35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Synchronous Simulation for Deformation of Liver and Gallbladder with Stretch and Compression Compensation

Yuping Duan[1]; Weimin Huang[1]; Huibin Chang[2]; Kyaw Kyar Toe[1]; Tao Yang[1]; Jiayin Zhou[1]; Jiang Liu[1]; Soo Kng Teo[3]; Chi Wan Lim[3]; Yi Su[3]; Chee Kong Chui[4] and Stephen Chang[5]

***Abstract*— One challenge in surgical simulation is to design stable deformable models to simulate the dynamics of organs synchronously. In this paper, we develop a novel mass-spring model on the tetrahedral meshes for soft organs such as the liver and gallbladder, which can stably deform with large time steps. We model the contact forces between the organs as a kind of forces generated by the tensions of repulsive springs connecting in between the organs. The simulation system couples a pair of constraints on the length of springs with an implicit integration method. Based on the novel constraints, our simulator can efficiently preserve the volumes and geometric properties of the liver and gallbladder during the simulation. The numerical examples demonstrate that the proposed simulation system can provide realistic and stable deformable results.**

I. INTRODUCTION

As one of the important training components, elastic deformation models have been greatly studied in the last two decades [1], [2], [3] for surgical simulation. A good overview of deformable models has been reported in [3], e.g., mass-spring models, finite element methods and finite difference approaches. Among all these models, mass-spring models [4], [5], [6], [7] are one of the most popular techniques to simulate the animation of soft bodies due to their simple structures.

A mass-spring model consists of a set of point masses that are connected by ideal weightless elastic springs. The points masses are either derived from the edges of a two-dimensional polygonal mesh for representing surface of the object, or they might originate from a three-dimensional mesh mode representing both the surface and internal structure of the object. The other spring component obeys Hooke's law, which states that the force with which the spring pushes back is linearly proportional to the distance from its equilibrium length. With the external forces (due to contact, gravity, etc.) and internal forces (due to the deformation of springs), mass-spring systems are usually formulated as time-varying partial differential equations based on Newton's Second Law which, after discretization, become numerically solvable as ordinary differential equations (ODEs).

In the literature, both explicit schemes [6], [7] and implicit/semi-implicit schemes [4], [8], [9] were explored to solve ODEs emerging from mass-spring models. Explicit schemes compute the next status of the system from the internal forces estimated at the current iteration while implicit schemes obtain the next status from the internal forces at the next iteration. It is well-known that the explicit scheme is numerically unstable unless the time step is made small enough for interactive applications. Compared to explicit schemes, implicit schemes are usually unconditionally stable but require to solve a large sparse system in each time step.

In this paper, we describe a simulation system for multiple bodies deformation that uses tetrahedral meshes and relies on the implicit numerical integration scheme. By creating the springs in between the contact surfaces of the liver and gallbladder, we convert the contact forces into the internal forces from the tensions of springs. For mass-spring models, when a concentration of large forces occurs in a small region of the soft body, the simulation result falls into the problem of local deformation (the so-called "super-elastic" effect), which makes it seems unrealistic [10]. To overcome this problem, we derive a couple of constraints on the length of springs as an additional post processing step for the implicit integration. These constraints can prevent the soft bodies from over-stretching and over-compressing, which preserve the volumes and geometry properties of the soft bodies during the simulation.

II. MASS-SPRING MODEL

Our mass-spring model for the liver and gallbladder is designed based on volumetric meshes. A data structure that describes the set of vertices, edges and tetrahedrons of the mesh is illustrated in Fig. 1, both the tetrahedral mesh of the liver and a tetrahedron unit are given. More specifically,

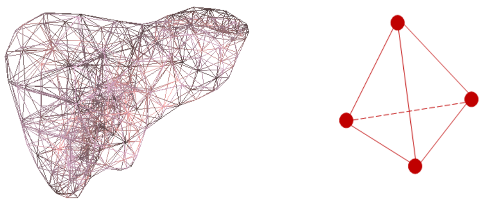


Fig. 1. Representation of volumetric meshes for the liver.

[1] Y. Duan, W. Huang, K.K. Toe, T. Yang, J. Zhou and J. Liu are with Institute for Infocomm Research, A*STAR, Singapore, {duany, wmhuang, kktoe, tyang, jzhou, jliu}@i2r.a-star.edu.sg

[2] H. Chang is with Tianjin Normal University, China, changhuibin@gmail.com

[3] S.K. Teo, C.W. Lim and Y. Su are with Institute of High Performance Computing, A*STAR, Singapore, {teosk, limcw, suyi}@ihpc.a-star.edu.sg

[4] C.K. Chui is with Department of Mechanical Engineering, National University of Singapore, Singapore, mpecck@nus.edu.sg

[5] S. Chang is with National University Hospital, Singapore, surv7@nus.edu.sg

978-1-4577-0216-7/13/$26.00 ©2013 IEEE

① For a vertex, we store its current and rest positions;
② For an edge, we store the two adjacent vertices;
③ For a tetrahedron, we store the four vertices formed the tetrahedron.

Suppose our mass-spring model consists of n particles. Each mass particle is linked to its neighbors by massless springs of the rest length greater than zero. For the ith particle, $i = 1, \ldots, n$, it has the mass $m_i \in \mathbb{R}$ and position $\boldsymbol{x}_i \in \mathbb{R}^3$. The geometric state of all the particles is simply $\boldsymbol{m} \in \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^{3n}$. The same component notation applies to the force $\boldsymbol{f} \in \mathbb{R}^{3n}$ acting on the tetrahedral mesh and $\boldsymbol{f}_i \in \mathbb{R}^3$ on the ith particle. The mass-spring system can be described by the Newton's Second Law as follows

$$\boldsymbol{m}\ddot{\boldsymbol{x}} = \boldsymbol{f}, \tag{1}$$

where $\ddot{\boldsymbol{x}}$ is the second derivative of the position with respect to time.

III. Forces

The force in (1) denotes the total force in the system, which includes internal spring forces, damping forces, contact forces and user specified external forces.

A. Spring Forces

Springs are modelled with linear elasticity. The force acting on mass i generated by the spring connecting i and j is in direct proportion with the extension of the spring. Therefore, according to Hooke's Law, the spring force is defined as follows

$$\boldsymbol{f}_i^s = k_s(\|\boldsymbol{x}_j - \boldsymbol{x}_i\| - l_0) \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}, \tag{2}$$

where k_s is the spring's stiffness and l_0 is the rest length, i.e., the distance between the two particles when the spring exerts no force.

B. Damping Forces

Due to imperfect elasticity of physical bodies, the energy dissipation occurs during the deformation. We use the spring damping to represent the viscous force. These damping forces are proportional to the difference of velocity $\boldsymbol{v} \in \mathbb{R}^{3n}$ projected onto the spring and are momentum conserving as well, which can be defined as

$$\boldsymbol{f}_i^d = k_d \frac{(\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}, \tag{3}$$

where k_d is the spring's damping constant.

C. Contact Forces

The gallbladder is connected to the lower surface of the liver at the gallbladder fossa by connecting tissues. Due to the anatomical position of the gallbladder to the liver, when one applies external pull forces on the liver/gallbladder, there exist contact forces in the contact surface of the liver and gallbladder. Such contact forces emerge in pair on both the liver side and gallbladder side. Therefore, we can regard them as the internal forces in the mass-spring system. To define these contact forces, we artificially create a set of repulsive springs in between the contact surfaces of the liver and gallbladder to repulse them in collision. Suppose the mesh models representing the liver and gallbladder are composed of two kinds of springs with different properties. The springs connecting the liver and gallbladder is the third kind of springs in the system. Thus, the contact forces between the liver and gallbladder are also expressed as the spring forces and damping forces on the connecting vertices. We can easily control the performance of the contact forces by setting proper stiffness and damping constant to the connecting springs between the liver and gallbladder.

IV. Implicit Integration

In the numerical simulation, Equation (1) is written as two coupled first order equations by introducing the velocity $\boldsymbol{v} \in \mathbb{R}^3$ as follows

$$\begin{cases} \dot{\boldsymbol{v}} = \boldsymbol{f}(t)/\boldsymbol{m}, \\ \dot{\boldsymbol{x}} = \boldsymbol{v}. \end{cases} \tag{4}$$

The most popular implicit integration scheme used in computer graphics is the implicit Euler scheme [4]. We construct a mass matrix $\boldsymbol{M} \in \mathbb{R}^{3n \times 3n}$ which is a diagonal matrix with the main diagonal vector [m_1, m_1, m_1, m_2, m_2, m_2, ..., m_n, m_n, m_n]. The implicit scheme for (4) can be described as follows

$$\begin{cases} \boldsymbol{v}(t+\tau) = \boldsymbol{v}(t) + \tau \boldsymbol{M}^{-1}\boldsymbol{f}(t+\tau), \\ \boldsymbol{x}(t+\tau) = \boldsymbol{x}(t) + \tau \boldsymbol{v}(t+\tau), \end{cases} \tag{5}$$

where τ is the time step size. Hereafter we denote $\boldsymbol{f}(t) := \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{v}(t), t)$. The general way to solve such a system is to use the Newton-Raphson method. This method starts at a guess for the unknown $\boldsymbol{v}(t)$ and iteratively improves this guess. To this end, the forces are linearized at current state by Taylor expansion as follows

$$\begin{aligned} \boldsymbol{f}(t+\tau) = \boldsymbol{f}(t) &+ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Big|_t \cdot (\boldsymbol{x}(t+\tau) - \boldsymbol{x}(t)) \\ &+ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\Big|_t \cdot (\boldsymbol{v}(t+\tau) - \boldsymbol{v}(t)) + \frac{\partial \boldsymbol{f}}{\partial t}\Big|_t. \end{aligned} \tag{6}$$

By substituting the force (6) into the velocity update of (5), we have

$$\begin{aligned} \boldsymbol{v}(t+\tau) = \boldsymbol{v}(t) + \tau \boldsymbol{M}^{-1}\Big(&\boldsymbol{f}(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}(t+\tau) - \boldsymbol{x}(t)) \\ &+ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}(\boldsymbol{v}(t+\tau) - \boldsymbol{v}(t))\Big). \end{aligned}$$

We can further simplify the above equation using $\boldsymbol{x}(t + \tau) \approx \boldsymbol{x}(t) + \tau \boldsymbol{v}(t+\tau)$ and obtain

$$\begin{aligned} \boldsymbol{v}(t+\tau) = \boldsymbol{v}(t) + \tau \boldsymbol{M}^{-1}\Big(&\boldsymbol{f}(t) + \tau \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \cdot \boldsymbol{v}(t+\tau) \\ &+ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}(\boldsymbol{v}(t+\tau) - \boldsymbol{v}(t))\Big). \end{aligned} \tag{7}$$

Let $\boldsymbol{K} := \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Big|_t, \boldsymbol{D} := \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\Big|_t$. We can rearrange (7) to the following linear system

$$\Big(\boldsymbol{M} - \tau \boldsymbol{D} - \tau^2 \boldsymbol{K}\Big)\boldsymbol{v}(t+\tau) = \Big(\boldsymbol{M} - \tau \boldsymbol{D}\Big) \cdot \boldsymbol{v}(t) + \tau \boldsymbol{f}(t). \tag{8}$$

The implicit integration method generates a large sparse linear system (8) at each time step, which is solved by Gauss-Seidel iteration in our simulation.

Remark: Since both the spring force (2) and damping force (3) are functions of position, $\boldsymbol{K}$ is composed of two parts (i.e., $\boldsymbol{K} := \boldsymbol{K}^1 + \boldsymbol{K}^2$): $\boldsymbol{K}^1$ is produced by the spring force and $\boldsymbol{K}^2$ is derived from the damping force.

V. Volumetric Position-based Constraints

We obtain both the position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$ in each iteration of the implicit integration. We further introduce constraints to the position as the post-processing process to regularize the simulation results.

Given a constraint function $C(\boldsymbol{x})$, we find the correction $\Delta\boldsymbol{x}$ on position such that $C(\boldsymbol{x} + \Delta\boldsymbol{x}) = 0$ as for cloth simulation in [11]. The constraint function is linearized in the neighborhood of the current solution using

$$C(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx C(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} C(\boldsymbol{x}) \cdot \Delta\boldsymbol{x} = 0. \qquad (9)$$

Let $\Delta\boldsymbol{x}$ be in the direction of $\nabla_{\boldsymbol{x}} C$, that is $\Delta\boldsymbol{x} = \lambda \nabla_{\boldsymbol{x}} C(\boldsymbol{x})$. Therefore, we can solve $\Delta\boldsymbol{x}$ from (9) as follows

$$\Delta\boldsymbol{x} = -\frac{C(\boldsymbol{x})}{|\nabla_{\boldsymbol{x}} C(\boldsymbol{x})|^2} \nabla_{\boldsymbol{x}} C(\boldsymbol{x}). \qquad (10)$$

Consequently, we have the following correction formula for an individual vertex $\boldsymbol{x}_i$, $i = 1, \ldots, n$,

$$\Delta\boldsymbol{x}_i = -\frac{C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)}{\sum\limits_i |\nabla_{\boldsymbol{x}_i} C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)|^2} \nabla_{\boldsymbol{x}_i} C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n).$$

Due to the three-dimensional structure of the tetrahedral mesh, such correction have to be a global action to guarantee the stability of the system under the constraints. Therefore, for the vertex $\boldsymbol{x}_i$, $i = 1, \ldots, n$, we sum up all the terms $\Delta\boldsymbol{x}_i$ contributed by the edges containing the vertex $\boldsymbol{x}_i$.

Since both the liver and gallbladder are far from having ideal elasticity, we design a pair of constraints on the spring length: over-stretching and over-compressing correction.

A. Over-stretching Correction

We set a critical stretching rate γ_s to the springs to protect the spring from being stretched too much. More specifically, when the length of the spring exceeds $(1 + \gamma_s) * l_0$, the constraint is applied to correct the spring back to $(1+\gamma_s)*l_0$. Therefore, we define the over-stretching correction to be an inequality constraint as follows

$$C_{stretch}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + \gamma_s) * l_0 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq 0, \qquad (11)$$

Assume the above inequality constraint, the corrections on the vertex $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from (11) are computed as follows

$$\begin{aligned} \Delta\boldsymbol{x}_i = \Delta^+ = & \quad C_{stretch}(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \\ \Delta\boldsymbol{x}_j = \Delta^- = & -C_{stretch}(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}. \end{aligned}$$

After the implementation of the over-stretching constraint, the position $\boldsymbol{x}_i$, $i = 1, \ldots, n$ is updated as follows

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \sum_{i \in \mathcal{N}_i^+} \Delta^+ + \sum_{i \in \mathcal{N}_i^-} \Delta^-, \qquad (12)$$

where $\mathcal{N}_i^+$ denotes the set of edges that start from x_i and $\mathcal{N}_i^-$ denotes the set of edges that end at x_i.

B. Over-compressing Correction

Correspondingly, as long as the length of the spring is less than $(1 - \gamma_c) * l_0$, where γ_c is the critical compressing rate, we use another constraint to move the spring back to $(1 - \gamma_c) * l_0$. The constraint is defined as

$$C_{compress}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| - (1 - \gamma_c) * l_0 \geq 0. \quad (13)$$

Similarly, the corrections on the vertex $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the constraint (13) are calculated by follows

$$\begin{aligned} \Delta\boldsymbol{x}_i = \Delta^+ = & \quad C_{compress}(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \\ \Delta\boldsymbol{x}_j = \Delta^- = & -C_{compress}(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}. \end{aligned}$$

In the end, we update the position of each vertex $\boldsymbol{x}_k$, $k = 1, \ldots, n$ after applying the over-compressing constraint based on Equation (12).

VI. NUMERICAL EXAMPLES

The anatomical model of the liver and gallbladder used in our system are described in TABLE I. We follow the liver environment defined in [12] to set up the boundary conditions for the liver. Since the posterior border of the liver is connected to the diaphragm by the coronary ligament, it is reasonable to regard that region as the boundary condition for the liver in the numerical implementation.

TABLE I

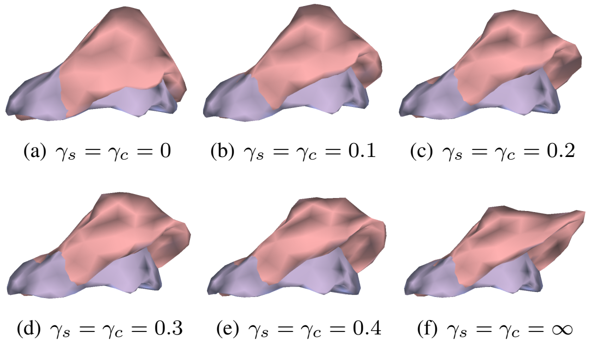
Overview of mesh structure.

Mash	Vertices	Edges	Tetrahedrons
Liver	302	1552	1001
Gallbladder	238	1178	1741

In our simulation, we use five iterations of Gauss-Seidel method to approximate the solution of the sparse linear system (8) at each time step.

A. Liver Deformation

Fig. 2 shows the deformation of the liver under the external forces. In this experiment, we choose different critical deformation rate for both the over-stretching and over-compressing constraints on the length of springs to track the performance of the proposed method. In Fig. 2, both the original liver mesh (blue one) and the deformed liver mesh (pink one) are displayed for comparison. From the experiment, we can conclude that the simulation system converges to different results with different γ_s and γ_c. Therefore, we can efficiently overcome the “super-elastic” deformation arising in the mass-spring model by choosing appropriate deformation rates. In surgery simulations, we can set the deformation rates according to the mechanical properties of soft tissues.



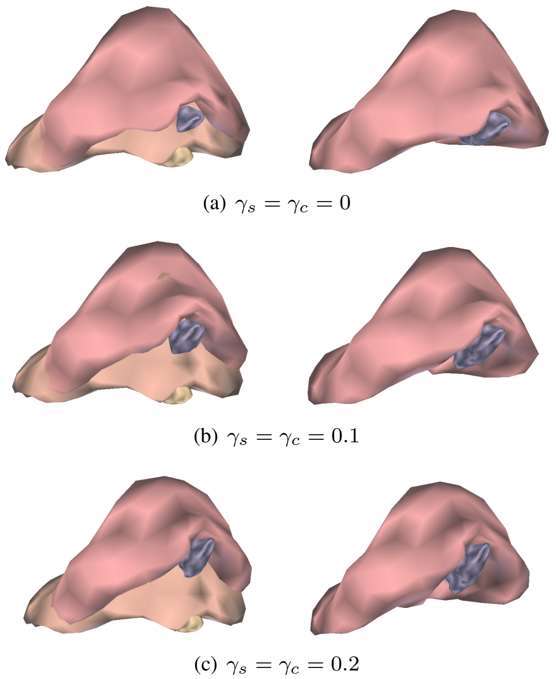
(a) $\gamma_s = \gamma_c = 0$ (b) $\gamma_s = \gamma_c = 0.1$ (c) $\gamma_s = \gamma_c = 0.2$

(d) $\gamma_s = \gamma_c = 0.3$ (e) $\gamma_s = \gamma_c = 0.4$ (f) $\gamma_s = \gamma_c = \infty$

Fig. 2. Liver deformation with the length constraints.

B. Liver and Gallbladder Deformation

For the simulation of the liver and gallbladder system, we model the gallbladder be full of bile. The external forces are applied on the gallbladder and the liver deforms following the contact forces from the gallbladder side. Fig. 3 shows the simulation results of the liver and gallbladder when the system converges. The deformation of the liver illustrates that the repulsive springs can well model the contact forces in the contact surfaces. With the addition of the position-based constraints, our simulator can efficiently avoid the collision between the liver and gallbladder during the simulation. The results demonstrate that the implicit integration method performs stable for multiple objects system.



(a) $\gamma_s = \gamma_c = 0$

(b) $\gamma_s = \gamma_c = 0.1$

(c) $\gamma_s = \gamma_c = 0.2$

Fig. 3. Liver and gallbladder deformation with the length constraints.

C. Performance Analysis

In the last place, we display a summary in TABLE II on the performance of the associated animations shown in Fig. 2 and 3. The largest deformation during the simulation shows that the constraints work very efficient in controlling the deformation. Furthermore, the proposed method can improve the total iteration required for the simulation by applying the position-based constraints (except for the case $\gamma_s = \gamma_c = 0$, which is a very hard constraint). Therefore, the proposed mass-spring model is reliable for practical implementations.

TABLE II

SYSTEM PERFORMANCE FOR SIMULATIONS IN FIGURE 2 AND 3.

γ_s	γ_c	time/iteration(s)	total iteration	largest deform
0	0	0.028	816	38.33
0.1	0.1	0.028	459	60.88
0.2	0.2	0.028	498	78.70
0.3	0.3	0.028	527	86.97
0.4	0.4	0.028	531	94.26
∞	∞	0.025	567	118.234
0	0	0.08	1874	40.94
0.1	0.1	0.08	1351	65.49
0.2	0.2	0.08	1536	79.18

VII. CONCLUSIONS

We have presented an implicit formulation with novel constraints for simulating the deformation of the liver and gallbladder in laparoscopic cholecystectomy. In our approach, we modeled the contact forces between the liver and gallbladder by connecting the liver and gallbladder with repulsive springs. We also incorporated the constraints on the length of springs to preserve geometry properties of the liver and gallbladder during the deformation. The performance of the proposed model is well demonstrated by the numerical examples.

REFERENCES

[1] S. Cotin, H. Delingette, and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.

[2] U. Meier, O. Lopez, C. Monserrat, M. Juan, and M. Alcaniz, “Real-time deformable models for surgery simulation: a survey,” *Computer Methods and Programs in Biomedicine*, vol. 77, no. 3, pp. 183–197, 2005.

[3] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, “Physically based deformable models in computer graphics,” vol. 25, no. 4, pp. 809–836, 2006.

[4] D. Baraff and A. Witkin, “Large steps in cloth simulation,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 43–54.

[5] G. San-Vicente, I. Aguinaga, and J. Celigueta, “Cubical mass-spring model design based on a tensile deformation test and nonlinear material model,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 2, pp. 228–241, 2012.

[6] D. Terzopoulos and K. Fleischer, “Modeling inelastic deformation: viscolelasticity, plasticity, fracture,” in *ACM Siggraph Computer Graphics*, vol. 22, no. 4, 1988, pp. 269–278.

[7] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, “Elastically deformable models,” in *ACM Siggraph Computer Graphics*, vol. 21, no. 4, 1987, pp. 205–214.

[8] K. Choi and H. Ko, “Stable but responsive cloth,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 604–611, 2002.

[9] M. Desbrun, P. Schröder, and A. Barr, “Interactive animation of structured deformable objects,” in *Graphics Interface*, 1999, pp. 1–8.

[10] X. Provot, “Deformation constraints in a mass-spring model to describe rigid cloth behaviour,” in *Graphics Interface*, 1995, pp. 17–19.

[11] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, “Position based dynamics,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109–118, 2007.

[12] M. Vidrascu, H. Delingette, and N. Ayache, “Finite element modeling for surgery simulation,” in *Proceedings of the IEEE*, vol. 86, 1998, pp. 490–503.