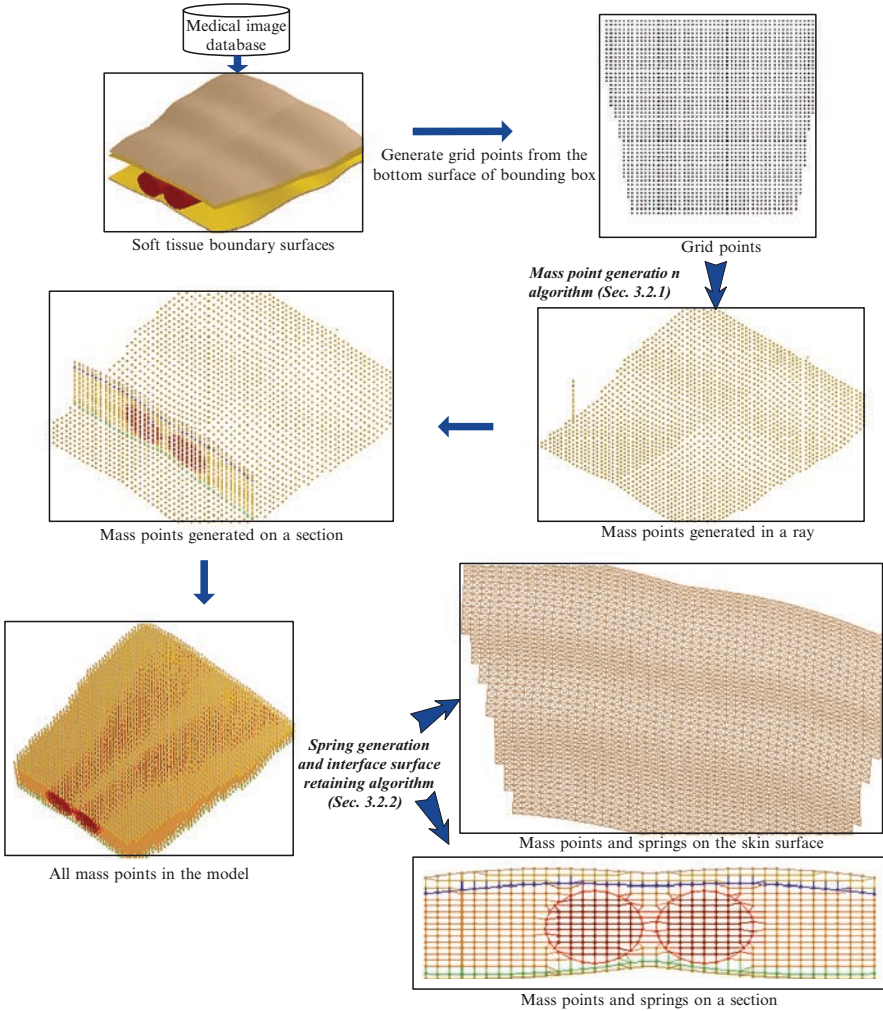


Fig. 2.3 Procedure of constructing heterogeneous volumetric models

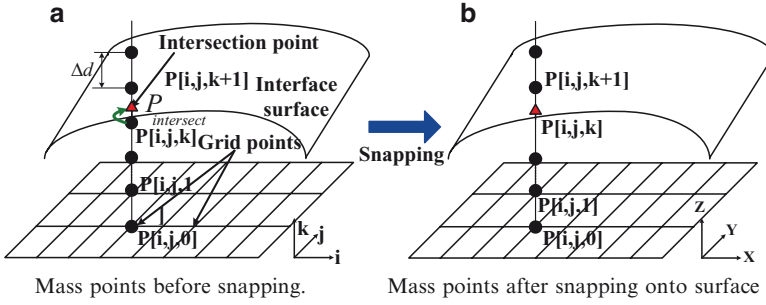


Fig. 2.4 Mass points snapped to intersection surface points, (a) Mass points before snapping, (b) Mass points after snapping onto surface

technique developed in our earlier work in [5] to construct volumetric heterogeneous models and the technique is briefly explained as follows.

First, rays in one direction are used to generate all mass points. Then rays in the other two directions are used to adjust the positions of mass points around interface surfaces, without generating new mass points. As shown in Fig. 2.4, uniform rays are cast from grid points $P[i, j, 0]$ on the two-dimensional X – Y plane to intersect with all the accessible boundary surfaces. Mass points are evenly generated at the interval Δd along the ray as shown below:

$$P[i, j, k + 1] = P[i, j, k] + \Delta d \cdot \vec{r} \quad (2.1)$$

where \vec{r} is a unit vector in the ray direction; Δd is the grid distance between adjacent mass points. Δd is defined by the maximum of d_{\min} and $d_{\text{threshold}}$ as follows:

$$\Delta d = \max\{d_{\text{threshold}}, d_{\min}\} \quad (2.2)$$

In (2.2), d_{\min} is the minimal distance between two intersection points of different interface surfaces in the model, and $d_{\text{threshold}}$ is a preset value to control the mass point density in case d_{\min} is too small. As shown in Fig. 2.4a, when a ray intersects an interface surface at the point $P_{\text{intersect}}$, the closest mass point to $P_{\text{intersect}}$ along the ray is $P[i, j, k]$ if the following condition is true:

$$|P_{\text{intersect}} - P[i, j, k]| \leq \frac{\Delta d}{2} \quad (2.3)$$

Using (2.3), a mass point can be snapped into its closest intersection point on an interface surface. As shown in Fig. 2.4b, $P[i, j, k]$ is the closest mass point to $P_{\text{intersect}}$ and it is snapped to $P_{\text{intersect}}$ as a result. Figure 2.5 shows the flowchart of the detailed procedures for generating mass points in heterogeneous volumetric models.

Using (2.1)–(2.3), mass points generated by the tri-ray procedure can be refined into the intersection points located on the surfaces. Mass points are gradually generated along the ray until the ray reaches the outmost boundary surface. This

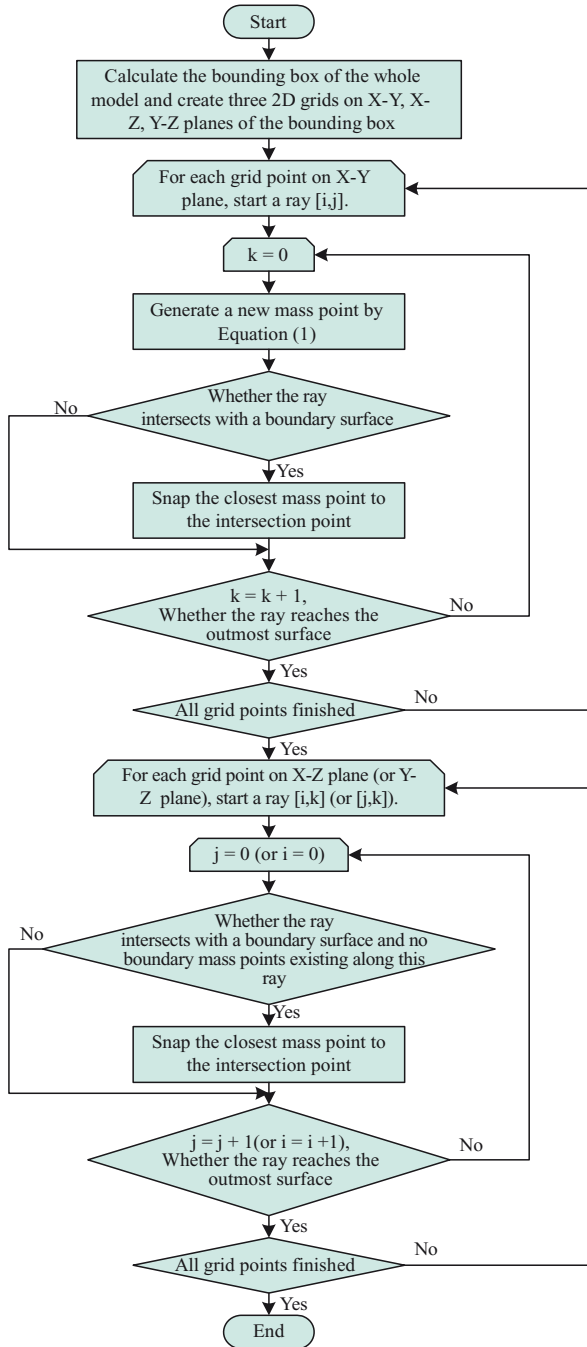


Fig. 2.5 Flowchart of generating the volumetric heterogeneous models

process is repeated until all rays in the Z direction are processed. After this step, newly generated mass points are shown in the X – Z plane of Fig. 2.6a. However, mass points on the interface surface cannot separate Material 1 and Material 2. They also cannot preserve the shapes of original interface surfaces, as shown in Fig. 2.6a.

These problems may be solved by adjusting some mass points around the interface surfaces using rays in the X and Y directions, respectively. For example, rays along the X direction are cast to intersect the interface surfaces. Existing mass points closest to the intersection points are snapped to these intersection points, without generating new mass points. Figure 2.6b shows an example of adjusting mass points along the X direction. Ray $k+1$ intersects the interface surface at point M and point N . Mass point $P[i-1, k+1]$, which is closest to point M , and mass point $P[i+3, k+1]$, which is closest to point N , are snapped to points M and N , respectively. After the mass point adjustment step, a smooth boundary interface is formed to separate the two materials (Fig. 2.6c). Figure 2.7 gives an example of mass points on one section of human thigh model using the tri-ray node snapping algorithm. Larger mass points are located on boundary or interface surfaces. The example shows that the node snapping algorithm can preserve boundary and interface shapes in deformable objects during the construction of heterogeneous deformable models.

Most mass points by the tri-ray node snapping algorithm are uniformly distributed inside the deformable model if they are not snapped onto any interface surface. These mass points maintain a regular neighboring topology so that they can be directly connected to their neighboring mass points by springs. On the contrary, mass points snapped onto interface surfaces are irregularly distributed due to snapping. Mass points snapped on an interface surface are supposed to form a single mass spring layer to represent this interface surface. However, the direct neighboring mass point connection method is not able to achieve this.

Figure 2.8a shows an example of interface points generated by the tri-ray node snapping algorithm. Notice that the new interface points are not located at the exact

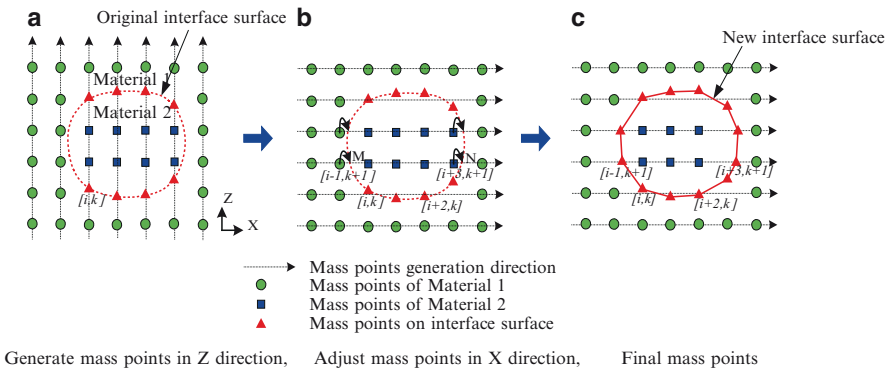


Fig. 2.6 Mass point generation with different bio-tissue materials. (a) Generate mass points in Z direction, (b) Adjust mass points in X direction, (c) Final mass points

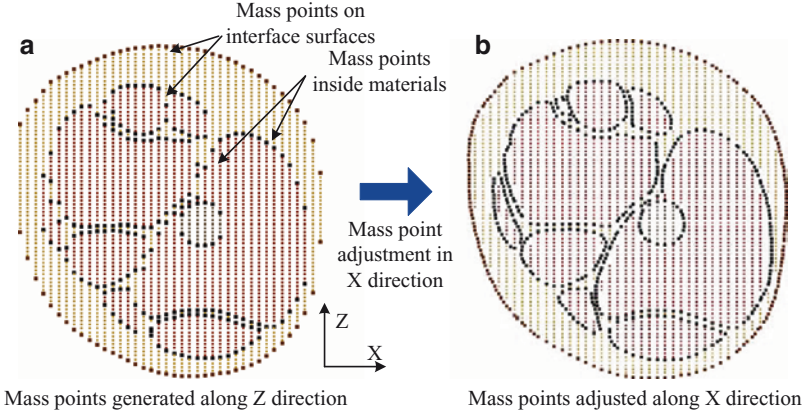


Fig. 2.7 Node snapping algorithm and mass-spring point construction of human thigh example

grid nodes. In this paper, an interface marching technique has been developed to construct the interface surface combined with the node snapping procedure. Using the concept of the marching cube algorithm developed in our earlier work presented in [13], the interface surface is connected through marching along the three-dimensional surface contour. In Fig. 2.8b, the original grid nodes generated by the tri-ray method are snapped onto the interface surface. When marching along the interface contours ceases, the interface contours need to be connected and merged. As shown in Fig. 2.8c, a continuous interface surface has been successfully constructed. Details of the Interface Marching Algorithm are shown in Fig. 2.9.

The last step of heterogeneous deformable model generation is to assign material properties to mass points and springs. In our heterogeneous deformable models, different materials are separated by the interface surfaces. By selecting any mass point, all of its homogeneous mass points can be chosen simultaneously and material properties (e.g., mass) can be specified efficiently, which is an advantage of our heterogeneous model.

Similarly, we can assign physical properties (e.g., elastic modulus) to springs in this model. If a spring is a homogeneous spring that connects two homogeneous mass points, it is either on an interface surface or within a material. Material properties of the interface surface or the material are assigned to the spring. If a spring is a heterogeneous spring, the mass point at one end of the spring must be on an interface surface and the mass point at the other end of the spring belongs to another material. The spring is considered to have material properties of that material. All springs exhibiting the same physical properties can be found starting from any of these springs, since they are always connected with each other. Therefore, their spring properties can be assigned in a batch.

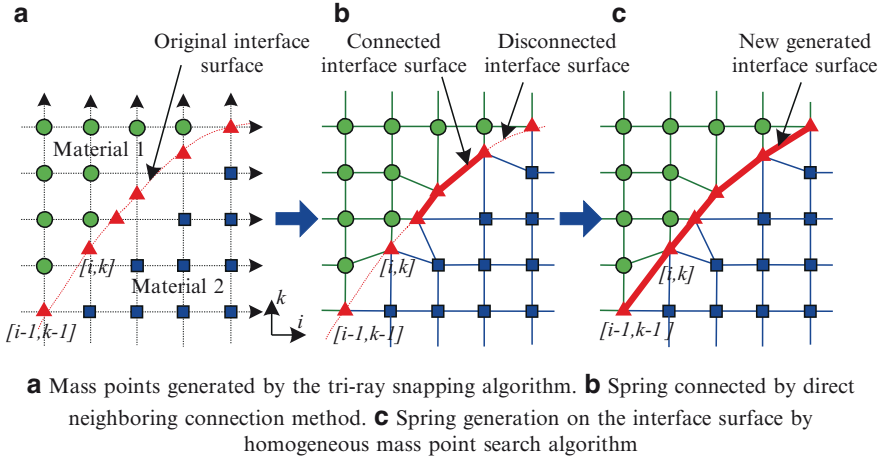


Fig. 2.8 Technique for Interface Surface Marching. (a) Mass points generated by the tri-ray snapping algorithm. (b) Spring connected by direct neighboring connection method. (c) Spring generation on the interface surface by homogeneous mass point search algorithm

2.4 System Modeling of Spring Force Rendering and Solving Deformation Equilibrium

Local deformation of soft tissues is often assumed in surgical simulators to reduce the computational burden of deformable object simulation [24]. In this paper, local deformation is controlled by a preset small force value. Mass points are considered to be active only if their total spring forces are larger than the threshold force. Only active mass points are used in simulation.

In the conventional Mass Spring Model, the spring force of a mass point i is often calculated by total spring forces from its neighboring springs as follows:

$$f_i = \sum_{t=0}^n \frac{k_{it} \Delta l_{it} \vec{d}_{it}}{|\vec{d}_{it}|} \quad (2.4)$$

where k_{it} is the spring coefficient between mass points i and t ; Δl_{it} is the spring length deviation from its original length; $\{\vec{d}_{it}\}$ is the direction vector pointing from mass point i to its neighboring mass point t .

During the simulation, it is possible that a spring will become over-compressed or stretched to its extreme length as spring ij in Fig. 2.10. In this case, (2.4) becomes inaccurate because the spring force will not change even if mass point i (Fig. 2.10) is further pushed down. In the extreme compression or stretching case, we assume that a spring will become a rigid object without break or fracture. A force accumulation method is presented to accumulate the spring force acting on the rigid object. For example, spring ij in Fig. 2.10 is extremely compressed and is considered as a rigid object. The spring force on one end mass point (e.g., mass point j) is accumulated to the other end mass point (e.g., mass point i) so that the total spring force at mass point i is calculated as follows:

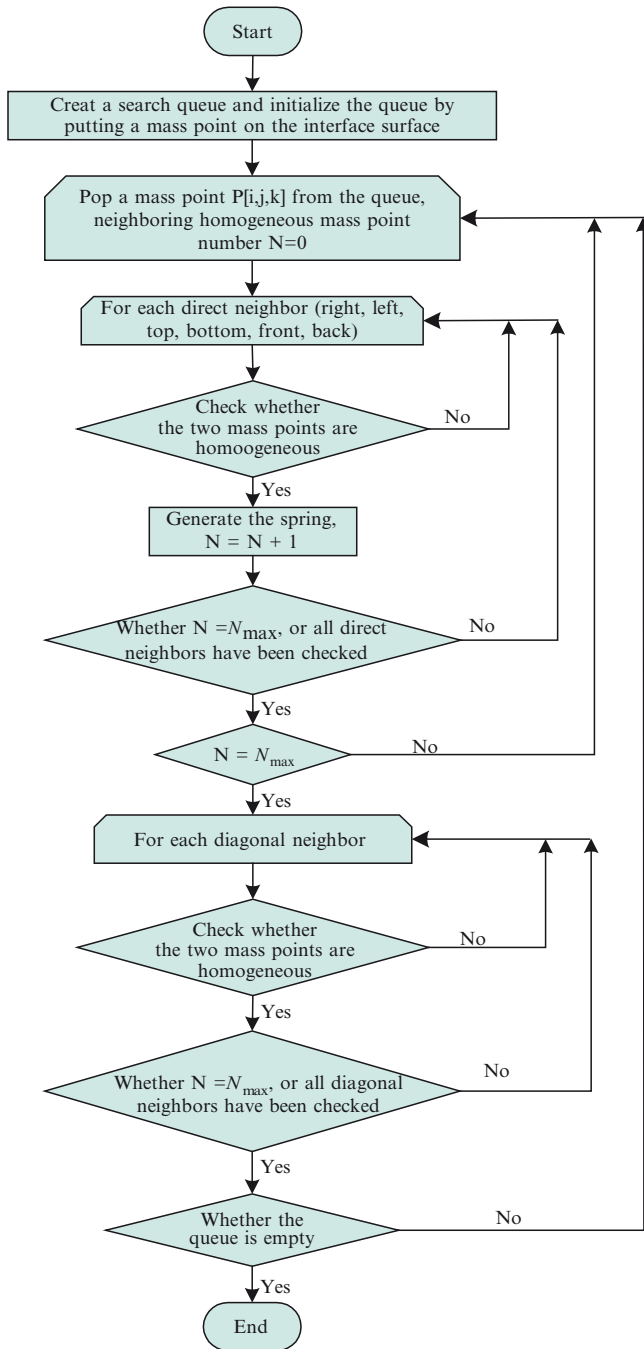


Fig. 2.9 Flowchart of interface marching algorithm

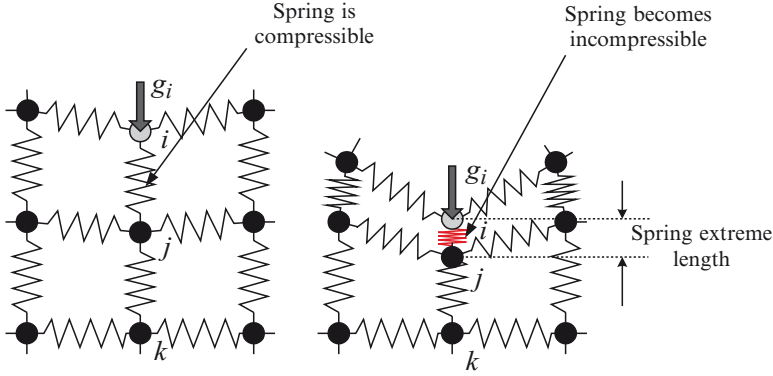


Fig. 2.10 Spring becomes rigid (incompressible) at its extreme length

$$\begin{cases} f_i = \sum_{\{i \in n_i, j \neq i\}} \frac{k_{ij} \Delta l_{ij} \vec{d}_{ij}}{|\vec{d}_{ij}|} + f_j \\ f_j = \sum_{\{i \in n_j, j \neq i, i \neq k\}} \frac{k_{ji} \Delta l_{ji} \vec{d}_{ji}}{|\vec{d}_{ji}|} + f_k \end{cases} \quad \text{if } \Delta l_{ij} \geq \alpha_{\max} l_{ij} \text{ or } \Delta l_{ij} \leq \alpha_{\min} l_{ij}, \Delta l_{jk} \geq \alpha_{\max} l_{jk} \text{ or } \Delta l_{jk} \leq \alpha_{\min} l_{jk} \quad (2.5)$$

where n_i and n_j are the neighboring mass points of mass point i and j . f_k can be calculated recursively using the second formula in (2.5) if there are more extreme springs connecting to mass point k . A mass point touched by the haptic tool is defined as a contact mass point. Haptic feedback forces are calculated by interpolating spring forces of contact mass points. The use of a force accumulation method as well as appropriate material properties from heterogeneous deformable models can be used to provide realistic haptic force rendering for VR applications.

To find the equilibrium state of a deformable object under deformation forces, the governing force system of the deformable object is formulated first. The mass spring system of a deformable object is a dynamic system, which is governed by following second-order differential equations:

$$m_i \ddot{x}_i + d_i \dot{x}_i = g_i + f_i \quad (2.6)$$

where m_i , d_i are mass and damping coefficient of mass point i ; \ddot{x}_i and \dot{x}_i are its acceleration and velocity, respectively; g_i is total external force and f_i is total internal spring force of mass point i . Several numerical integration schemes have been used to solve these Ordinary Differential Equations, such as explicit Euler method, fourth-order Runge–Kutta method and Verlet method [25,26]. All these numerical integration schemes suffer the same problem of slow convergence to system equilibrium, since each mass point may oscillate around its equilibrium position for a few times. A large integration time step h can accelerate convergence but may bring system instability.

In surgical techniques, it is often observed that dynamic behaviors of biological soft tissues cannot be readily determined by means of conventional surgical manipulation [26]. Biological soft tissues can reach their static equilibrium quickly. A quasi-static algorithm was used to find the static solution without considering inertia, mass, or damping [27]. However, this algorithm still needs to find a proper integration step experimentally to balance fast convergence and real-time performance.

In our earlier work presented in [19], a constrained local static integration method was developed for quick convergence and stable simulation. Static equilibrium of each moving mass point is calculated locally within its neighboring bounding box (NBB), as shown in Fig. 2.11. The NBB of mass point $[i, j, k]$ is defined by its six direct neighboring mass points (Fig. 2.11). Movement of each mass point is constrained within the NBB by using the following conditions:

$$\begin{cases} x_{[i-1,j,k]} \leq x_{[i,j,k]} \leq x_{[i+1,j,k]} \\ y_{[i,j-1,k]} \leq y_{[i,j,k]} \leq y_{[i,j+1,k]} \\ z_{[i,j,k-1]} \leq z_{[i,j,k]} \leq z_{[i,j,k+1]} \end{cases} \quad (2.7)$$

The local static equilibrium force method is applied only to aforementioned active mass point i shown as follows:

$$\sum_{j=0}^{n_i} K_{ij} [P_j - P_i - (P_{0,j} - P_{0,i})] = g_i \quad (2.8)$$

where P_i is the current position of mass point i to be calculated; K_{ij} is the spring coefficient of spring ij ; P_j is the known current position of mass point j ; $P_{0,i}$ and $P_{0,j}$ are the initial positions of mass points i and j ; g_i is the external force on mass point

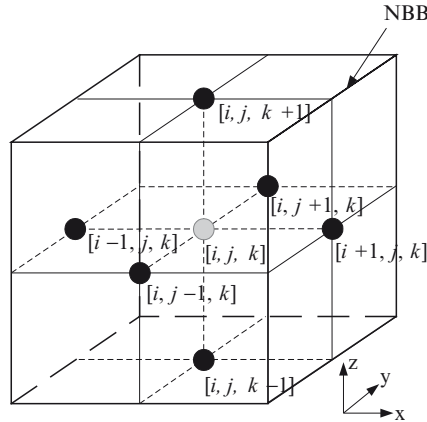


Fig. 2.11 Neighboring bounding box (NBB) of a mass point

i ; and n_i is the number of the neighboring mass points of mass point i . The unknown position P_i in (2.8) can be expressed as follows:

$$P_i = \frac{-g_i + \sum_{j=0}^{n_i} K_{ij} [P_j - (P_{0,j} - P_{0,i})]}{\sum_{j=0}^{n_i} K_{ij}} \quad (2.9)$$

The static equilibrium for P_i is a local and temporary solution, which may be changed according to the dynamic perturbation of the nodes in the deformable models. For example, the variable P_j in (2.9) may be changed due to the dynamic movement of P_i resulting from the change in the local static equilibrium. In this paper, an iterative procedure is applied to (2.9) with the iteration index t shown as follows:

$$P_i^{t+1} = \frac{\sum_{j=0}^{n_i} K_{ij} [P_j^t - (P_{0,j} - P_{0,i})]}{\sum_{j=0}^{n_i} K_{ij}} \quad \text{where } t = 0, 1, 2, \dots \quad (2.10)$$

During determination of the final equilibrium, each mass point moves toward its final solution, normally within a few iterations. When some active mass points only move very short distance from their last iterative locations and the responsive spring force is smaller than a predefined threshold value, these mass points are considered to have reached the final equilibrium status. When no more active mass points are present, the iteration is completed and the final equilibrium of the system is accomplished.

2.5 System Implementation and Examples

The presented techniques have been implemented at North Carolina State University on a 2.4 GHz dual-CPU workstation with 2 GB memory and a high-end graphics card, using Visual C++ .Net 2003 and OpenGL® library. The system is integrated with a 6-DOF (degree of freedom) Phantom® desktop haptic device, which provides 3-DOF force feedback. Figure 2.12 shows the lab setup of the implemented system with the 6-DOF haptic interface. An example heterogeneous deformable model was deformed by using the haptic device, as shown in Fig. 2.12. Multiple threads technique has been applied to deal with haptic rendering, graphics, and collision detection, respectively.

Figure 2.13 shows a procedure of using the presented techniques to construct the heterogeneous deformable model using medical imaging data. Using the presented techniques, a model of human right thigh is generated, as shown in Fig. 2.13. Skin, facial, femur, and nine types of muscles are integrated into one heterogeneous thigh

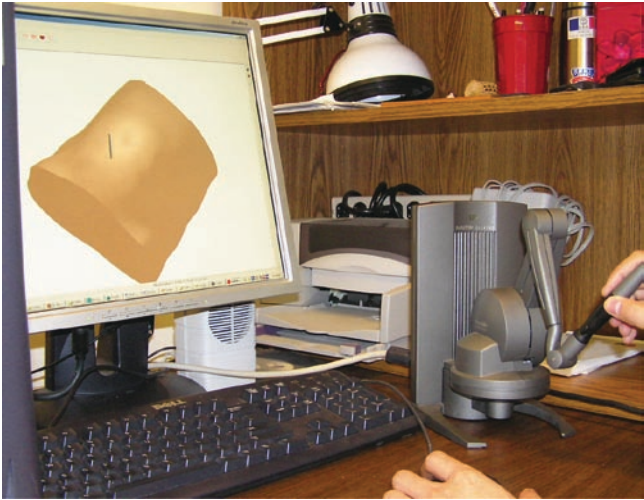


Fig. 2.12 Lab-setup of heterogeneous deformable modeling with a haptic force-feedback device

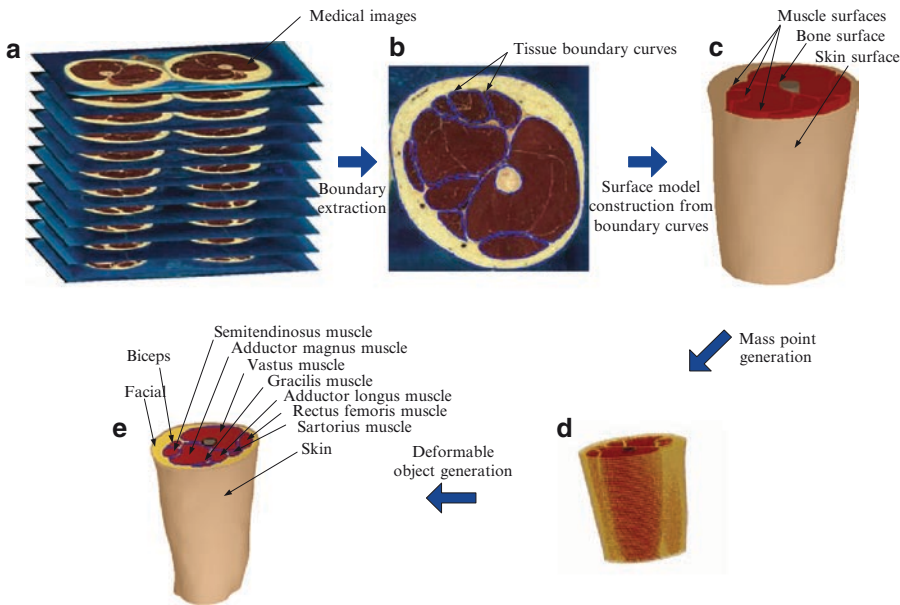


Fig. 2.13 Construction of a heterogeneous volumetric model from medical images

model. First, color medical images of different height sections are placed in a layer-by-layer manner as shown in Fig. 2.13a. The boundary curves of soft tissues and femur in the right thigh are sketched using splines as shown in Fig. 2.13b. From boundary curves on different layers, surface models of each tissue are constructed as shown

in Fig. 2.13c. The tri-ray node snapping algorithm is then used to generate the mass points (Fig. 2.13d). During the mass point generation process, a flag is set for each mass point to indicate which tissue it belongs to. Figure 2.13e shows the final rendered heterogeneous deformable model.

Figure 2.14 shows a detailed profile of the deformation using a small compressive force via a haptic device. Several tissue layers including skin, subcutaneous fat, fascia, and muscles are involved in the deformation. The deformation is propagated from the model surface into the heterogeneous model. The upper layers exhibit larger deformation than deeper layers, as shown in Fig. 2.14. On the basis of our proposed techniques, no penetration happens between tissue layers during the deformation. The result is consistent with deformation behavior of actual soft tissues.

Figure 2.15a shows another heterogeneous deformable model of human anterior wall. This model includes 4,248 triangles on the skin layer and a total of 53,199 mass points inside the model. Tissue types are indicated by different colors, and interface surfaces between soft tissues are represented by bold lines. As shown in Fig. 2.15, a haptic tool is manipulated to push against the abdomen, similar to the interactive palpation on a patient's abdomen by medical examiners. As shown in Fig. 2.15a–c, an experiment is performed by pushing the model using the haptic tool along two paths with different depths. Figure 2.15d, e shows the resulting forces along the two paths. When the tool is moved just underneath the skin, there is no apparent force change because only fat tissues are deformed slightly (Fig. 2.15d). When the tool is pushed deep enough as shown in Fig. 2.15e, muscles are indirectly involved in the deformation process. Since muscles exhibit much higher stiffness values than skin or fat, a larger force is generated. As a result, the user can feel the existence of muscles by force feedback. Due to hand shaking during movement, the tool paths as well as the resulting forces demonstrate some fluctuation.

Figure 2.16 shows an example of haptic manipulation of a human right leg model, in which bone is involved. Bone is a rigid inorganic–organic composite material, which is generally considered to be incompressible. When soft tissues are gradually pushed against the bone surface, soft tissues will be compressed to their mechanical limit. Once the soft tissues can no longer be compressed, forces will increase dramatically if the user continues to push against the model. In this case, users can feel the bone using a haptic device.

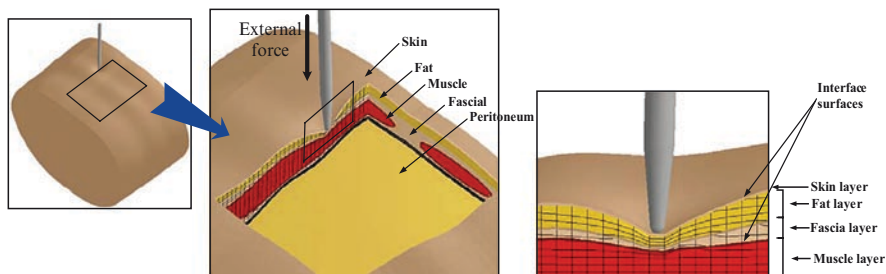
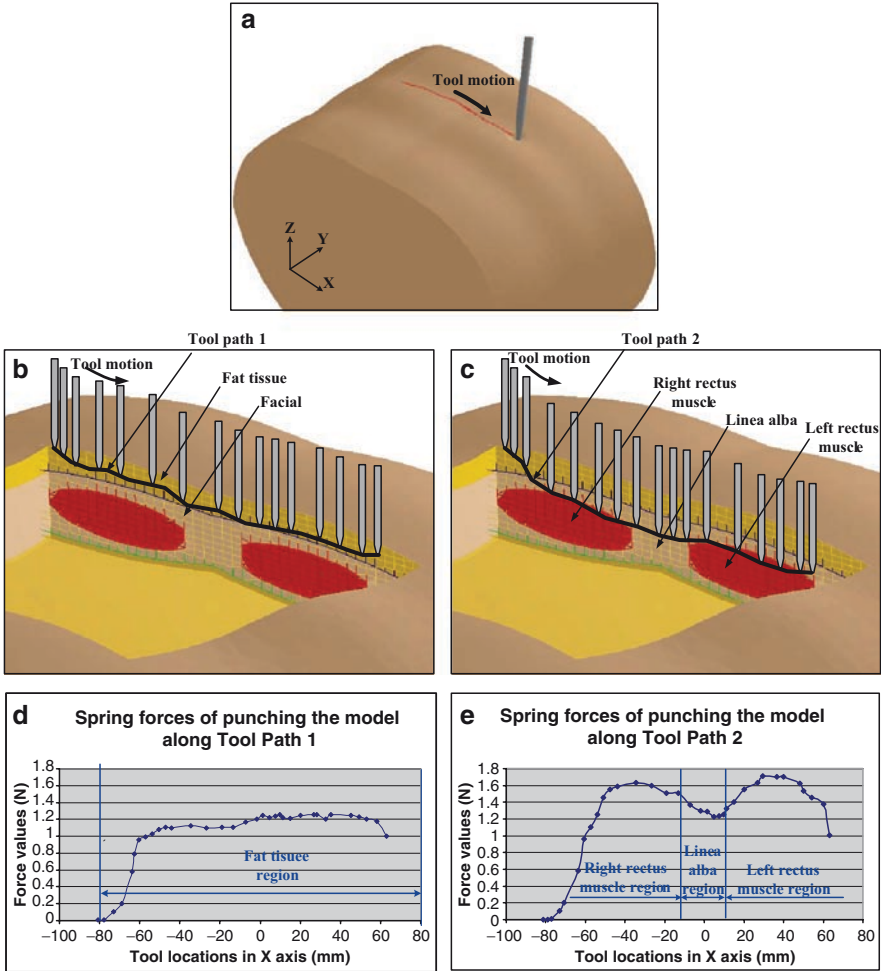


Fig. 2.14 Example of deformation on a heterogeneous anterior abdominal model via the haptic interface



a Deformation of a model. **b** Fat tissues pressed along Path 1. **c** Muscles pressed along Path 2. **d** Response forces along Path 1. **e** Response forces along Path 2.

Fig. 2.15 Response forces via the haptic interface when pushing a tool along the different trajectories on the model. **(a)** Deformation of a model. **(b)** Fat tissues pressed along Path 1. **(c)** Muscles pressed along Path 2. **(d)** Response forces along Path 1. **(e)** Response forces along Path 2

2.6 Conclusions

In this paper, modeling of soft biological tissues and development of a haptic-based force-feedback interface for bio-manufacturing and medical applications have been presented. Detailed techniques for constructing heterogeneous deformable models were discussed for representing the internal structures of deformable biological

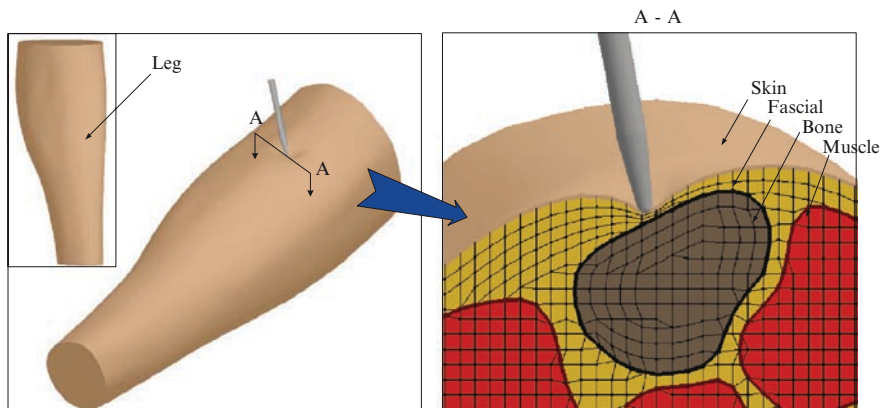


Fig. 2.16 Deformation on an example of human leg model with rigid bones via the haptic force feedback interface

objects that possess multiple components and nonuniform material properties. A tri-ray node snapping algorithm was presented to generate a volumetric heterogeneous deformable model from a set of object interface surfaces between different materials. A constrained local static integration method was presented for simulating deformation and accurate force feedback for heterogeneous structures. The presented heterogeneous deformable object modeling and haptic rendering can greatly enhance the realism and fidelity of virtual reality environments. By integrating the heterogeneous deformable model into the virtual environments, users can both visually see different materials inside the deformable objects as well as feel them when touching the deformable object via a haptic device. The presented techniques may be used for several technological applications, including surgical simulation, bio-product design, bio-manufacturing, and medical applications.

Acknowledgment This work was partially supported by the National Science Foundation (NSF) Grants (DMI-0300297, CMMI-0553310) to North Carolina State University. Their support is greatly appreciated.

References

1. Akagi Y, Kitajima K (2006) Computer animation of swaying trees based on physical simulation. *Comput Graph* 30(4):529–539
2. Al-khalifah A, Roberts D (2004) Survey of modeling approaches for medical simulators. In: *Proceedings of 5th Intl Conf. Virtual Reality & Assoc. Tech.*, Oxford, UK, pp 321–329
3. Dewaele G, Cani M-P (2004) Interactive global and local deformations for virtual clay. *Graph Models* 66:352–369
4. Lin S, Lee Y-S, Narayan R (2008) Heterogeneous material modeling and virtual prototyping with 5-DOF haptic force feedback for product development. *Int J Mechatronics and Manuf Syst* 1(1):43–67

5. Lin S, Lee Y-S, Narayan R, Shin H. Bio-tissues modeling and interface development for bio-manufacturing and medical applications. In: Proceedings of 2007 Industrial Engineering Research (IERC) conference, Nashville, TN, 19–23 May 2007, pp 281–286
6. Georgii J, Westermann R (2005) Interactive simulation and rendering of heterogeneous deformable bodies. In: VMV 2005, Erlangen, Germany, 16–18 Nov 2005
7. Sami S, Maciel A, Boulic R, Thalmann D (2004) Evaluation and visualization of stress and strain on soft biological tissues in contact. In: Proceedings of shape modeling conference, pp 255–262
8. Desbrun M, Schroder P, Barr A (1999) Interactive animation of structured deformable objects. In: Proceedings of 1999 conference on graphics interface, pp 1–8
9. Mendoza C, Laugier C (2003) Tissue cutting using finite elements and force feedback. In: Proceedings of the international symposium on surgery simulation and soft tissue modeling, pp 175–182
10. Bourguignon D, Cani M (2000) Controlling anisotropy in mass-spring systems. In: Proceedings of the 11th eurographics workshop, Interlaken, Switzerland, Aug 2000, pp 21–22
11. Picinbono G, Delingette H, Ayache N (2003) Non-linear anisotropic elasticity for real-time surgery simulation. *Graph Models* 65:305–321
12. Ren Y, Lai-Yuen SK, Lee Y-S (2006) Virtual prototyping and manufacturing planning by using tri-dexel models and haptic force feedback. *Virtual Phys Prototyping* 1(1):3–18
13. Zhu W, Lee Y-S (2005) A visible sphere marching algorithm of constructing polyhedral models from dixel models for haptic virtual sculpting. *Robot Comput Integrated Manuf* 21(1):19–36
14. Lin S, Lee Y-S, Narayan R (2007) Collaborative haptic interfaces and distributed control for product development and virtual prototyping. In: Proceedings of ASME manufacturing science and engineering conference. ASME MSEC2007-31214
15. Peng X, Zhang W, Leu MC (2006) Freeform modeling using sweep differential equation with haptic interface. *Virtual Phys Prototyping* 1(3):183–196
16. Ye J, Campbell RI (2006) Supporting conceptual design with multiple VR based interfaces. *Virtual Phys Prototyping* 1(3):171–181
17. Gibson I (2006) Rapid prototyping: from product development to medicine and beyond. *Virtual Phys Prototyping* 1(1):31–42
18. Basdogan C, De S, Kim J, Muniyandi M, Kim H, Srinivasan MA (2004) Haptics in minimally invasive surgical simulation and training. *IEEE Comput Graph Appl* 24(2):56–64
19. Lin S, Lee Y-S, Narayan R (2007) Heterogeneous soft material modeling and virtual prototyping with 5-DOF haptic force feedback for product development. In: Proceedings of 2007 international conference on advanced research in virtual and physical prototyping (VRAP2007), VRAP: 187–193
20. Fung YC (1993) *Biomechanics: mechanical properties of living tissues*, 2nd edn. Springer-Verlag, New York
21. Meier U, Lopez O, Monserrat C, Juan MC, Alcaniz M (2005) Real-time deformable models for surgery simulation: a survey. *Comput Methods Programs Biomed* 77:183–197
22. Mollemans W, Schutyser F, Cleynenbreugel JV, Suetens P (2003) Tetrahedral mass spring model for fast soft tissue deformation. In: Proceedings of surgery simulation and soft tissue modeling: international symposium, IS4TM, pp 145–154
23. Samanta K, Koc B (2005) Feature-based design and material blending for free-form heterogeneous object modeling. *Comput Aided Des* 37:287–305
24. Nakao M, Kuroda T, Oyama H, Sakaguchi G, Komeda M (2006) Physics-based simulation of surgical fields for preoperative strategic planning. *J Med Syst* 30(5):371–380
25. Bielser D, Maiwald VA, Gross MH (1999) Interactive cuts through 3-dimensional soft tissue. *Comput Graph Forum* 18(3):31–38
26. Sørensen TS, Mosegaard J (2006) An introduction to GPU accelerated surgical simulation. In: The 3rd symposium on biomedical simulation, Zurich, Switzerland, lecture notes in computer science (series No. 4072), pp 93–104
27. Brown J, Sorkin S, Bruyns C, Latombe JC, Montgomery K, Stephanides M (2001) Realtime simulation of deformable objects: tools and application. In: Computer animation, Seoul, Korea, 7–8 Nov 2001, pp 228–236

Printed Biomaterials

Novel Processing and Modeling Techniques for
Medicine and Surgery

Narayan, R.; Boland, Th.; Lee, Y.-S. (Eds.)

2010, XIV, 124 p. 162 illus., 73 illus. in color., Hardcover

ISBN: 978-1-4419-1394-4