Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time

Lang Xu\textsuperscript{1,2}, Yuhua Lu\textsuperscript{1,2} and Qian Liu\textsuperscript{1,2}

\textsuperscript{1}Britton Chance Center for Biomedical Photonics, School of Engineering Sciences, Wuhan National Laboratory for Optoelectronics-Huazhong University of Science and Technology, Hubei, Wuhan 430074, People’s Republic of China
\textsuperscript{2}Key Laboratory for Biomedical Photonics, Huazhong University of Science and Technology, Ministry of Education, 1037 Luoyu Road, Hubei, Wuhan 430074, People’s Republic of China

We propose a novel method to simulate soft tissue deformation for virtual surgery applications. The method considers the mechanical properties of soft tissue, such as its viscoelasticity, nonlinearity and incompressibility; its speed, stability and accuracy also meet the requirements for a surgery simulator. Modifying the traditional equation for mass spring dampers (MSD) introduces nonlinearity and viscoelasticity into the calculation of elastic force. Then, the elastic force is used in the constraint projection step for naturally reducing constraint potential. The node position is enforced by the combined spring force and constraint conservative force through Newton’s second law. We conduct a comparison study of conventional MSD and position-based dynamics for our new integrating method. Our approach enables stable, fast and large step simulation by freely controlling visual effects based on nonlinearity, viscoelasticity and incompressibility. We implement a laparoscopic cholecystectomy simulator to demonstrate the practicality of our method, in which liver and gallbladder deformation can be simulated in real time. Our method is an appropriate choice for the development of real-time virtual surgery applications.

1. Introduction

Surgery simulators can help novices become familiar with real surgical procedures without human loss. They enable scientific and repeatable training through visual rendering via software
and interactive feedback via a hardware-based manipulator. Compared with actual surgical training, simulators reduce the cost of learning and guarantee the training’s effectiveness [1]. Simulating the deformation of soft tissue is necessary for rendering three key aspects of a surgery simulator: the surgical environment, organs’ interactions with the manipulator and force feedback [2]. A well-balanced deformation method can more closely simulate clinical manifestations. Further, accurate simulation of deformation, tearing, bleeding and force feedback make training more efficient. Modelling soft tissue as a viscoelastic body represents more complex nonlinear behaviours than a simple elastomer and viscous body, including geometric features such as creep, stress relaxation and incompressibility [3]. Therefore, it is crucial to develop a stable and fast method with which to simulate the complex deformation behaviours of biological soft tissue in real time.

The development of a real-time deformation algorithm has been accompanied by improvements in computing hardware. In early work, poor computational performance required the use of geometric techniques to simulate deformation, including spline and patch, freeform deformation and skin-based animation. In these techniques, physical accuracy is sacrificed for computational efficiency, and the system has no knowledge about the material being deformed [4]. With advances in computing power, two types of physics-based approaches have flourished. One solves partial differential equations based on a continuum model. This approach is typified by a finite-element method based on meshing object space, and uses the same strategy as the finite-volume and finite-differential methods [5]. The second approach, a meshless method based on shape functions and support domains has been used increasingly widely, such as with smoothed particle hydrodynamics and radial basis function methods [6]. The advantage of continuum methods is that they are sufficiently accurate to reflect an object’s physical characteristics. They are widely used in structural analysis and electromagnetic radiation calculation [7]. Owing to deformation introducing the mesh distortion which reduced accuracy, finite-element method is naturally difficult to deal with large deformation. But the meshless method does not require connection information between simulation node, it works well under large deformation. Also, the material point method uses particles called material point to keep the physical properties and calculate the deformation gradient under background grid, it has been widely used in large deformation, multiphase simulation and fluid dynamics [8]. However, these methods are prohibitively time consuming, and are usually employed in offline applications. Our simulation aims to reflect the significant characteristics of soft tissue (such as viscoelasticity and incompressibility) in surgical applications; for real-time applications, it is appropriate to choose a physics-like method, using mass spring dampers (MSD) [9] and position-based dynamics (PBD) [10]. By abstracting typical features into this model, we achieve high computational efficiency.

Unlike the finite-element method, which uses the discrete solution domain of a partial difference equation, the mass spring method models discrete objects directly. An object consists of node masses with no size and elastic springs with no mass. Deformation is generated by applying elastic force to a node. The state of the system can then be determined by solving algebraic equations derived from Hooke’s Law and Newton’s second law. The mass spring method is simple to calculate and has a clearly structured topology; however, because nodes can only affect adjacent nodes, force propagation is limited and somewhat delayed, which causes local superelasticity [11]. A volume mesh topology and nonlinear spring can mitigate this effect [12]. When updating the node position, numerical integration is required for iteration. An explicit Euler step would cause the result to fail to converge, owing to second-order error. Using implicit Verlet or Runge–Kutta integration can improve solution accuracy [13]. Deformation in the model is directly related to spring stiffness, but the physical properties only indicate blurry correlation with spring stiffness. Therefore, it is difficult to adjust the parameters of MSD, which is a pressing concern for their application [14].

As with MSD, PBD discretizes an object as a set of node masses (figure 1), although the relationship between nodes is described as a constraint function. The constraint function is a scalar function that takes node positions as variables. When the constraint equilibrium state changes, node position is modified via constraint projection iteration to satisfy the constraint function [15]. Therefore, deformation calculation is a constraint-function optimization problem. Different constraint functions enable the realization of multiple deformation features. For example, distance and dihedral angle constraints are used to simulate cloth [16]. Strain energy constraint is used to simulate soft bodies [17]. A Darboux frame is used to simulate an elastic rod [18]. Depending on the Gauss–Seidel method used to directly modify the node position, the PBD method is unconditionally stable. Despite its many advantages, the conventional PBD method has two notable problems. One is that constraint stiffness and deformation effects depend on iteration count and timestep, while the iteration count determines the constraint convergence and computation costs. Hence, it is necessary to decouple these parameters to more easily control the model.
Figure 1. The same strain limit effects of separate description in PBD by distance constraint and in MSD by a spring damper.

The second problem is that no force or time parameters participate in the constraint projection step. Thus, it is difficult to simulate the viscoelasticity and nonlinearity of soft tissue, as these are time-related force characteristics. The distance-dependent variable constraint parameter can be used to fit the nonlinearity by controlling the node position directly [19]. However, due to the lack of time and force parameters in constraint projection step, the viscoelasticity cannot be integrated into.

This study proposes a new method for integrating viscoelastic and nonlinear springs into PBD. It realizes fast and stable simulation of rich and complex deformation behaviour, such as the viscoelasticity, nonlinearity and incompressibility of biological soft tissue. In this method, deformation effects do not depend on the iteration count and timestep. The contributions of this study are as follows:

— a new method of integrating viscoelastic and nonlinear springs into PBD;
— examples that demonstrate the validity of the proposed method compared with analytical solutions of viscoelasticity;
— a simulation of liver and gallbladder deformation, demonstrating the different structures of the organs;
— a laparoscopic cholecystectomy simulator that demonstrates the practical applicability of the proposed method.

2. Material and methods

2.1. Integration method

The objects that we aim to simulate are each composed of a set of mass points, constraints and springs. The goal of deformation calculation is to find the actual position of every mass point at each step. The position of a mass point is controlled by its total force load. For our method, based on Newton’s second law of motion

$$f_{\text{total}} = f_{\text{constraint}} + f_{\text{spring}} + f_{\text{external}} = M \ddot{x},$$

(2.1)

where $x = [x_1 \ x_2 \ \ldots \ x_n]^T$ is the column vector of the mass point position, the $\ddot{x}$ is the second-order derivative of mass point position to time which are commonly described as the acceleration of mass points. $M$ is the diagonal matrix of point masses, $f_{\text{constraint}} = [f_{c1} \ f_{c2} \ \ldots \ f_{cm}]^T$ is the column vector of constraint forces, and $f_{\text{spring}}$ and $f_{\text{external}}$ are the column vector of the spring and external forces, respectively.

To determine $f_{\text{constraint}}$, constraint energy potential is defined as

$$U(x) = \frac{1}{2} C(x)^T \alpha^{-1} C(x),$$

(2.2)

where $C(x) = [C_1 \ C_2 \ \ldots \ C_m]^T$ is a column vector of the constraint function, and $\alpha$ is a block diagonal compliance matrix corresponding to the inverse stiffness of the constraint. As for the elastic and gravity
potential, the force generated by the energy potential is in the direction of the maximum energy change to the mass point position. Thus, the constraint force is defined as the gradient of the mass point position of energy potential:
\[ \mathbf{f}_{\text{constraint}} = -\nabla \mathbf{x} = -\nabla \mathbf{C}(\mathbf{x})^\top \mathbf{a}^{-1} \mathbf{C}(\mathbf{x}). \] (2.3)

To solve the mass point position state at each timestep, we discretize equation (2.1) with timestep \( \Delta t \), using \( n \) as an index of the current timestep:
\[ \mathbf{f}_{\text{constraint}}^{n+1} + \mathbf{f}_{\text{spring}}^{n+1} + \mathbf{f}_{\text{external}}^{n+1} = \mathbf{M} \left( \frac{x^{n+1} - 2x^n + x^{n-1}}{\Delta t^2} \right). \] (2.4)

Similar to extended position-based dynamics (XPBD) [20], we introduce a Lagrange multiplier to decompose the force into its directional and scalar components. Further, we use the time interval and the inverse stiffness of constraint together to obtain \( \tilde{\alpha} = \alpha / \Delta t^2 \) for simplifying the description
\[ \lambda = -\tilde{\alpha}^{-1} \mathbf{C}(\mathbf{x}). \] (2.5)

Substituting equations (2.5) and (2.3) into equation (2.4) and rearranging the result yields
\[ \nabla \mathbf{C}(\mathbf{x}^{n+1})^\top \lambda_{n+1} = \mathbf{M}(\mathbf{x}^{n+1} - (\mathbf{x}^n + (\mathbf{x}^n - \mathbf{x}^{n-1}) + \mathbf{f}_{\text{spring}}^{n+1} + \mathbf{f}_{\text{external}}^{n+1})). \] (2.6)

The right-hand side of the equation, which represents the predicted position, can be simplified as \( \mathbf{x}^* = (\mathbf{x}^n + (\mathbf{x}^n - \mathbf{x}^{n-1}) + \mathbf{f}_{\text{spring}}^{n+1} \Delta t^2 + \mathbf{f}_{\text{external}}^{n+1} \Delta t^2) \) and is calculated prior to the constraint projection step. To determine the mass point position at timestep \( n+1 \), we must solve the following constraint-function-based minimization problem.
\[ \mathbf{g}(\mathbf{x}^{n+1}, \lambda_{n+1}) = \nabla \mathbf{C}(\mathbf{x}^{n+1})^\top \lambda_{n+1} - \mathbf{M}(\mathbf{x}^{n+1} - \mathbf{x}^*) = 0 \] (2.7)
and
\[ \mathbf{h}(\mathbf{x}^{n+1}, \lambda_{n+1}) = \lambda_{n+1} \tilde{\alpha} + \mathbf{C}(\mathbf{x}^{n+1}) = 0. \] (2.8)

Analysing the above equations, our goal is to determine \( \lambda \) and \( \mathbf{x} \) for timestep \( n+1 \). The number of functions is equal to the number of variables, so there is a unique solution. Because the constraint function is abstracted on the physical deformation behaviour, it is often not the linear form of mass point position. There is no analytical solution for mass point position directly. To solve these nonlinear equations, we linearize them with the Newton–Raphson method and solve them iteratively. We set the iteration count to \( i \) and eliminate the \( n+1 \) timestep:
\[ \mathbf{g}(\mathbf{x}_{i+1}, \lambda_{i+1}) = \mathbf{g}(\mathbf{x}_i, \lambda_i) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{g}}{\partial \lambda} \Delta \lambda = 0 \] (2.9)
and
\[ \mathbf{h}(\mathbf{x}_{i+1}, \lambda_{i+1}) = \mathbf{h}(\mathbf{x}_i, \lambda_i) + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{h}}{\partial \lambda} \Delta \lambda = 0. \] (2.10)

Our goal is to obtain \( \Delta \mathbf{x} \) and \( \Delta \lambda \) in every iteration, ultimately using these values to update \( \lambda_{i+1} = \lambda_i + \Delta \lambda \) and \( \mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x} \). This system requires two approximations (as does XPBD) that do not significantly influence the final solution. First, consider that when iteration count \( i = 0 \), \( \mathbf{g}(\mathbf{x}_i, \lambda_i) = 0 \), and for large iteration counts, \( \mathbf{g}(\mathbf{x}_i, \lambda_i) \) remains small, making \( \mathbf{g}(\mathbf{x}_i, \lambda_i) \) negligible. Second, it is expensive to determine the second-order partial derivative of the constraint function:
\[ \frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \frac{\partial \nabla \mathbf{C}(\mathbf{x})^\top \lambda_i}{\partial \mathbf{x}} - \mathbf{M}, \] (2.11)

whereas rejecting the first part of \( \partial \mathbf{g} / \partial \mathbf{x} \) has little influence on the convergence speed of the result. Therefore, it is reasonable to let \( \partial \mathbf{g} / \partial \mathbf{x} = -\mathbf{M} \). For more detail, see [20]. Next, we can solve the linearized equations (2.9) and (2.10):
\[ -\mathbf{M} \Delta \mathbf{x} + \nabla \mathbf{C}(\mathbf{x})^\top \Delta \lambda = 0 \] (2.12)
and
\[ \lambda_i \tilde{\alpha} + \mathbf{C}(\mathbf{x}_i) \Delta \mathbf{x} + \tilde{\alpha} \Delta \lambda = 0. \] (2.13)

Then, we get \( \Delta \mathbf{x} = \mathbf{M}^{-1} \nabla \mathbf{C}(\mathbf{x}_i)^\top \Delta \lambda \) from equation (2.12), and substituting it into equation (2.13), get
\[ \Delta \lambda = \frac{\lambda_i \tilde{\alpha} + \mathbf{C}(\mathbf{x})}{\nabla \mathbf{C}(\mathbf{x}) \mathbf{M}^{-1} \nabla \mathbf{C}(\mathbf{x})^\top + \tilde{\alpha}}. \] (2.14)

Once \( \Delta \lambda \) is solved, \( \Delta \mathbf{x} \) can be determined easily.
Table 1. Physical model control parameters and their corresponding descriptions.

| parameter | description               |
|-----------|---------------------------|
| $\alpha$  | constraint stiffness      |
| $b_0$     | direct velocity damp for stability |
| $b_1$     | viscoelasticity           |
| $k_1$     | nonlinear spring stiffness|
| $k_2$     | linear spring stiffness   |

Next, we integrate the MSD \( f_{\text{spring}} \), to model the nonlinearity and viscoelasticity of soft tissue. The nonlinear spring stiffness and mass point damper are introduced as follows. Here, we calculate the spring force per mass point \( i \),

\[
f_{\text{spring}} = f_{\text{damper}} + \sum_{j=0}^{k} f_{ij}
\]

and \( f_{\text{damper}} \) as

\[
f_{\text{damper}} = (b_0 + b_1 \cdot |x_i - x_0^i|) \dot{x}.
\]

This direct velocity damp force to the point mass mimics viscoelasticity [21]. The spring elastic force is defined as

\[
f_{ij} = k_{ij} \frac{X_i - X_j}{||X_i - X_j||},
\]

\[
k_{ij} = \begin{cases} k_1 \Delta l_{ij} + k_2 \Delta l_{ij}^3 & |\Delta l_{ij}| \leq \Delta l_c \\ (A + B(|\Delta l_{ij}| - |\Delta l_c|))\text{sgn}(\Delta l_{ij}) & |\Delta l_{ij}| > \Delta l_c \end{cases}
\]

and

\[
A = k_1 \Delta l_c + k_2 \Delta l_c^3, \quad B = k_1 + 3k_2 \Delta l_c^2.
\]

\( k_{ij} \) corresponds to \( \Delta l_{ij} = ||x_i - x_j|| - l_{ij}^0 \). Therefore, spring stiffness is a third-degree polynomial at low displacements, and linear at higher displacements.

The concept of our new method is to separate the behaviour model into two components: nonlinear viscoelastic MSD reflect the general physical characteristics of soft tissue; and XPBD handles complex effects, such as incompressibility and maintains the models’ stability. Control parameters are listed in table 1 for clarification.

2.2. The algorithm

Using the derivations above, we can summarize the flow of our algorithm as follows. First, nonlinearity and viscoelasticity are reflected in the spring force calculation step and the extended spring force is introduced as external force to estimate the mass position. Then, the constraint projection loop step begins to modify the position to satisfy the constraint-function equation. We use XPBD to keep the deformation effect from depending on the iterator count. We benefit from using the Gauss–Seidel method, which maintains the stability of the final mass point’s position.

From the above steps, compared to conventional PBD, the only part that is added is the spring force calculated in the predicting position step. Thus, the time cost per timestep of this method is slightly higher than PBD. However, it is still acceptable and can meet the demand of real-time deformation applications, while achieving unconditional stability and modelling more complex deformation behaviours.

2.3. Collision detection and response

A precise and fast collision detection method is crucial for avoiding penetration and triggering the deformation calculation, lest the trainee’s immersion and concentration be disrupted. For this purpose, we use the modified space hash algorithm [22]. As shown in figure 2, in our model, the simulated objects are constructed as triangles and spheres.
Algorithm

1: for all particles \( i \) do
2: initialize \( x_i = x_i^0, v_i = v_i^0, m_i = m \)
3: for all springs \( j \) do
4: initialize \( l_{\text{restlength}}, b_0, b_1, k_1, k_2 \)
5: for all constraints \( k \) do
6: initialize \( \alpha_k \) and other constraint specific parameters
7: loop (timestep = \( \Delta t \), timestep count = \( n + 1 \))
8: for all particles \( i \) do
9: predict position \( x_i^* = (x_i^n + (x_i^n - x_i^{n-1})) + f_{\text{spring}}^{n+1} \Delta t^2 + f_{\text{external}}^{n+1} \Delta t^2 \)
10: for all particles \( i \) do
11: generate collision constraint \( C_{\text{coll}} (\alpha = 0) \)
12: for all constraints \( k \) do
13: reset \( \lambda_k = 0 \)
14: loop max iteration count times
15: solve constraint function \( C \) to get \( \Delta \lambda_k, \Delta x \)
16: update \( \lambda_k = \lambda_k + \Delta \lambda_k, x^* = x^n + \Delta x \)
17: for all particles \( i \) do
18: update \( v_i = (x_i^* - x_i)/\Delta t \)
19: update \( x_i = x_i^* \)

Figure 2. Detection of deformable body collision using the modified space hash algorithm.

First, we calculate the AABB bound box of all primitives. Second, the space hash algorithm is enforced to distribute the box index to a hash index array. Third, we traverse the hash array and execute collision detection for each element with the same index. The AABB bound box is only used to project primitive
to the axial plane in the $x$-, $y$- and $z$-directions. We maintain a two-dimensional array for each primitive detection state to avoid repeat detection.

After collision detection is completed, collision response moves the primitives to a collision free state. In conventional PBD, two types of distance constraints are generated for collision response.

Sphere–sphere collisions, as in Figure 3, can be handled as a vertex distance constraint. For sphere centre points $q$ and $p$, and radii $r_1$ and $r_2$, the constraint function is

$$C(q, p) = |q - p| - r_1 - r_2. \quad (2.20)$$

Sphere–triangle collisions and triangle–triangle collisions, demonstrated in Figure 4, can be handled as a vertex and triangle distance constraint. The triangle–triangle collisions are resolved using three separate vertex and triangle distance constraints. For sphere centre point $q$; radius $r$ and triangle points $q_1$, $q_2$ and $q_3$, the constraint function should be

$$C(q, p_1, p_2, p_3) = (q - p_1) \cdot \frac{p_2 \times p_3}{|p_2 \times p_3|} - r. \quad (2.21)$$

After collision constraints are generated, they are added to later constraint projection steps for collision response calculation. For all collision constraint, $\alpha = 0$ to guarantee that the constraint equation is fulfilled as PBD.

2.4. Model example

There is a vital link between the structure of soft tissue and organ deformation behaviour. We can naturalistically simulate effects involving an organ’s topological structure by introducing corresponding geometric constraints on mass point position. This technique enables our method to quickly and stably simulate complex behaviours. As an example, we demonstrate two typical organ models: the liver and gallbladder. The liver is an entity organ, meaning that it is composed of closely arranged cells. Conversely, the gallbladder is a cavity organ, composed of stratiform distribution surface cells. We use a closed triangular surface mesh to construct the topology of the gallbladder, and a tetrahedral volume mesh to construct the topology of the liver.
In more detail, model expression is separating into three steps (Figure 5). First, the organ surface mesh models are constructed from the ‘Virtual Chinese Human’ colour slice dataset [23]. Second, for entity organs such as the liver, we generate a tetrahedral volume mesh. Finally, the basic elements for deformation calculating are generated from topology information such as nodes, springs and constraints. Different volume constraints are enforced for volume preservation when modelling the structural differences in the liver and gallbladder. Given the local volume preservation of liver, tetrahedral volume constraint is applied:

$$C(p_1, p_2, p_3, p_4) = \frac{1}{6} (p_{2,1} \times p_{3,1}) \cdot \Delta p_{4,1} - V_0.$$  \hspace{1cm} (2.22)

Similarly, to model the volume preservation of gallbladder, a closed surface mesh volume constraint is applied, as below:

$$C(p_1, p_2, \ldots, p_N) = \left( \sum_{i=1}^{n_{\text{triangles}}} (p_{i1} \times p_{i2}) \cdot p_{i3} - V_0 \right).$$ \hspace{1cm} (2.23)

In the viscoelasticity calculation step of the mass spring model (MSM), there would be superelasticity effects on the spring that load a large force near the contact point (Figure 6). To mitigate these effects, we introduce the below overstretching and compression constraints:

$$C(p_1, p_2) = (k_{\text{ratio}} l_0)^2 - ((p_1 - p_2) - l_0)^2.$$ \hspace{1cm} (2.24)

By integrating the above constraints with the viscoelastic MSM, we can naturally simulate typical soft tissue deformation effects.
3. Results

3.1. Viscoelasticity and nonlinearity

Soft tissue, as a polymer, has nonlinear, creeping and stress-relaxing characteristics. To verify that the proposed integration model can represent these features, we built a simple cube model with 60 mm long edges for testing (figure 7). Each direction is composed of six cube primitives. The centre point of the upper surface was selected as a test point. The corresponding force load and displacement constraints are applied to it.

First, we tested the nonlinearity of the model. We applied a constant displacement velocity constraint to the test point. The direction of velocity was perpendicular to the upper surface. Then, through the force balance state of the test point, we obtain the force state. Figure 8 indicates that while the displacement is small, the relationship between displacement and force is nonlinear, and then it gradually transitions to a relatively linear relationship. When the displacement triggers the overstretching constraint, the slope of the figure is flat. Meanwhile, as the displacement velocity increases, the force increases. This phenomenon is consistent with the nonlinear characteristics of soft tissue.

Second, we tested creeping (figure 9). We still applied a constant force to the test point perpendicular to the upper surface. For the same values of $b_1$, the displacement of the test point exhibits nonlinear growth over time. As $b_1$ increases, the displacement of the test point is smaller, and finally, stabilizing to equal the displacement sample. This phenomenon is consistent with the creeping characteristics of soft tissue.

Third, we tested stress relaxation (figure 10). Before force sampling, the test point is moved to a same displacement of 20 mm over 0.5 s. Then, keeping the test point fixed, its force is sampled for 2 s. In the beginning of the test, due to the momentary displacement applied on test point, the instant elastic force
loaded on the connected mass points around test point are much larger than the damping force, it causes the slight oscillatory response of the total loading force. But it quickly stabilized under the constraint of damping force by viscoelasticity. With the increase of the viscoelasticity coefficient $b_1$, the trend of force declining becomes slower and the oscillatory response becomes weaker. The oscillatory response also can be optimized by variable timestep simulation. On the general trend, the model maintains the tendency of forcing decline over time, and the decay rate becomes slower as the viscoelasticity coefficient $b_1$ increases. This phenomenon is consistent with the stress-relaxing characteristics of soft tissue.

It is not easy to find viscoelastic model parameter to directly correspond with the actual physical parameters. However, the effects of the model have the characteristic of real organs. Further, by adjusting the control parameters, the extent of the modelled organs’ viscoelasticity changes consistently. Therefore, the proposing model meets the requirement of visual plausibility for visual surgery applications.

### 3.2. Comparative analysis of liver model

PBD and MSM are chosen to compare the numerical accuracy of our nonlinear viscoelastic model with that of the existing real-time model while ensuring visualization. To determine the simulation criteria, FEBio software is used to construct the constitutive model of the liver, which is based on the finite-element model (FEM). From [24,25], the Mooney–Rivlin model with viscoelasticity is used to represent the incompressible viscoelastic characteristics of the liver. The Mooney–Rivlin model is the finite-element model using polynomial forms of the strain energy. By extending the basic neo-Hookean model with the addition of the second invariant of the right Cauchy–Green strain tensors. It can be used
Table 2. The parameter description of finite-element model.

| type             | description                          | parameter name | value  |
|------------------|--------------------------------------|----------------|--------|
| hyperelasticity  | 5-Constant Mooney–Rivlin             | $c_1$          | 242.3  |
|                  |                                      | $c_2$          | −53.4  |
|                  |                                      | $c_3$          | 4140.1 |
|                  |                                      | $c_4$          | −8583.5|
|                  |                                      | $c_5$          | 4419.6 |
| viscoelasticity  | relaxation times                     | $r_1$          | 0.27   |
|                  |                                      | $r_2$          | 0.03   |
|                  | viscoelastic coefficients            | $g_1$          | 0.235  |
|                  |                                      | $g_2$          | 0.333  |

to model material that exhibits some limited compressibility. The Mooney–Rivlin model with 2-Constant parameters models rubber well. The 5-Constant parameters of the Mooney–Rivlin model can be used to model liver. With the additional option of viscoelasticity, the soft tissue characteristics of liver can be reflected in a more comprehensive way. In table 2, the parameters of Mooney–Rivlin model of 5-Constant and viscoelastic model of 4-Constant are determined. During the test, the right part of the liver is enforced fixed constraints. The left part of liver is loaded by a 5 N force (figure 11e). Then, we record the final stable state of each model for comparison. From the rendered picture of the liver, our model is closest to FEM (figure 11f). The MSM model cannot ensure the incompressible which with large local deformation (figure 11h), and the deformation of PBD is relatively small owing to the strong constraint applied (figure 11g). In order to measure the global effect of each model, the whole surface node distance difference parameter $Q$ is calculated, which is defined by the adding each node distance to the FEM standard.

$$Q = \sum_{i=0}^{n} |p_i^{\text{model}} - p_i^{\text{FEM}}|.$$  

(3.1)

Smaller $Q$ demonstrates a smaller overall variance to the FEM model. It is found that the value of $Q$ of our model is 2.57, that of PBD is 4.75 and that of MSM is 6.20. The above results show that our model has higher accuracy, while ensuring visual effects.

3.3. Volume preservation

The structural features of the entity organs and cavity organs can be expressed by two different volume constraints. For the volume preservation effects of entity organs, we apply a local volume constraint. The tetrahedral volume constraint only influences the node position of related tetrahedra. Here, we use the cantilever beam model for testing by detecting the volume change of a cantilevered beam as it sags under gravity (figure 12). From the initial state ($t = 0$ s) to the natural sag ($t = 1$ s), the volume change ratio remained relatively low (figure 13). For volume preservation in cystic organs, we apply a global volume constraint. The closed surface volume constraint influences the position of all surface nodes at the same time. In this study, we use a falling gallbladder model for testing, by detecting the volume change of a gallbladder falling and pressed by a plane (figure 14). Owing to the enforcement of the global volume constraint, although the volume change ratio increases, it maintains an acceptable level (figure 15).

The two volume constraints above can constrain the volume of the model within acceptable levels, thus reflecting the structural features of the corresponding organ.

3.4. Stability

It is critical that soft tissue deformation methods for real-time virtual surgery applications can meet stability requirements. If unrealistic organ explosion occurs during interaction between the organs and instruments, the trainee’s experience and immersion will be poor. Numerical integration errors make it easy to trigger such phenomena in traditional mass spring methods. By integrating PBD constraints into the spring force solution, we improve the stability of the model. In figure 16, we use an abnormal
displacement constraint to produce an unrealistic model. Then, we release the constraint to obtain the recovery measure of the model. The result shows that our model is more stable than a conventional MSM.

3.5. Virtual laparoscopic cholecystectomy application

To verify the practicality of our method, we constructed a laparoscopic virtual surgery application. We built physical models for two important organs, the liver and gallbladder, using the proposed method.
Viscoelastic springs, and local volume and overstretching constraints were used to model the entity organ (in this case, the liver). Viscoelastic springs, and global volume and overstretching constraints were used to model the cystic organ (the gallbladder). Our application was built on the Unity3D game engine [26]. The system runs on a Windows platform with Intel Core i7-7700 CPU @ 3.6 GHz and NVIDIA GTX 1070 GPU. The program runs in a single thread without multiple thread optimization for comparing computational costs. The key modules of the system include collision detection and deformation calculation, scene rendering and manipulator data acquisition and processing. The three
Figure 16. The cube model is compressed to an unstable state: (a–c) reflects that our model can recover its original shape; (d–f) reflects the inability of the conventional MSD to recover.

Figure 17. Laparoscopic scene views: (a) the resting state of the liver and gallbladder, (b) the gallbladder is dragged by its membrane with the liver using the clamp and (c) interactive operation clamping on the gallbladder.
steps in the cholecystectomy are shown in figure 17. Surgical scene information was recorded by the Unity3D frame rate profiler tool (table 3), indicating that a high frame rate (greater than 30 fps) was maintained during the operation. In figure 18, the average time cost per frame of three steps in figure 17 was recorded. From step (a) to step (b), and step (b) to step (c), the contacts between instruments and organs are increasing. Owing to collision response, the increase in constraint numbers leads to a slight increase in time cost. Nevertheless, the total time cost per frame is always below the real-time standard requirements (30 fps). Meanwhile, the fluctuation of time cost during the operation is small.

### 4. Conclusion

To simulate the viscoelasticity, nonlinearity and incompressibility of soft tissue deformation in real time, we have derived a new PBD method that integrates viscoelastic MSD. In this method, viscoelasticity and other time-related characteristics are simulated based on MSD. Then, the spring and external forces are combined with a constraint force generated by a PBD constraint function to correct the movement of particles. This method succeeds in controlling complex deformation behaviour through multiple constraint types, and is independent of iteration count and unconditionally stable. We generated a series of simple examples, which indicated our ability to control the extent of viscoelasticity through parameter adjustment. Finally, we applied this method to simulate soft tissue deformation in a laparoscopic cholecystectomy. This simulation showed the realistic deformation effects of the liver and gallbladder, and verified the practicality of the proposed method. The new method can meet the demand of modelling soft tissue deformation in real-time applications such as virtual surgery.

In future work, we plan to build a viscoelastic spring based on PBD through more controllable distance constraints. The XPBD framework has shown that the periodicity of a spring oscillator can be

![Figure 18. Average time cost per frame of corresponding steps in figure 17.](image)

Table 3. Constructed data and time information captured from the Unity3D profiler.

| item                  | category | value         |
|-----------------------|----------|---------------|
| geometry data         | vertex   | 96.9 k units  |
|                       | triangle | 157.4 k units |
| physical data         | node     | 650 units     |
|                       | constraint | 3163 units   |
|                       | spring   | 1938 units    |
| collision primitive   | triangle | 848 units     |
|                       | sphere   | 248 units     |
| time cost             | render   | 0.3 ms        |
|                       | physic   | 20.51 ms      |
|                       | collision| 3.45 ms       |
simulated by adjusting the parameters corresponding to the inverse stiffness of a realistic spring. Under extreme deformations, such as momentary displacement or force, the system has slight oscillations, which can be improved by introducing variable timestep simulation. Meanwhile, the variable timestep can be used for reducing time cost. Also, force feedback plays a significant role in measuring the effectiveness of the operation and enhancing the immersion of the trainers. Using the method proposed above, the feedback force in the deformation process can be easily solved by analysing the motion state of mass points. In the future application, we will add force feedback devices such as servo motors to produce reaction forces on the medical instrument and study the numerical accuracy of feedback force. Further, it would be worthwhile to build an overall relationship between model and physical parameters, as in the case of Young’s modulus or Poisson’s ratio.

Data accessibility. FEBio software suite: (https://febio.org/) 3D Organ model data: liver and gallbladder (http://dx.doi.org/10.5061/dryad.1176k) [27].

Authors’ contributions. L.X. and Y.L. performed FEM simulations for comparison study. L.X. wrote the main body of the article. Q.L. edited and finalized the article. All authors discussed the results and commented on the manuscript at all stages. All authors gave final approval for publication.

Competing interests. The authors declare no competing interests.

Funding. This work was supported by the National High Technology Research and Development Programme of China (grant no. 2012AA02A606).

Acknowledgement. The authors thank FEBio software suite for providing the nonlinear finite-element analysis tool in biomechanics.

References

1. Nagendran M, Gunusamy KS, Aggarwal R, Loizidou M, Davidson BR. 2013 Virtual reality training for surgical trainees in laparoscopic surgery. Cochrane Libr. 8, CD006575. (doi:10.1002/14651858.CD006575.pub2)

2. Moglia A, Ferrari V, Morelli L, Ferrari M, Mosca F, Cuschi eni A. 2016 A systematic review of virtual reality simulators for robot-assisted surgery. Eur. Urol. 69, 1065–1080. (doi:10.1016/j.eururo.2015.09.021)

3. Dagdeviren C et al. 2015 Conformal piezoelectric systems for clinical and experimental characterization of soft tissue biomechanics. Nat. Mater. 14, 728–736. (doi:10.1038/nmat4289)

4. Moore P, Molloy D. 2007 A survey of computer-based deformable models. In Machine Vision and Image Processing Conf., Kildare, Ireland, 5–7 September, pp. 55–66. IEEE. (doi:10.1109/MIVIP.2007.31)

5. Fan R, Sacks MS. 2014 Simulation of planar soft tissues using a structural constitutive model: finite element implementation and validation. J. Biomech. 47, 2043–2054. (doi:10.1016/j.jbiomech.2014.03.014)

6. Zhang G, Wittek A, Iolds J, Xin X, Miller K. 2014 A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue. Eng. Anal. Boundary Elem. 42, 60–66. (doi:10.1016/j.enganabound.2013.08.004)

7. Yu D, Wang M, Liu QG. 2015 Development of Chinese reference man deformable surface phantom and its application to the influence of physique on electromagnetic dosimetry. Phys. Med. Biol. 60, 6831–6846. (doi:10.1088/0031-9155/60/17/6833)

8. Bardenhagen SG, Kober EM. 2004 The Generalized interpolation material point method. Comput. Method. Eng. Sci. 5, 477–496.

9. Cotin S, Delingette H, Ayache N. 1999 Real-time elastic deformations of soft tissues for surgery simulation. IEEE Trans. Vis. Comput. Graph 5, 62–73. (doi:10.1109/2945.764872)

10. Muller M, Heidelberger B, Hennix N, Ratcliff J. 2007 Position based dynamics. J. Vis. Commun. Image Represent. 18, 109–118. (doi:10.1016/j.jovcirt.2007.01.005)

11. Dochev V, Vasilev T. 2003 Efficient super-elasticity handling in mass-spring systems. In Int. Conf. on Computer Systems and Technologies, pp. 483–490. Rousse, Bulgaria: ACM. (doi:10.1145/793620.793701)

12. Mollemans W, Schutys F, Van Cleynenbreugel J, Suetsens P. 2003 Tetrahedral mass spring model for fast soft tissue deformation. In Information Systems, pp. 145–154. Juan-Les-Pins, France: Springer-Verlag.

13. Cotin S, Delingette H, Ayache N. 2000 A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. Vis. Comput. 16, 437–452. (doi:10.1007/PL00007215)

14. Bianchi G, Solenthaler B, Szelyl G, Haders M. 2004 Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. In Medical Image Computing and Computer Assisted Intervention – MICCAI 2004. Lecture Notes in Computer Science, vol. 3217, pp. 47–54. Berlin, Germany: Springer.

15. Bender J, Muller M, Otaduy MA, Teschner M, Macklin P. 2014 A survey on position-based simulation methods in computer graphics. Comput. Graph Forum 33, 228–251. (doi:10.1111/cgf.12346)

16. Macklin M, Ller M, Chentanez N, Kim TY. 2014 XPBD: Interactive Techniques, pp. 47. Vancouver, Canada: ACM. (doi:10.1145/2641086.2641585)

17. Muller M, Heidelberger B, Hennix N, Ratcliff J. 2007 Position based dynamics. J. Vis. Commun. Image Represent. 18, 109–118. (doi:10.1016/j.jovcirt.2007.01.005)

18. Duan Y, Huang W, Chang H, Chen W, Zhou L, Tea SK, Su Y, Cui CK, Chang S. 2016 Volume preserved mass-spring model with novel constraints for soft tissue deformation. IEEE J. Biomed. Health Inform. 20, 268–280. (doi:10.1109/JBMI.2014.2370059)

19. Macklin M, Muller M, Chentanez N. 2016 XPBD: Position-based simulation of compliant constrained dynamics. In Motion in Games, pp. 49–54. Burlingame, California: ACM. (doi:10.1145/2944258.2944272)

20. Basafa E, Farahmand F. 2017 Real-time simulation of the nonlinear visco-elastic deformations of soft tissues. Comput. Assist. Radiol. Surg. 6, 297–307. (doi:10.1186/s11548-010-0508-6)

21. Teschner M, Heidelberger B, Muller M, Pomerantes D, Gross M. 2003 Optimized spatial hashing for collision detection of deformable objects. In Vision Modeling and Visualization.

22. Tang L, Chung MS, Liu Q, Shin DS. 2010 Advanced features of whole body sectional images: virtual Chinese human. Clin. Anat. 23, 523–529. (doi:10.1002/ca.20975)

23. Fu YB, Chui CK. 2014 Modelling and simulation of porcine liver tissue indentation using finite element method and uniaxial stress–strain data. J. Biomech. 47, 2430–2435. (doi:10.1016/j.jbiomech.2014.04.009)

24. Macklin M, Ller M, Chentanez N, Kim TY. 2014 XPBD: Interactive Techniques, pp. 47. Vancouver, Canada: ACM. (doi:10.1145/2641086.2641585)

25. Stefanei M, Simon C, Herwe D. 2017 Non linear biomechanical model of the liver: Hyperelastic constitutive laws for finite element modeling. In Biomechanics of living organs, pp. 602. Amsterdam, The Netherlands: Elsevier.

26. Unity3D. 2017. Available from https://www.unity3d. com (accessed 31 May 2017).

27. Xu L, Liu Y, Liu Q. 2018 Data From: Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time. Dryad Digital Repository. (doi:10.5061/dryad.1176k)