

A Novel Modeling Framework for Multilayered Soft Tissue Deformation in Virtual Orthopedic Surgery

Jing Qin · Wai-Man Pang · Yim-Pan Chui ·
Tien-Tsin Wong · Pheng-Ann Heng

Received: 12 September 2008 / Accepted: 1 December 2008 / Published online: 24 December 2008
© Springer Science + Business Media, LLC 2008

Abstract Realistic modeling of soft tissue deformation is crucial to virtual orthopedic surgery, especially orthopedic trauma surgery which involves layered heterogeneous soft tissues. In this paper, a novel modeling framework for multilayered soft tissue deformation is proposed in order to facilitate the development of orthopedic surgery simulators. We construct our deformable model according to the layered structure of real human organs, and this results in a multilayered model. The division of layers is based on the segmented Chinese Visible Human (CVH) dataset. This enhances the realism and accuracy in the simulation. For the sake of efficiency, we employ 3D mass-spring system to our multilayered model. The nonlinear passive biomechanical properties of skin and skeletal muscle are achieved by introducing a bilinear elasticity scheme to the springs in the mass-spring system. To efficiently and accurately reproduce the biomechanical properties of certain human tissues, an optimization approach is employed in configuring the parameters of the springs. Experimental data from biomechanics literatures are used as benchmarking references. With the employment of Physics Processing Unit (PPU) and high quality volume visualization, our framework is

developed into an interactive and intuitive platform for virtual surgery training systems. Several experiments demonstrate the feasibility of the proposed framework in providing interactive and realistic deformation for orthopedic surgery simulation.

Keywords Multilayered deformable model · Bilinear modeling · Virtual orthopedic surgery

Introduction

Traditional ways of training medical procedures on live patients is paradoxical because the training process itself may imperil the health of patients. Practising these procedures on cadavers lacks tissue realism. The limited availability of cadavers is another big concern. The use of animals cannot be considered as the perfect substitutes of human body due to the anatomical difference between animal and human as well as some other ethic issues.

In recent decades, virtual-reality (VR) based medical simulations have become more and more prevalent considering the system reusability, the cost effectiveness and the better learning efficiency they offered to medical education. Considerable research efforts have been dedicated to the development of computer-assisted surgical simulators that enable doctors and students to learn human anatomy and surgical skills in a virtual environment. These simulators could assist in surgical planning, training and rehearsal, as well as the investigation of new medical approaches.

To realistically model human tissues, deformation is an indispensable requirement of every surgical simulator. Deformation models must consider the needs

Electronic supplementary material The online version of this article (doi:10.1007/s10916-008-9237-6) contains supplementary material, which is available to authorized users.

J. Qin (✉) · W.-M. Pang · Y.-P. Chui ·
T.-T. Wong · P.-A. Heng
Department of Computer Science and Engineering,
The Chinese University of Hong Kong,
Shatin N. T., Hong Kong
e-mail: jqin@cse.cuhk.edu.hk

of surgical simulator, including prompt response to interactive operation, precise preservation of tissue behavior, efficient computation of reaction forces for manipulative devices and realistic render the deformed appearance. Although a lot of deformable models have been proposed, development of an interactive simulator still retains a challenging task due to the complexity of the physics-involved computation, especially when satisfying realism should also be guaranteed. The problem becomes more complicated in orthopedic trauma surgery simulation where multiple tissues such as bones, skin, adipose tissues and muscle, of which the biomechanical properties are heterogeneous, are involved.

Recently, a specialized accelerator, named Physics Processing Unit (PPU), is developed to provide computing power for dynamic motion and interaction, and accordingly to highly accelerate the physically based simulations [1]. PPU has been successfully used to simulate real physics with the scale, sophistication, fidelity and level of interactivity that dramatically alters in many physically based computation-intensive games. In this paper, we exploit the relative computational advantage of PPU to simulate the non-linear stress-strain mechanical behavior of soft tissues. To the best of our knowledge, this work is the first to use PPU in surgical simulation system.

On the other hand, in order to conform to the realism of soft tissue behaviors, we must take biomechanics into consideration. The mechanical behavior of soft tissues is closely related to their layered structure, multi-component composition, and location. Therefore, a deformable model generator is proposed to automatically generate the multilayered structure of soft tissues and incorporate related biomechanical information for deformation and visualization. Furthermore, a bilinear model is employed to simulate the non-linear stress-strain behavior of soft tissues. To ensure that the macroscopic stress-strain curve of our model is consistent with real biomechanical data, simulated annealing is used to optimize parameters of individual springs in the model.

To ensure the anatomical fidelity, segmented Chinese Visible Human (CVH) datasets are adopted to implement a prototype in order to demonstrate the validity of the proposed framework. Experimental results show our system can rapidly solve complex mass-spring model in virtue of PPU while realistic deformation can be achieved by carefully planning the mesh structure based on the CVH datasets and optimizing and configuring the properties of the masses and springs according to the non-linear mechanical curves. With an integration of high quality volume visualiza-

tion, our framework can serve as an intuitive platform for the development of orthopedic surgery simulation.

In summary, our novel modeling framework has following properties:

- First, we construct our deformable model semi-automatically according to the layered structure of real human organs, and this results in a multilayered model. This multilayered model are constructed based on Chinese Visible Human dataset, which can provide more anatomical details than magnetic resonance imaging (MRI) and computer tomography (CT) data usually used in previous orthopedic simulators.
- Second, an efficient bilinear model is developed to simulate the non-linear stress-strain behavior of soft tissues and simulated annealing is used to optimize the spring parameters of this model.
- Finally, to the best of our knowledge, this is the first work to use PPU in surgical simulation system and our experiment results demonstrate that PPU can greatly improve the simulation performance.

The rest of this paper is organized as follows. In Section “[Background](#)”, we will brief the background related to this work. The design and implementation details of our framework are given in Section “[Framework](#)”. Section “[Results](#)” shows the prototype developed based on our framework and experiments conducted on it. Finally, conclusion is drawn with some discussion in Section “[Discussion and conclusion](#)”.

Background

Many academic and commercial virtual reality simulators related to orthopedics had been proposed for the training of surgeons [2–4]. Most of the available orthopedic simulators can be categorized based on the nature of surgery involved, namely arthroscopy [5] and trauma surgery [6]. Most of these simulators focus on the interactivity and training procedures, while the presented realism is compromised because of the limited computational resources and the lack of reference to realistic mechanical and anatomical data. For example, the deformation of all tissues usually looks similar, but not behaves the same as the real tissue.

Hardware acceleration for rendering has long been used in many graphics applications including surgical simulation. Similar to rendering, physically based deformation is another essential and computational intensive component in surgical simulation. Many researchers [7–9] had tried to exploit the power of programmable GPU for physics computation and obtained

impressive increase in time performance. However, the model’s topology usually has to be carefully designed and considerable amount of computation still need to be done on CPU ahead. This restricts the use of some complex topology of human tissues. On the other hand, most of surgical simulators involve both graphic rendering and physical simulation. If GPU is adopted to accelerate both of them, the load of GPU will increase and the whole system performance will accordingly be influenced. Recently, a specialized hardware accelerator for physics—Physical Processing Unit (PPU) is released for complex dynamic motion, interaction and physics computation [1]. Its internal read/write memory bandwidth, processing cores for physics computation and special memory architecture provides a new choice to accelerate physically based computation in surgical simulation systems.

To ensure anatomical fidelity, we construct our multilayered deformable model based on Chinese Visible Human dataset. The advent of the visible human (VH) datasets (Visible Human Project (VHP) [10], Visible Korean Human (VKH) [11], the Chinese Visible

Human (CVH) [12, 13]) has provided standardized digital human data sets for virtual reality based medical education and simulations since they can provide more anatomical details than magnetic resonance imaging (MRI) and computer tomography (CT) data.

Lots of efforts have been devoted to the biomechanical modeling of human soft tissue [14, 15] in the last decades. Most of them focus mainly on the methodologies in solving complicated differential equations and the corresponding numerical correctness and efficiency. However, seldom of them discuss about a systematical method in configuring parameters in the models, so that consistent biomechanical behavior of real tissue is maintained. Motivated by this, we propose a semiautomatic approach in configuring the multilayered mass-spring model.

Framework

Our modeling framework is shown in Fig. 1. The whole framework can be divided into two main modules: the

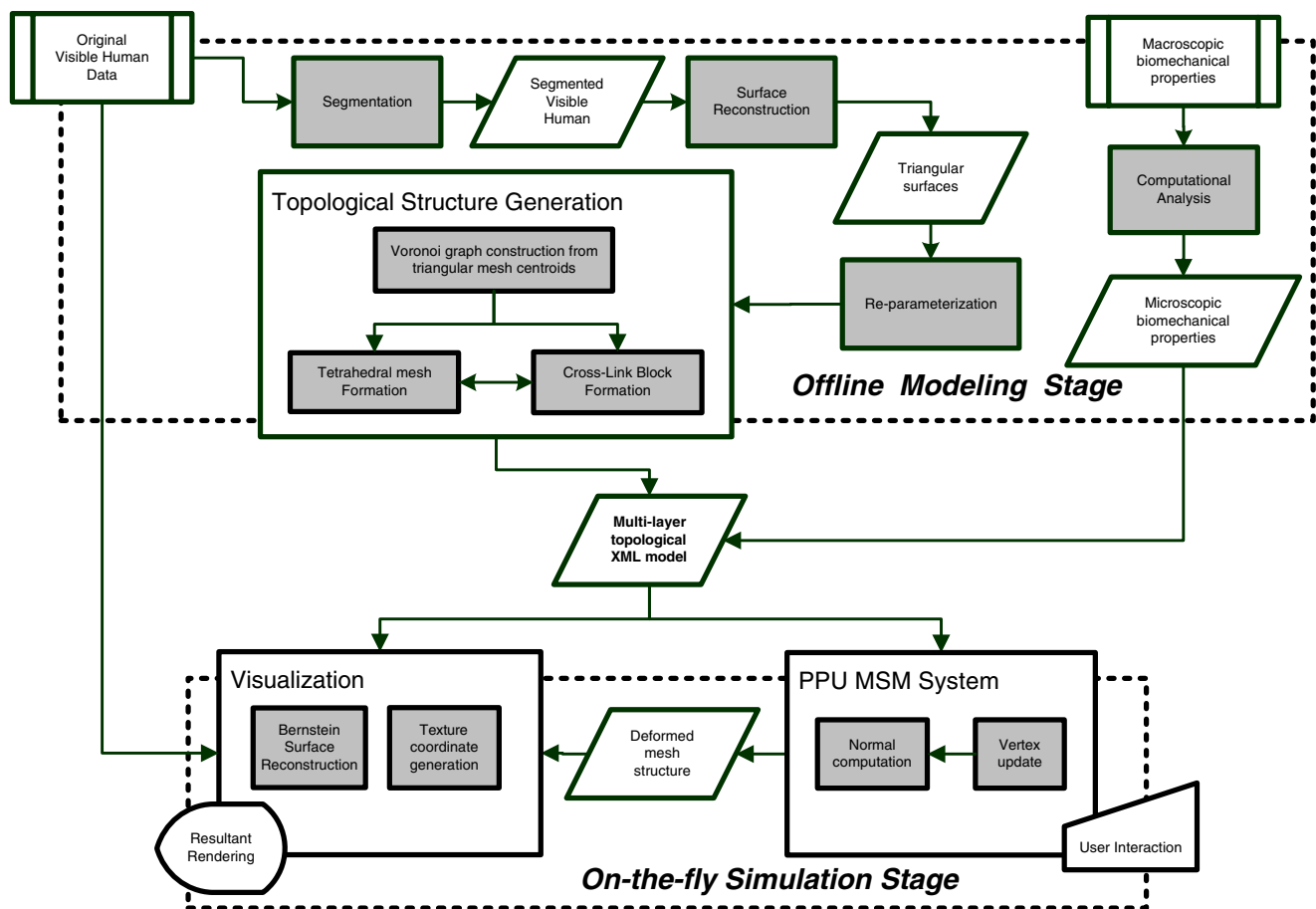


Fig. 1 The overall framework

Topological Structure Generation and the PPU-based Mass Spring System. After identifying various tissues based on visible human anatomical images, surface meshes are extracted from the segmented data. A reparametrization process is carried out on the preliminary surface meshes in order to construct the computationally efficient model. Afterwards, a mass-spring topological structure is automatically generated based on the multilayered structure of soft tissue. Meanwhile, the macroscopic and microscopic biomechanics of soft tissue will be analyzed. A simulated annealing module is performed off-line in order to pre-compute necessary bilinear spring parameters. Both the connectivity information and the biomechanical properties are stored within the same model. Our multilayered topological model is stored in an XML format for the ease of extendability. This topological model is then fed into the PPU-based Mass Spring System for physically based deformation computation. Finally, some auxiliary modules such as edges extraction, normal re-computation, volume texturing as well as Bézier grid patching are implemented for high quality volume visualization.

Deformable model generation

Segmentation of visible human data

Different tissues or organs exhibit different biomechanical properties. Therefore, partitioning these tissues or organs out of one particular image modality is important for reconstructing topological structure and configuring corresponding mechanical parameters. To identify different anatomical structures from the visible human cryosectional slices, we make use of semiautomatic segmentation methods. Skin, adipose tissues, muscle, bone, etc are labeled for further geometric modeling.

Geometric modeling of tissue surface

Geometric modeling can be done by representing structural topology through a number of methods. In surgical simulation applications, polygonal meshes are one of the most common modeling methods due to its simplicity and easy incorporation of computational models. In our framework, we first extract surface mesh models from the segmented data and then proceed on reconstructing volumetric mesh model. Polygonal surface is extracted as Stereolithography (STL) format, a widely adopted industrial standard input for rapid prototyping, which is a list of the triangular facets that describe a computer generated solid model. In-between every two

adjacent tissues, for example, skin and adipose tissues, one single surface layer is constructed.

Although the extracted surface conforms to the actual anatomical structure, the “unstructured” property of the stereolithography facets may introduce difficulties on generating computationally efficient deformable models afterwards. In this sense, we carry out a reparameterization process on the preliminary STL meshes. A set of rays are shot from some regularized grid and terminated at a centralized line segment. A set of sampling points lying on the original STL mesh can then be taken into another re-meshing step. Since these sampling points are distributed in a regular manner, the remeshing process would be a direct triangulation of the grid-clustered points without any extra sorting process. Using such a reparameterization process, we do not have to perform any geometric mesh simplification such as progressive mesh and other manual smoothing procedures. This ensures the automatization and efficiency of our Deformable Model Generator (DMG).

Multilayered topological structure reconstruction

Having obtained all tissue surfaces, multiple surface meshes can be reorganized to form a volumetric multilayered structure. The DMG generates the multilayered structure and incorporates related biomechanics information for deformation and visualization. The data flow of topological reconstruction is shown in Fig. 2. The input of this multilayered topology reconstruction is the reparameterized surface meshes obtained in our previous geometric modeling step. The output is an XML file containing the mechanical properties of masses and springs as well as the connectivity information. The utilization of XML format ensures the flexibility in the exchange of data throughout our framework.

To begin the model generation, we first compute the centroid of every triangle of the surface mesh (Fig. 2a).

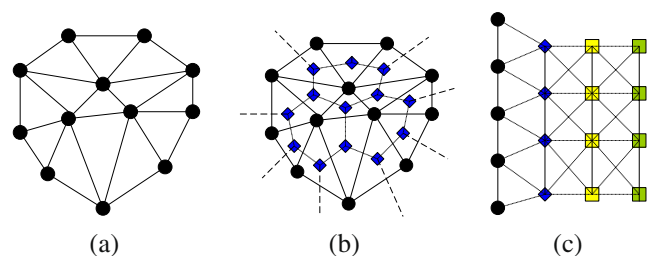


Fig. 2 Multilayered modeling **a** base triangular mesh, **b** centroid-connected voronoi graph, and **c** cross sectional view of tetrahedral block and cross-linked cubic block

From every centroid, bottom masses of the first layer are generated by shooting ray from the centroid towards the centreline until hitting the next layer. Thus, the number of new masses is equal to the number of triangles of the surface mesh. We make use of a voronoi graph topology to organize the new masses at the bottom of the first layer. Every mass of the surface mesh has a queue storing the masses generated from the triangles with it as a vertex. After finishing generating all new masses, every queue are sorted based on the triangle ID of the surface mesh and the sorted masses are connected one by one. Thus, the bottom of the first layer forms a surface composed of many polygons (Fig. 2b). We triangulate every polygon according to global topology and connect top and bottom masses of the first layer to create a skin layer formed by tetrahedrons, while cross-linked cubic block are constructed by crossly connecting the masses between two different layers to create the following layers (Fig. 2c).

During the procedure, an edge table is used to track the connection information of the masses. Based on the connection information as well as the properties of masses and springs, an XML file describing the geometrical model is generated for deformation and visualization modules. Some auxiliary modules such as boundary extraction algorithm, masses density adjustment and element property setting are provided for the flexibility and practicability in generating the geometrical model. Figure 3 shows cylindrical and plane surface mesh, and the corresponding multilayered reconstructed models generated from DMG. The mul-

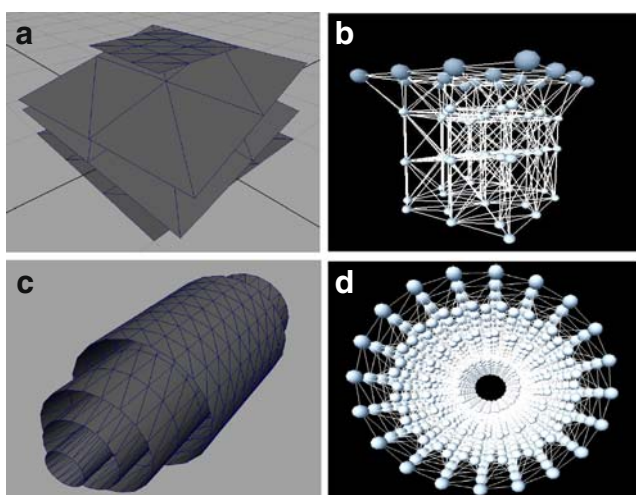


Fig. 3 Examples of the reconstruction process from regularized OBJ meshes to volumetric grid model **a** simple planar surfaces and **b** the corresponding volumetric grid; **c** cylindrical surfaces and **d** the corresponding cylindrical volume mesh

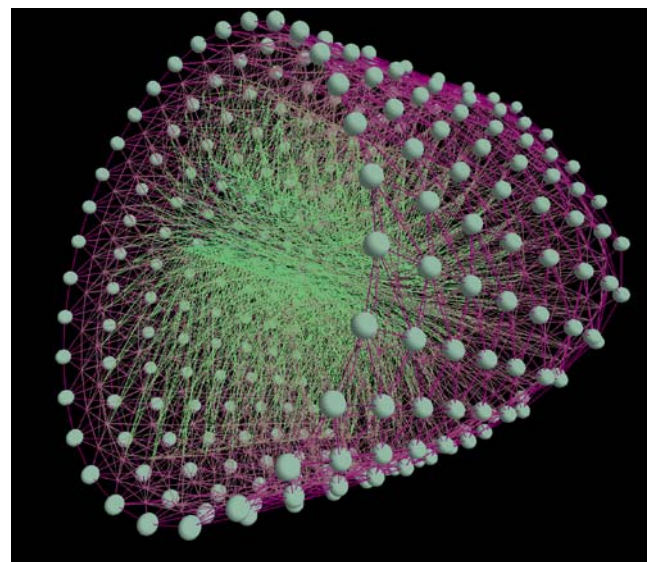


Fig. 4 Examples of the generated topological structure from visible human

tilayered mass-spring model generated from Chinese Visible Human (CVH) upper limb is shown in Fig. 4.

Bilinear spring modeling and optimization

In this section, we discuss the details of the spring modeling and optimization in the PPU-based MSM as well as the on-the-fly simulation process of soft tissue deformation.

According to the biomechanics literatures [16, 17], the elastic properties of many body tissues are not simply linear. Taking the skin and passive properties of skeletal muscle as examples, the stress-strain curve usually has an exponential like shape. In general, it has a very low stiffness at the beginning and an extremely high stiffness after the tissue is stretched to a certain extent. Therefore, a bilinear stress-strain relation can nicely describe the elasticity of body tissues. By approximating to a piecewise linear function, we form the bilinear stress-strain curve. Based on this curve, we can use a nonlinear mass-spring system to simulate the tissues to produce a convincing dynamic effect of body tissue.

The PPU mass-spring system

PPU provides built-in linear mass-spring modeling. Similar to a typical mass-spring system, we can define the mass of an individual node and the elastic properties of an individual spring. As illustrated in Fig. 5, the stress-strain curve of the linear mass-spring model is defined by 5 parameters: compress

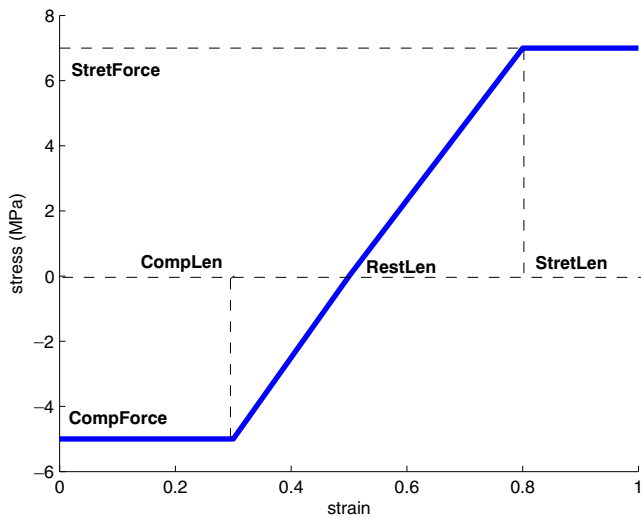


Fig. 5 The stress–strain curve of linear spring defined in PPU

force (*CompForce*), compress length (*CompLen*), rest length (*RestLen*), stretch force (*StretForce*) and stretch length (*StretLen*).

Modeling of bilinear springs

In order to maximize the simulation performance, we extend the PPU-accelerated linear mass-spring system to support bilinear springs. Our bilinear modeling is to place two linear springs with different elastic configurations in parallel. The basic idea of combining two linear springs in producing a bilinear spring is that the first linear spring effective in the first phase of the stress-strain curve, while the second linear spring becomes activated when the turning point is reached. Figure 6 shows how

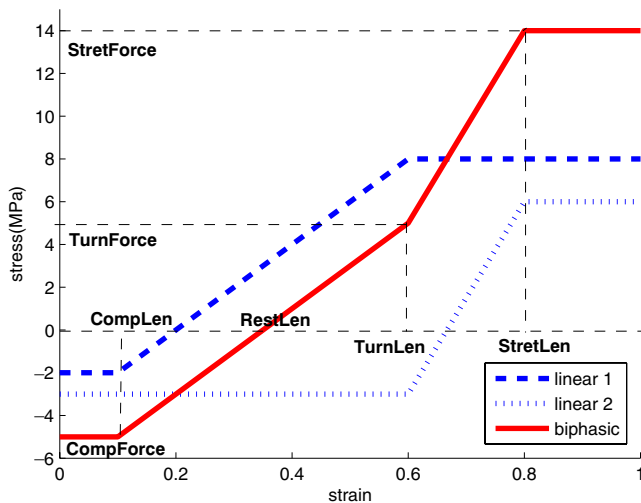


Fig. 6 The stress–strain curve of a bilinear spring composed from 2 linear springs

Algorithm 1 Bilinear decomposition.

```

Spring 1
CompLen1 ← CompLenBP
StretLen1 ← TurnLenBP
CompForce1 ← CompForceBP/2
StretForce1 ← TurnForceBP – CompForce2
RestLen1 ← CompForce1/K1 – CompLen1

Spring 2
CompLen2 ← TurnLenBP
StretLen2 ← StretLenBP
CompForce2 ← CompForceBP – CompForce1
StretForce2 ← StretForceBP – StretForce1
RestLen2 ← CompForce2/K2 – CompLen2
    
```

the two linear stress-strain curves are combined to form the bilinear curve.

Comparing to a linear spring, the bilinear spring has two more parameters which are the turning length and turning force (*TurnLen* and *TurnForce*), corresponding to where the change of stiffness take place. Since the bilinear springs are constructed with linear springs, the decomposition into linear springs can be easily and automatically derived during the deformable model is created. Algorithm 1 lists the formula for the bilinear decomposition, subscript 1 and 2 corresponds to the two different linear springs, while BP represents the composed bilinear spring. Here K_1 and K_2 are the stiffness of the two springs, which constitute one bilinear spring. Note that the compression forces (*CompForce*) of spring 1 and 2 can be set to any arbitrary values between $[0, CompForce_{BP}]$, while the only requirement is that their sum must be equal to $CompForce_{BP}$. In our simulations, we set them as half of $CompForce_{BP}$.

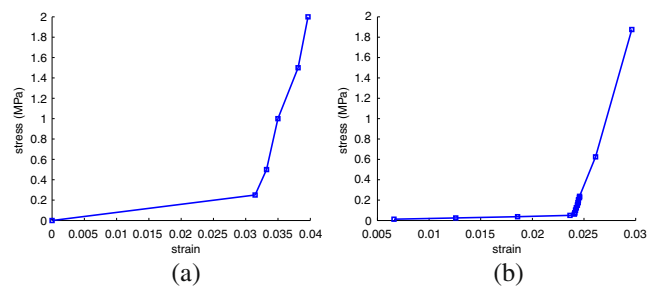


Fig. 7 Comparison of a single bilinear spring and the macro behavior; **a** a bilinear spring with $K_1 = 0.9, K_2 = 170.0$, **b** the corresponding macro behavior with $K_1 = 0.11, K_2 = 19.02$ which is similar to the dermis’ mechanical property

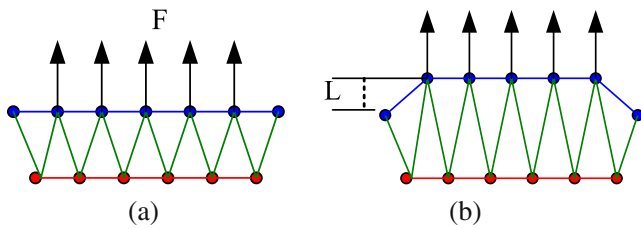


Fig. 8 Experiment on collecting macrostructural stress–strain relation of multilayered model. **a** The model at rest state and an external force exerted on it **b** the stretch length L is recorded at equilibrium

Analysis of macroscopic behavior in multilayer model

As we are modeling the body tissue as a mass-spring system, the biomechanical data of tissue properties is always referring to the behavior of the entire mass-spring system instead of a single spring’s elasticity. Figure 7 shows an experimental result of using bilinear spring in our multilayer model. The microscopic and macroscopic elastic behaviors of our multilayered model are plotted side by side.

From this experiment, we can realize the fact that though the microscopic and macroscopic stress-strain curves are different, microscopically bilinear springs give rise to the same macroscopic bilinear elastic behavior too. Since we cannot directly apply the macroscopic biomechanical data onto individual springs, more in-depth analysis has to be carried out in order to match the macroscopic elastic behavior of the model with the biomechanical data.

An optimal set of elastic properties in individual springs that provides a matched macrostructural elastic behavior to the entire model can be found using many optimization methods, like simulated annealing and downhill simplex method [18]. In our framework, we

choose simulated annealing which is very suitable for an optimization problem with large number of parameters with no need to evaluate the first derivative in the objective function. Details of simulated annealing on continuous domain can be referred to [18].

In order to reduce the dimensionality of our optimization problem and to conform with the mechanical properties of different soft tissue, each layer within the multilayered model is optimized independently. For layers modeled by bilinear springs, e.g. skin and muscle, totally 3 parameters, which are the first phase stiffness (K_1), turning length (*TurnLen*) and the second phase stiffness (K_2), have to be optimized. For layers modeled by linear springs, such as adipose tissues, only the stiffness parameter optimization is necessary.

The objective function of the optimization process is given by Eq. 1:

$$\psi = \sum_i w_i (p_i - q_i)^2 \tag{1}$$

where p_i is the evaluated value of i -th parameter from the experiment while q_i is the corresponding measured value from biomechanical literatures for that particular tissue and w_i is the weighting for the parameter.

In order to evaluate the value of p_i , experiments are carried out on the multilayered model, Fig. 8 illustrates how the experiment is performed. By applying appropriate forces on the mass-spring structure, we record its corresponding strain at its rest state. The recorded data are processed by regression in order to get the corresponding stiffness or other necessary properties. For example, in case of the bilinear spring modeling, the regression is performed repeatedly. Each time we assume a data point as a turning point and based on the point to divide all the points into two groups. Then, a least-square fitting of line is performed separately on

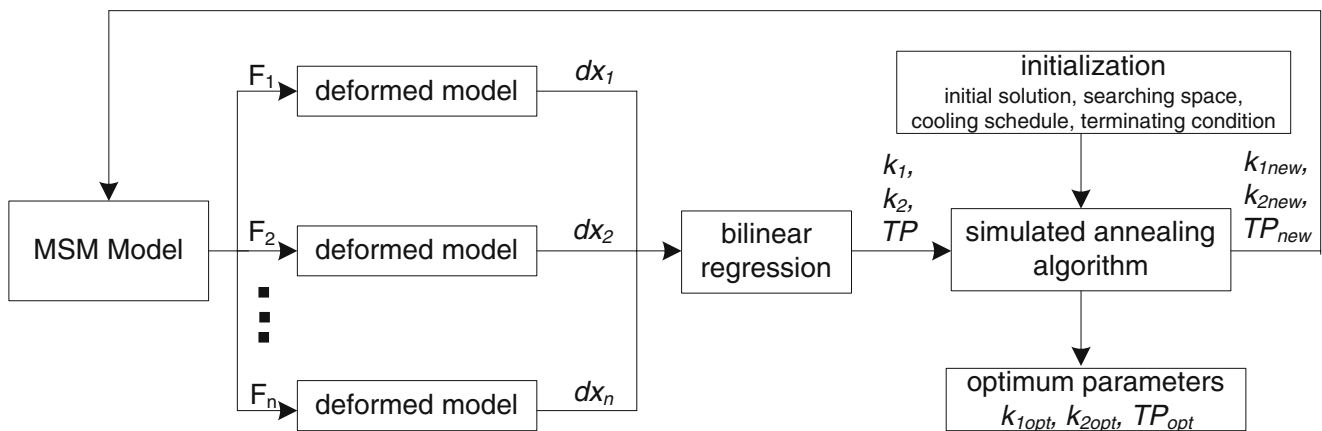


Fig. 9 The optimization process for the elastic properties of individual bilinear springs in our multilayered mass-spring (MMS) model

the two groups of points, and the corresponding fitting error is also computed. We repeat the process for all data point, and the one with minimum error is the best fitting result. Therefore, we can get the turning length and two spring stiffness accordingly.

The optimization process evaluates the objective function repeatedly until a minimum is obtained. Figure 9 shows the flow chart of the optimization process for the elastic properties of bilinear individual springs in our framework. By applying forces with various magnitudes on the mass-spring structure, we record its corresponding strain dx_i at its equilibrium state. So if n difference forces F_i are applied, we obtain n different dx_i . Then, the recorded data are processed by bilinear regression in order to get the macroscopic k_1, k_2 , and TP , which are fed into the simulating annealing algorithm. The difference between these elastic properties and the corresponding measured value from biomechanical literatures is calculated by an objective function (Eq. 1). If the terminating condition is fulfilled, we stop the optimization process and obtain the optimum parameters; otherwise, we calculate a new set of parameters in the search space based on the cooling schedule and fed it back into the MSM model for another iteration. The optimization process is continued till the terminating condition is satisfied.

Visualization

In medical simulation application, visual effects contribute much to the realism of perception. Based on a deformed volumetric grid, we first identify all edge nodes. Surfaces are reconstructed from these edge nodes where the distance of these nodes are obtained from the mass-spring model. From the visualization viewpoint, the mesh resolution used in the mass-spring model representation is usually not dense enough for realistically rendering of the appearance of soft tissues. Therefore, it is necessary to make extra enhancement on the surface mesh for generating higher rendering quality. On top of the regular surface grid, we deploy adaptive Bernstein-Bézier patches to meliorate the visual quality of surfaces. Fourth-order 2-dimensional Bézier patch is exploited to replace every quadrilateral grid cell:

$$S\left(\frac{u}{n}, \frac{v}{m}\right) = \sum_{i=0}^k \sum_{j=0}^l B_i^k\left(\frac{u}{n}\right) B_j^l\left(\frac{v}{m}\right) P_{ij} \tag{2}$$

where $k = l = 4$, u and v are the integer step parameters along the edges of one particular patch. $B(u)$ and $B(v)$ are the Bernstein basis functions along the direction U and V . P_{ij} can be the vertices, normals,

colors or texture coordinates. n and m are the number of subdivisions along U and V respectively. In other words, u and v range from $(0, \dots, n)$ and $(0, \dots, m)$ respectively.

In order to allow adaptive resolution on each Bézier patch, the number of subdivision n and m can be adaptively selected by computing the local curvature determined by the normal vector of the corner vertices:

$$m = \left\lceil \max(\arccos(\mathbf{N}_0 \cdot \mathbf{N}_1), \arccos(\mathbf{N}_2 \cdot \mathbf{N}_3)) \times \frac{1}{\phi} \right\rceil \tag{3}$$

$$n = \left\lceil \max(\arccos(\mathbf{N}_0 \cdot \mathbf{N}_2), \arccos(\mathbf{N}_1 \cdot \mathbf{N}_3)) \times \frac{1}{\phi} \right\rceil \tag{4}$$

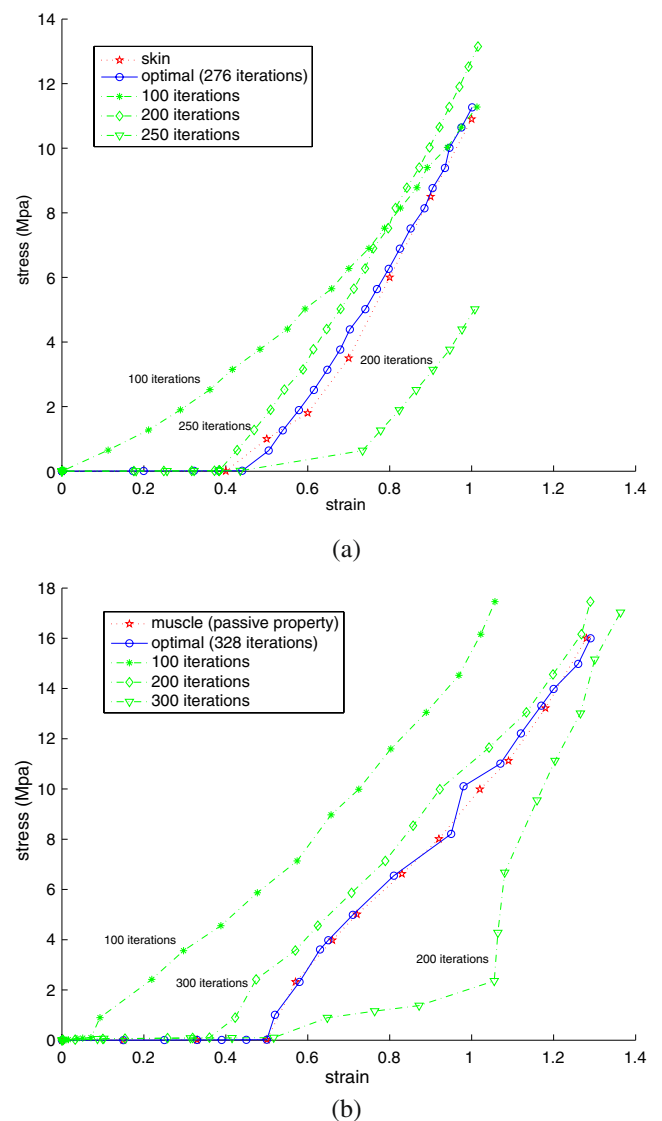


Fig. 10 The macrostructural stress–strain curves in different iterations during the optimization of elastic parameters of skin **a** and muscle **b**. (The optimal result and the real skin stress–strain curve are plotted for references)

Table 1 The optimized properties of springs in our virtual limb with 39×10 grid

	K_1 (KPa)	K_2 (KPa)	TP (in strain)
Skin	0.118	18.51	0.41
Muscle	0.03	2.53	1.56
Fatty tissue	0.0009	N/A	N/A

where N_0, N_1, N_2, N_3 are the normal vectors. ϕ is the threshold angle determining the smoothness of the mesh, while in our experiment, a value of $\frac{\pi}{18}$ generally produces acceptable visual effects.

To visualize the deformed soft tissue realistically, we exploit the hardware-accelerated 3D texture mapping. On one hand, a 3D texture volume is first reconstructed from a set of visible human cryosectional slices. We perform a simple pre-classification process to exclude unwanted voxels, for example, voxels depicting the frozen ice. On the other hand, a volume grid is mapped with the volume texture. Here, the 3D texture coordinates only have to be generated once during the initialization, since the texture coordinates of a node remain unchanged when it is transformed. In this sense, an interactive deformable texture rendering can be achieved.

Results

Bilinear spring optimization

Figure 10a and b show the stress-strain curves obtained in optimization processes of bilinear elastic properties of skin and muscle. The blue curve is the optimal result, while intermediate results in different iterations are plotted together. Table 1 shows the optimized properties of springs in our virtual upper limb with a 39×10 grid on the upper most layer, where the average grid size is about $1.5 \text{ cm} \times 1.6 \text{ cm}$. Note that we adopt the bilinear model to simulate the elastic properties of skin and muscle and linear spring model to simulate the elastic property of adipose tissues.

The optimal microscopic spring properties are dependent on the grid size of the mass-spring structure. For example, in our CVH upper limb with 18×5

Table 2 Individual spring properties of skin layer in different grid sizes

Spring	Grid size	K_1 (KPa)	K_2 (KPa)	TP (cm)
A	1 cm \times 1 cm	0.60991	490.88	0.85
B	1.5 cm \times 1.5 cm	0.71244	1389.37	1.18
C	2.0 cm \times 2.0 cm	0.86497	4957.34	1.54

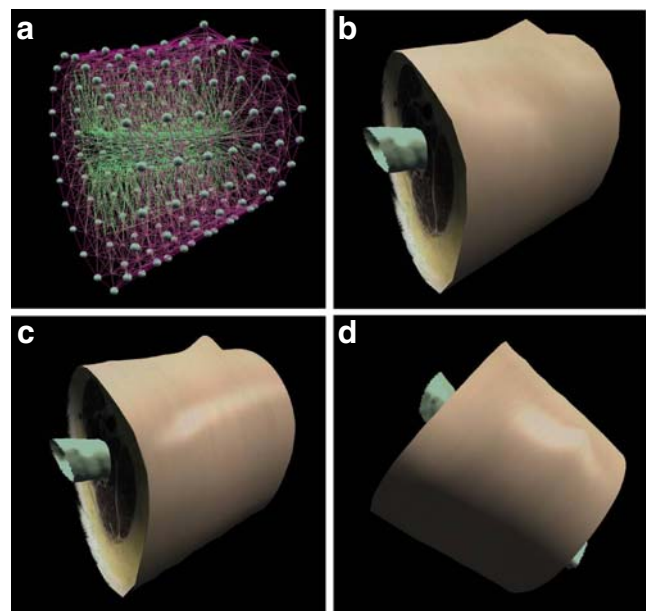


Fig. 11 Visualization result of CVH upper limb model on **a** the mass-spring model, **b** rendering with no Bézier re-patching, **c** & **d** rendering with Bézier re-patching

grids, the average grid size is about $1.8 \text{ cm} \times 1.7 \text{ cm}$. Therefore, we need to optimize the properties of the mass-spring model for every particular grid size. Table 2 shows the optimized skin layer spring properties for different sizes of grid in the mass-spring structure. Based on the model’s size and grid scales, we can find the best matched grid size and use the corresponding optimal spring properties.

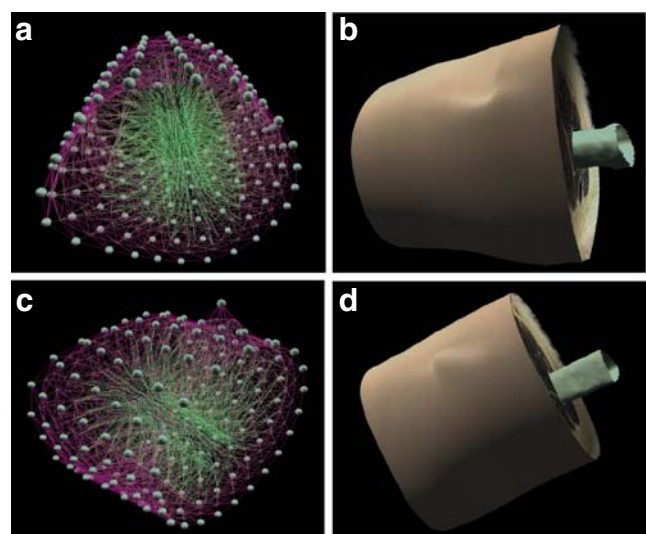


Fig. 12 Visualization result obtained by pushing or pulling the mesh (**a–d**)

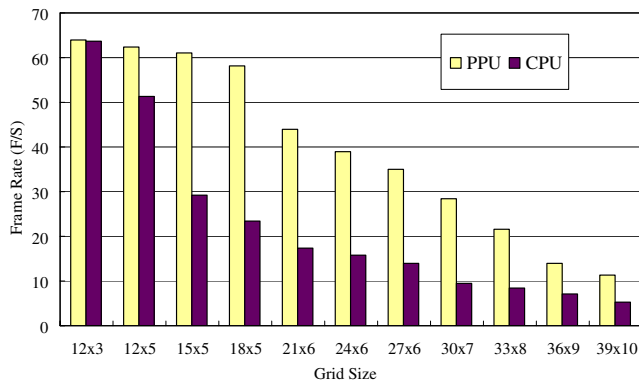


Fig. 13 Comparison on time performance of using PPU and CPU in our framework under different size of deformable model used

Simulation with CVH datasets

A prototype has been implemented based on the proposed framework. The rendering results of the CVH upper limb is shown in Fig. 11. In this experiment, the surface is directly rendered from the original mass-spring grid structure (shown in Fig. 11a) without undergoing any re-patching process. When the volume mesh is in the rest state, the appearance of the model is acceptable. However, once the mesh is deformed, like the situation shown in Fig. 11b, the jerky edges could be quite obvious.

In another set of experiments, we apply the adaptive Bernstein-Bézier re-patching. Figure 11c & d show how the visual quality of the surface can be greatly improved. In this example, the specular highlight is used to show up the improvement.

Several texture-mapped results and their corresponding deformed mesh structures are depicted in Fig. 12.

Timing performance

A series of experiments using deformable models with different grid-sizes have been done to compare the

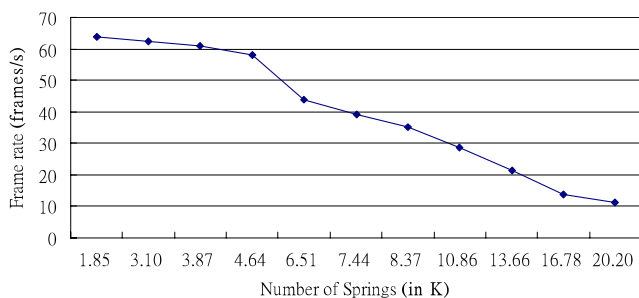


Fig. 14 Comparison on time performance trend

Table 3 Quantitative comparison between different size of deformable models

Grid size	Mass number	Spring number	Frame rate
12×3	264	1848	63.9
12×5	432	3096	62.5
15×5	540	3870	61
18×5	648	4644	58.1
21×6	903	6510	43.9
24×6	1032	7440	39
27×6	1161	8370	35.1
30×7	1500	10860	28.5
33×8	1881	13662	21.5
36×9	2304	16776	13.9
39×10	2769	20202	11.2

timing performance of our PPU based framework and that of a CPU based framework. Figure 13 shows the results of these experiments done on a PC equipped with Pentium 4 Dual Core 3.2 GHz CPU and 4 GB memory. It is observed that there is a significant speed improvement in all grid-sizes with the use of PPU, especially the 15×5 and 18×5 grids with an doubled improvement.

Figure 14 and Table 3 clearly reveal the speed performance trend with increase in complexity of the model in terms of total number of springs used. From the results, we observe the performance drop is small at the beginning, while the decrease in frame rate becomes stable after number of springs excess 4.5k. This may be due to the bottleneck at the old standard PCI bus.

In practice, for a complex model like 39×10 grids, the frame rate is still above 10 F/S, which can fulfill the requirement of most real-time applications. In contrast, when the number of springs is more than 10000 (i.e. 30×7 grid), the frame rates of CPU base framework are already under 10 F/S. This demonstrates the computational speed can be greatly accelerated by developing deformable surgical simulators based on PPU, especially when the deformation models are complicated.

Discussion and conclusion

With the idea of exploiting the computational power of the newly released PPU for medical simulation, we have successfully demonstrated a workable framework for simulating the soft tissues with the concern of both realism and speed. Choosing PPU as an alternative for physics acceleration instead of GPU, which is popularly used for parallelization and acceleration, is because of its easy to use and advantage in rapid prototyping. The well designed developing environment coming with

the PPU can significantly reduce the developing time. Moreover, PPU relieves the workload of GPU, as all physics computations run in PPU, while GPU works only for the rendering process. In order to provide a bilinear elastic behavior which appears in most of the human tissues, we had proposed a simple but efficient tactic on modifying the built-in PPU linear mass-spring model. Besides, in order to ensure the quality of simulation, a robust and systematic procedure is proposed to model the geometrical structure and biomechanical elastic properties of soft tissues. By utilizing CVH datasets, a realistic visual and deformation effect can be achieved. Experiments reveal the feasibility of the proposed framework in providing interactive and realistic deformation of human organ for orthopedic surgery simulation, even when relatively complex models are involved.

Acknowledgements This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK 4461/05M). This work is affiliated with the Virtual Reality, Visualization and Imaging Research Center at The Chinese University of Hong Kong as well as the CUHK MoE-Microsoft Key Laboratory of Human-Centric Computing and Interface Technologies.

References

- Nealen, A., Muller, M., Keiser, R., Boxerman, E., & Carlson, M. (2005). Physically based deformable models in computer graphics. In *Eurographics state of the art report*.
- Olga, S., Alexei, S., & Sen, H. (2007). Orthopaedic surgery training simulation. *Journal of Mechanics in Medicine and Biology*, 7(1), 37–53. World Scientific.
- Bruce, D., Nordquist, P., Skarman, E., Mark, T., Gina, B., & David, B. (2006). Integrated lower extremity trauma simulator. In *Proceedings of medicine meets virtual reality 2006* (pp. 19–24). IOS Press.
- Heng, P., Cheng, C., Wong, T., Wu, W., Xu, Y., Xie, Y., et al. (2006). Virtual reality techniques: Application to anatomic visualization and orthopaedics training. *Clinical Orthopaedics and Related Research*, 442, 5–12.
- McCarthy, A., Moody, L., Waterworth, A., & Bickerstaff, D. (2006). Passive haptics in a knee arthroscopy simulator: Is it valid for core skills training? *Clinical Orthopaedics and Related Research*, 442, 13–20.
- Reinig, K., Lee, C., Rubinstein, D., Bagur, M., & Spitzer, V. (2006). The united states military's thigh trauma simulator. *Clinical Orthopaedics and Related Research*, 442, 45–56.
- Joachim, G., & Rüdiger, W. (2005). Mass-spring systems on the gpu. *Simulation Modelling Practice and Theory*, 13, 693–702.
- Mosegaard, J., & Sorensen, T. (2005). Gpu accelerated surgical simulators for complex morphology. In *Proceedings of the 2005 IEEE conference on virtual reality* (pp. 147–153). Washington, DC: IEEE Computer Society.
- Rasmusson, A., Mosegaard, J., & Sørensen, T. S. (2008). Exploring parallel algorithms for volumetric mass-spring-damper models in cuda. In *International symposium on computational models for biomedical simulation* (pp. 49–58).
- Spitzer, V., Ackerman, M., Scherzinger, A., & Whitlock, D. (1996). The visible human male: A technical report. *Journal of the American Medical Informatics Association*, 3(2), 118–130.
- Park, J., Chung, M., Hwang, S., Lee, Y., Har, D., & Park, H. (2005). Visible Korean human: Improved serially sectioned images of the entire body. *IEEE Transactions on Medical Imaging*, 24(3), 352–260.
- Zhang, S., Heng, P., Liu, Z., Tan, L., Qiu, M., et al. (2003). Creation of the chinese visible human data set. *Anatomical Record (Part B: New Anatomy)*, 275, 190–195.
- Heng, P., Zhang, S., Xie, Y., Wong, T., Chui, Y., & Cheng, C. (2006). Photorealistic virtual anatomy based on chinese visible human data. *Clinical Anatomy*, 19(3), 232–239.
- Székely, G., Brechbuehler, C., Hutter, R., Rhomberg, A., & Schmid, P. (1998). Modeling of soft tissue deformation for laparoscopic surgery simulation. In *MICCAI '98: Proceedings of the first international conference on medical image computing and computer-assisted intervention* (pp. 550–561).
- Wang, X., & Devarajan, V. (2004). A honeycomb model for soft tissue deformation. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on virtual reality continuum and its applications in industry* (pp. 257–260).
- Fung, Y. (1993). *Biomechanics: Mechanical properties of living tissues*, (2nd ed.). Berlin: Springer.
- Maurel, W., Wu, Y., Thalmann, N., & Thalmann, D. (1998). *Biomechanical models for soft tissue simulation*. Berlin: Springer.
- Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (2002). *Numerical recipes in c++. The art of scientific computing*. Cambridge: Cambridge University Press.