Hindawi
Scientific Programming
Volume 2021, Article ID 8343312, 16 pages
https://doi.org/10.1155/2021/8343312

Research Article

A Hybrid Model Method for Accurate Surface Deformation and Incision Based on FEM and PBD

Shijie Tan,[1] **Hongjun Zhou**,[2] **and Jinjin Zheng**[1]

[1]*Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei 230001, China*
[2]*National Synchrotron Radiation Laboratory, University of Science and Technology of China, Hefei 230001, China*

Correspondence should be addressed to Shijie Tan; tsj@mail.ustc.edu.cn and Jinjin Zheng; jjzheng@ustc.edu.cn

Received 25 May 2021; Accepted 17 November 2021; Published 15 December 2021

Academic Editor: Tomàs Margalef

In some simulations like virtual surgery, an accurate surface deformation method is needed. Many deformation methods focus on the whole model swing and twist. Few methods focus on surface deformation. For the surface deformation method, two necessary characteristics are needed: the accuracy and real-time performance. Some traditional methods, such as position-based dynamics (PBD) and mass-spring method (MSM), focus more on the real-time performance. Others like the finite element method (FEM) focus more on the accuracy. To balance these two characteristics, we propose a hybrid mesh deformation method for accurate surface deformation based on FEM and PBD. Firstly, we construct a hybrid mesh, which is composed of a coarse volume mesh and a fine surface mesh. Secondly, we implement FEM on coarse volume mesh and PBD on fine surface mesh, and the deformation of fine surface mesh is constrained by the displacement of the coarse volume mesh. Thirdly, we introduced a small incision process for our proposed method. Finally, we implemented our method on a simple deformation simulation and a small incision simulation. The result shows an accurate surface deformation performance by implementing our method. The incision effect shows the compatibility of our proposed method. In conclusion, our proposed method acquires a better trade-off between accuracy and real-time performance.

1. Introduction

The mesh deformation algorithm plays a very important role in computer simulation technique, such as computer animation and soft tissue deformation simulation in virtual reality. In some simulations like virtual surgery, the surface deformation plays an important part in realistic perform effect. Generally speaking, two major problems need to be solved for surface deformation algorithm: low time consumption and high accuracy. Although many methods focus on one of these two problems, there is a huge demand for high accuracy deformation under real-time performance. Therefore, a suitable deformation algorithm needs a trade-off between the accuracy requirement and the real-time performance.

The finite element method (FEM) is usually used for the numerical discrete model of constitutive equation in elastic mechanics [1]. The standard method is used to create physical elements according to each geometric element and then to calculate the displacement of relevant element nodes according to the applied load [2]. The boundary conditions need to be well treated. The FEM can build different kinds of physical elements according to corresponding geometric units, such as triangle element in 2D and tetrahedral element in 3D [3]. The advantage of FEM is that it is close to the real world since it utilizes the actual physical parameters to get output with high accuracy of deformation. It is also able to simulate the complex mesh model [4]. However, FEM costs more computational time. The real-time performance needs to be improved [5]. At present, FEM is widely used in mesh deformation simulations. Paulus et al. proposed a new remesh method and rapid FEM for soft tissue simulation [6]. Courtecuisse et al. demonstrated a real-time tissue simulation based on the implicit numerical integration method in

nonlinear FEM [7]. Haouchine et al. proposed a method to enhance the internal structure of the liver mesh model in real time [8]. Tang and Wan proposed a constraint FEM for the interactive simulation for virtual surgery [9]. Kugelstadt et al. proposed a fast co-rotated FEM using operator splitting for whole mesh model swinging and twisting [10]. It acquires a good trade-off between accuracy and low time consumption but is not compatible with mesh dissection (model incision).

Position-based dynamics (PBD) has been widely used in recent years. Classical model methods such as FEM and MSM calculate the deformation according to Newton's second law. Compared with these methods, PBD directly calculates the position change based on the solution of the quasi-static problem. Therefore, PBD has better real-time performance and stability [11]. However, PBD is generally not as accurate as classical model methods. In recent years, PBD has been widely used in deformation simulations. Kubiak et al. proposed a surgical line simulation method based on the PBD method of Müller et al. [12, 13]. In order to assist the robot-assisted rehearsal and planning of partial nephrectomy, Camara et al. proposed a real-time simulation platform that allows surgeons to quickly construct a biomechanical model [14]. Pan et al. proposed an interactive method of the hybrid soft tissue model based on extended PBD [15]. Liu et al. proposed a PBD method for tetrahedral element mesh model to simulate the deformation on soft tissue [16].

For the mass-spring method (MSM), it has many advantages such as easy operation, simple modeling, and low time consumption. However, MSM needs suitable constraints. Unsuitable constraints will lead to fault and divergence of deformation results. In MSM, it is important to determine the spring parameters which include stiffness and damping coefficient. These parameters determine the accuracy of MSM. Due to the random selection of these parameters, the accuracy of deformation results is not high. Two ways of parameter selection are data measurement and formula analysis. Data measurement estimates the parameters by measuring the deformation data of real biological tissues. The formula analytical method is based on the constitutive equation of the material to deduce the parameters [17]. The MSM also has a wide range of applications. Skornitzke et al. proposed a mitral valve MSM model [18]. Gao et al. simulated the nonlinear anisotropy of biological materials based on MSM [19]. Zhang et al. proposed a three-stage method based on MSM to simulate soft tissue [20].

In this paper, we propose a hybrid mesh deformation method for accurate surface deformation based on FEM and PBD. Our proposed method combined the advantages of FEM and PBD to satisfy the requirement of accuracy and real time. First, we construct the hybrid mesh which composes a fine surface mesh and a coarse volume mesh. Then, the FEM and PBD is utilized to simulate deformation on the volume mesh and surface mesh, respectively. Finally, the deformation of surface mesh is constrained on the volume mesh. We also introduced a small incision process which is compatible with our proposed method.

The reminder of this paper is organized as follows: Section 2 briefly explains purpose of our proposed method. Sections 3 and 4 describe the mathematical schemes including FEM and PBD. Section 5 introduces the construction of the hybrid mesh. Section 6 presents the algorithm flow of our proposed method. Section 7 introduces the small incision processing for our proposed method. Section 8 introduces the experiment environments, shows the rendering effects and results of deformation, and demonstrates the compatibility of small incision processing. In Section 9, we conclude our work.

2. Overview

In previous work, FEM is used for high accuracy in fine mesh deformation but usually has high time consumption. PBD has the characteristics of low time consumption and low accuracy. In our method, FEM is used to calculate the deformation of the coarse tetrahedral mesh, and the PBD is used to perform the deformation of the fine surface triangle mesh. The nodes of the volume mesh which connect to the surface mesh are control points (black points in Figure 1). These points constrain the surface mesh and make its deformation within a more accurate range. Meanwhile, since the FEM is applied to the coarse tetrahedral mesh and the surface mesh uses the PBD, the time consumption is not too high. Thus, the real-time requirements can be met.

3. Finite Element Method

Our method uses a set of tetrahedral elements to separate the 3D domain. Based on this spatial discretion, the node displacement interpolation method is used to approximate the continuous displacement field. Specifically, according to equation (1), using the node of elements, the shape function is interpolated to form the displacement field function of the mesh element.

$$\mathbf{u}_e(x) = \mathbf{N}^e(x)\mathbf{u}^e. \tag{1}$$

$\mathbf{N}^e(x)$ is the shape function matrix of elements, and $\mathbf{u}^e$ is the displacement of the mesh element node. We choose linear interpolation as the shape function of the tetrahedron discretion. Through the shape function, the relationship between the strain and the displacement of the element node is established:

$$\boldsymbol{\varepsilon}_e(x) = \mathbf{B}^e(x)\mathbf{u}^e. \tag{2}$$

$\mathbf{B}^e(x)$ is the strain matrix of elements. According to the principle of virtual work, the relationship between nodal displacement $\mathbf{u}^e$ and external nodal force of elements $\mathbf{f}^e$ is in

$$\mathbf{K}^e\mathbf{u}^e = \mathbf{f}^e, \tag{3}$$

where $\mathbf{K}^e$ is the stiffness matrix of elements (equation (4)) and $\mathbf{D}$ is the elastic matrix:

$$\mathbf{K}^e = \int \mathbf{B}^{eT}\mathbf{D}\mathbf{B}^e \mathrm{d}x. \tag{4}$$

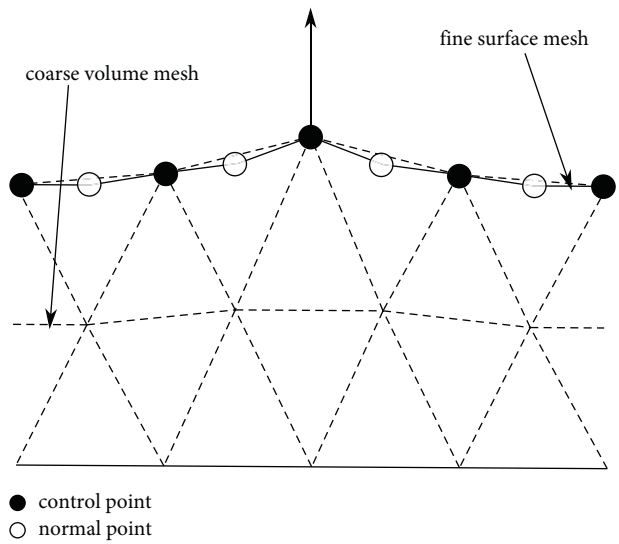


Figure 1: Overview of our proposed method for accurate surface deformation.

We combine each element stiffness matrix and element external force according to the global index of the mesh nodes to obtain the global stiffness matrix $\mathbf{K}$ and global external force $\mathbf{f}$ for the whole mesh. The relationship between global node displacement $\mathbf{u}$ and global external force vector $\mathbf{f}$ can be obtained in

$$\mathbf{K}\mathbf{u} = \mathbf{f}. \tag{5}$$

When it is necessary, solve equation (5) to calculate the deformation displacement $\mathbf{u}$. $\mathbf{K}$ is determined in preprocessing which costs no time in deformation calculation.

4. Position-Based Dynamics

In PBD, it is necessary to predict the position and velocity of each node and update the nodal displacement to proper position by constraint function. The establishment of the constraint function affects the stability and efficiency of the model. Thus, solving the constraint function becomes the most important part of PBD. Let the constraint function be $C(\mathbf{P}) = 0$. For the nodal predicted position $\mathbf{P}$, the nodal position projected by the constraint functions is $\mathbf{P} + \Delta\mathbf{P}$. So, constraint function $C(\mathbf{P} + \Delta\mathbf{P})$ is

$$C(\mathbf{P} + \Delta\mathbf{P}) \approx C(\mathbf{P}) + \nabla_P C(\mathbf{P})\Delta\mathbf{P}, \tag{6}$$

where $\Delta\mathbf{P}$ is the correction of $\mathbf{P}$ after the constraint projection and $\nabla_P C(\mathbf{P})$ is the gradient of the constraint function. In order to maintain the conservation of momentum and angular momentum, $\Delta\mathbf{P}$ and $\nabla_P C(\mathbf{P})$ have the same direction. Thus, a Lagrangian multiplier λ can be used to express $\Delta\mathbf{P}$ and $\nabla_P C(\mathbf{P})$ relationship:

$$\Delta\mathbf{P} = \lambda\nabla_P C(P) = -\frac{C(\mathbf{P})}{\left|\nabla_P C(\mathbf{P})\right|^2}\nabla_P C(\mathbf{P}). \tag{7}$$

In our method, we utilize two constraints to perform the deformation of fine surface mesh: stretch constraint and bending constraint. Two constraints are as follows.

4.1. Stretch Constraint. As Figure 2 shows, the stretch constraint simulates the elastic force between two nodes.

The stretch constraint represents equation (8), where l is the initial length of $\mathbf{P}_1\mathbf{P}_2$.

$$C_{\text{stretch}}(\mathbf{P}_1, \mathbf{P}_2) = \left|\mathbf{P}_1 - \mathbf{P}_2\right| - l. \tag{8}$$

According to equation (7), we obtained $\Delta\mathbf{P}_1$ and $\Delta\mathbf{P}_2$ in equation (9), where $w_i = 1/m_i$ $(i = 1, 2)$. m_1 and m_2 are the mass of $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\begin{aligned}\Delta\mathbf{P}_1 &= -\frac{w_1}{w_1 + w_2}\left(\left|\mathbf{P}_1 - \mathbf{P}_2\right| - l\right)\frac{\mathbf{P}_1 - \mathbf{P}_2}{\left|\mathbf{P}_1 - \mathbf{P}_2\right|},\\ \Delta\mathbf{P}_2 &= +\frac{w_2}{w_1 + w_2}\left(\left|\mathbf{P}_1 - \mathbf{P}_2\right| - l\right)\frac{\mathbf{P}_1 - \mathbf{P}_2}{\left|\mathbf{P}_1 - \mathbf{P}_2\right|}.\end{aligned} \tag{9}$$

4.2. Bending Constraint. As Figure 3 shows, the constraint simulates the bending between two triangle elements.

The bending constraint function is

$$C_{\text{bending}}(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4) = \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) - \varphi, \tag{10}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are the normal of $\Delta\mathbf{P}_1\mathbf{P}_3\mathbf{P}_2$ and $\Delta\mathbf{P}_1\mathbf{P}_2\mathbf{P}_4$ and φ is the initial angle between $\Delta\mathbf{P}_1\mathbf{P}_3\mathbf{P}_2$ and $\Delta\mathbf{P}_1\mathbf{P}_2\mathbf{P}_4$. According to equation (7), we obtained $\Delta\mathbf{P}_i$ $(i = 1, 2, 3, 4)$ in

$$\Delta\mathbf{P}_i = -\frac{4w_i}{\sum_j w_j}\frac{\sqrt{1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)^2\left(\arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) - \varphi\right)}}{\sum_j \left|q_j\right|^2}\mathbf{q}_i, \tag{11}$$

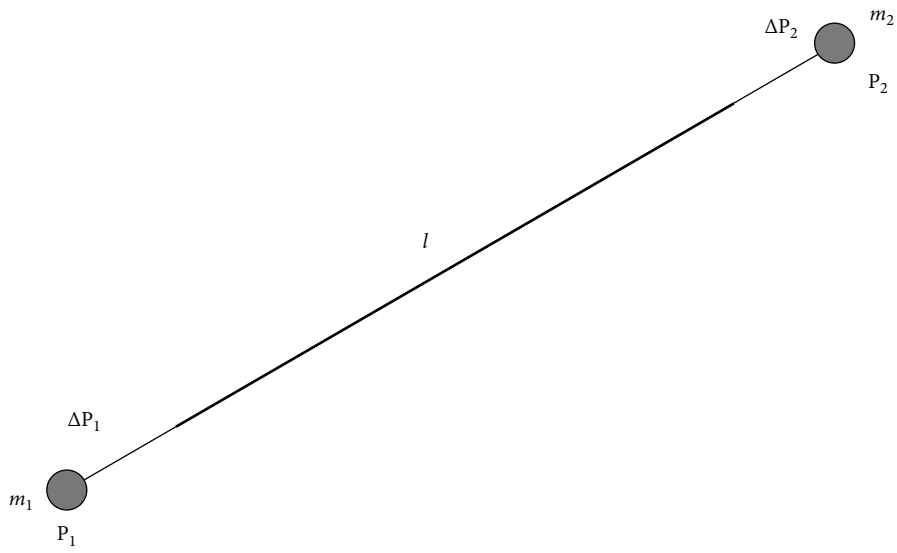


Figure 2: Stretch constraint.

where $w_i = 1/m_i$, m_i is the mass of $\mathbf{P}_i$, and $\mathbf{q}_i$ $(i = 1, 2, 3, 4)$ is

$$\begin{aligned}
\mathbf{q}_1 &= -\mathbf{q}_2 - \mathbf{q}_3 - \mathbf{q}_4, \\
\mathbf{q}_2 &= -\frac{\mathbf{P}_3 \times \mathbf{n}_2 + (\mathbf{n}_1 \times \mathbf{P}_3)(\mathbf{n}_1 \cdot \mathbf{n}_2)}{|\mathbf{P}_2 \times \mathbf{P}_3|} - \frac{\mathbf{P}_4 \times \mathbf{n}_1 + (\mathbf{n}_2 \times \mathbf{P}_4)(\mathbf{n}_1 \cdot \mathbf{n}_2)}{|\mathbf{P}_2 \times \mathbf{P}_4|}, \\
\mathbf{q}_3 &= \frac{\mathbf{P}_2 \times \mathbf{n}_2 + (\mathbf{n}_1 \times \mathbf{P}_2)(\mathbf{n}_1 \cdot \mathbf{n}_2)}{|\mathbf{P}_2 \times \mathbf{P}_3|}, \\
\mathbf{q}_4 &= \frac{\mathbf{P}_2 \times \mathbf{n}_1 + (\mathbf{n}_2 \times \mathbf{P}_2)(\mathbf{n}_1 \cdot \mathbf{n}_2)}{|\mathbf{P}_2 \times \mathbf{P}_4|}.
\end{aligned} \tag{12}$$

5. Construction of the Hybrid Mesh

The hybrid mesh includes a fine triangle surface mesh and a matching internal coarse tetrahedral volume mesh. Now, given a fine triangle surface mesh, the steps to generate the hybrid mesh are as follows:

(i) Step 1: input the fine triangle surface mesh M_S

(ii) Step 2: simplify M_S to a coarse triangle surface mesh M_S'

(iii) Step 3: according to Delaunay algorithm offered in open-source software Tetgen, use M_S' to generate a coarse tetrahedral volume mesh M_V

(iv) Step 4: combine M_S and M_V to obtain the hybrid mesh

The Illustration of these steps is shown in Figure 4. Since Step 3 uses the open-source algorithm, we do not introduce it here [21]. Now, Step 2 is introduced as follows. Given M_S, it is necessary to select some nodes from M_S as the nodes of the coarse surface triangle mesh M_S'. The selection of these nodes is based on the principle of minimum vertices cover; i.e., some nodes in M_S are selected to a node set V, so that each edge in M_S contains at least one node in V and the number of nodes in V is minimized. We use the method in [22] to obtain the V.

After V is obtained, the way to simplify M_S to M_S' is introduced in the following. As shown in Figure 5, for one node p_i in M_S and not in V, a node p_j which is in V and adjacent to p_i is chosen. After that, p_i is merged with p_j and the edge p_ip_j is removed. For all adjacent vertices p_k of p_i, if p_k is also adjacent to p_j, the edges p_ip_k are also removed. Otherwise, a new edge p_kp_j will be created after p_ip_k removed.

Then, the selection of p_j will become a problem. In order to find a proper p_j, we first calculate the average distance d_{avg} between p_i and p_k:

$$d_{\text{avg}} = \frac{1}{n} \sum_{k}^{n} d_{ik}, \tag{13}$$

where d_{ik} is the distance between p_i and p_k. And then, we select p_j that satisfies

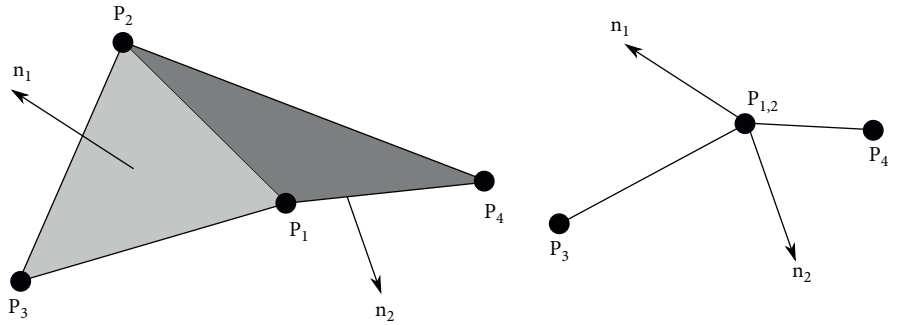


Figure 3: Bending constraint.

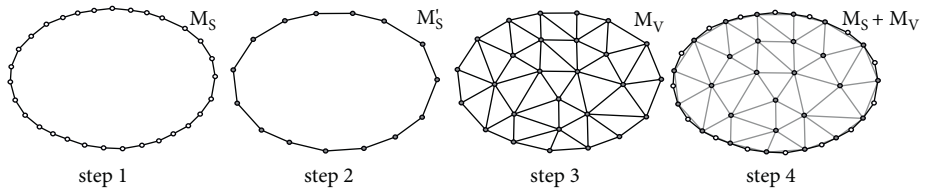


Figure 4: Construction of the hybrid mesh.

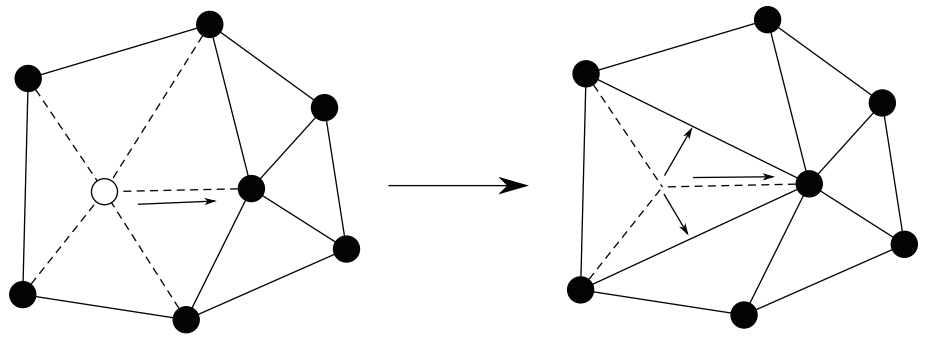


Figure 5: Simplification of M_S to M_S'.

$$d_{ij} = \min_k \left(\left| d_{ik} - d_{\text{avg}} \right| \right). \tag{14}$$

The node p_j is selected in this condition in order to guarantee a better mesh quality after simplification. When all the nodes p_i are simplified in this way, a coarse surface triangle mesh M_S' is obtained. Step 2 is now complete.

6. Deformation Algorithm Based on the Hybrid Mesh

The algorithm includes processing of external force, computing the deformation of the coarse volume mesh (M_V) using FEM, deformation of M_V, position prediction of each node on the fine surface mesh (M_S), and constraint projection on M_S. Figure 6 shows the illustration of the deformation algorithm on the 2D sectional view of the hybrid mesh. The black points indicate the shared nodes of M_V and M_S. These nodes are obtained from the M_S simplification in Section 5. We call these nodes *control nodes/points*. Control nodes share the same position information with nodes on the boundary of M_V. The white points represent normal nodes in M_S. M_V is represented by dotted lines, which means that it is not rendered in graphics. The solid line at the bottom of M_V indicates that it is subject to displacement constraints.

6.1. Processing of External Force. Figure 6(a) shows that a vertical upward force is applied to the hybrid mesh. Since M_S has many nodes, the default load point is on the node. If not, snap the load point to the closest node. After snapping, as Figure 6(b) shows, the position of load point may not be on the control node. In this condition, as Figure 6(c) shows, the equivalent force on the surrounding nodes of the external force is calculated for M_V. The relationship between equivalent force and external force satisfies linear interpolation as the shape function of the triangular element.

6.2. Deformation of M_V. After processing of external force, deformation of M_V is solved according to equation (5). Control nodes share the same position with the nodes on the boundary of M_V, so the position of control nodes is also

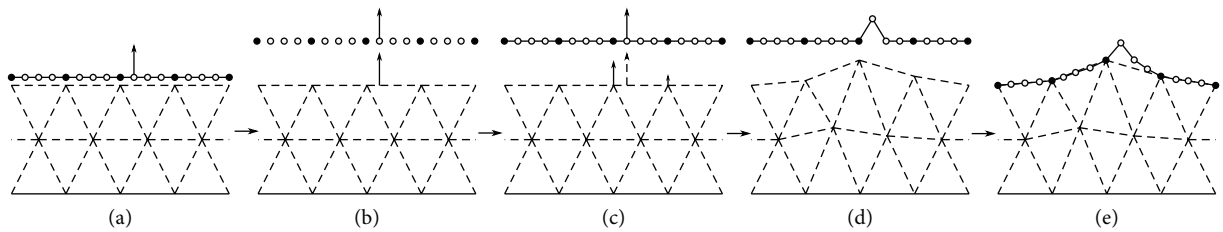


Figure 6: Overview of our proposed deformation algorithm. (a–c) Processing of external force. (d) Deformation of M_V and position prediction of M_S. (e) constraint projection of M_S.

obtained. Solving equation (5) will not take too long to run since M_V is coarse.

6.3. Position Prediction of M_S. For position prediction of M_S, like PBD, the node position changes under the external load, without considering the influence of internal forces between the nodes in M_S. This is node position prediction. The final node position of M_S will be determined after constraint projection. Take the M_S in Figure 6(d) as an example, where only one node is affected by the external load. Then, its position changes. The other vertices are not affected by external loads, so their positions have not changed. In this time, the nodes on M_S are ready for constraint projection.

6.4. Constraint Projection of M_S. As Figure 6(e) shows, since control nodes share the same position on M_S and the boundary of M_V, the displacement of control nodes in M_S is determined when deformation of M_V completes. In addition, constraint projection is also conducted for position correction, and the deformation position of normal nodes in M_S is determined. Since only M_S participates in graphic rendering and M_V does not, this is the final state of hybrid mesh at the current moment.

6.5. Algorithm Flow

Step 1. Initialization: compute the global stiffness matrix $\mathbf{K}$ of M_V, and initialize the position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$ of each node of M_S and the external force $\mathbf{f}$.

Step 2. In the current Δt, get $\mathbf{f}$, compute the deformation of M_V according to equation (11), obtain the position of nodes $\mathbf{p}_i'$ on the boundary of M_V, and update the velocity $\mathbf{v}_i^{\text{new}}$ of the nodes on M_S according to

$$\mathbf{v}_i^{\text{new}} = \mathbf{v}_i + \mathbf{f}\Delta t w_i, \tag{15}$$

where $w_i = 1/m_i$ (m_i is the node mass of M_S).

Step 3. Predict the position of each node $\mathbf{p}_i^{\text{new}}$ on M_S according to

$$\mathbf{p}_i^{\text{new}} = \mathbf{p}_i + \mathbf{v}_i^{\text{new}}\Delta t. \tag{16}$$

Step 4. Project constraints on M_S: this step has two cases. For control nodes, $\mathbf{p}_i^{\text{solve}} = \mathbf{p}_i'$. For other nodes on M_S, solve position correction $\Delta\mathbf{p}_i$ on M_S according to equations (9) and (11) and obtain the $\mathbf{p}_i^{\text{solve}}$ according to

$$\mathbf{p}_i^{\text{solve}} = \mathbf{p}_i + \Delta\mathbf{p}_i. \tag{17}$$

Step 5. According to equation (18), update the $\mathbf{p}_i$ and $\mathbf{v}_i$ of each node on M_S:

$$\mathbf{v}_i = \frac{\left(\mathbf{p}_i^{\text{solve}} - \mathbf{p}_i\right)}{\Delta t}, \qquad \mathbf{p}_i = \mathbf{p}_i^{\text{solve}}. \tag{18}$$

Step 6. Start the next Δt and return to Step 2.

7. Small Incision Processing

In some deformation simulations, operations such as incision on mesh are necessary. Generally, the incision on the surface of a model is small, so the incision displacement of M_V can be ignored. Moreover, M_V is not rendered in graphics and thus we focus on the incision of M_S. To this end, we take the incision treatment as a linear segment which is intersected by M_S and a sweep surface generated by the virtual cutting tool. When the operation of incision happens, calculate the intersection points on M_S. There are two types of the intersection points: edge points and facet points. When the operation ends, subdivide the triangular elements and update topology change. Besides, the update of constraint on M_S is also needed.

As Figure 7 shows, there are three element types for mesh topology change. We use the method of [23] to subdivide the triangular mesh.

To update the constraint functions, the original constraints of the elements in the incision area are deleted. And, the new constraints of the newly subdivided triangular elements are generated. The new constraints generation is introduced in Figure 8.

For the stretch constraint, as Figure 8(a) shows, the intersection P is an edge point. Calculate the barycentric coordinate of P ($\lambda_1, \lambda_2, \lambda_3$) (in Figure 8(a), $\lambda_3 = 0$). The initial length of AP l_{AP} is

$$l_{\text{AP}} = \frac{\lambda_1}{\lambda_1 + \lambda_2} l_{\text{AB}}, \tag{19}$$

where l_{AB} is initial length of AB in the original constraint. For stretch constraint, as Figure 8(b) shows, the intersection P is the facet point. The barycentric coordinate of P is (λ_1, λ_2, λ_3). Take the calculation of initial length AP l_{AP} as an example. According to law of cosine, angle α is

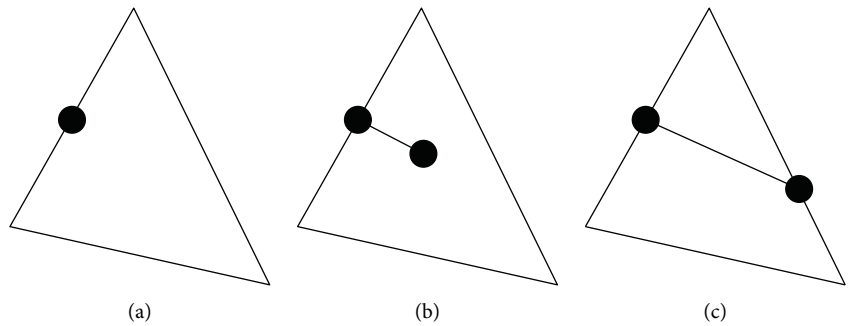


Figure 7: Three types of topology change: (a) uncut with an edge point; (b) partial cut; (c) full cut.

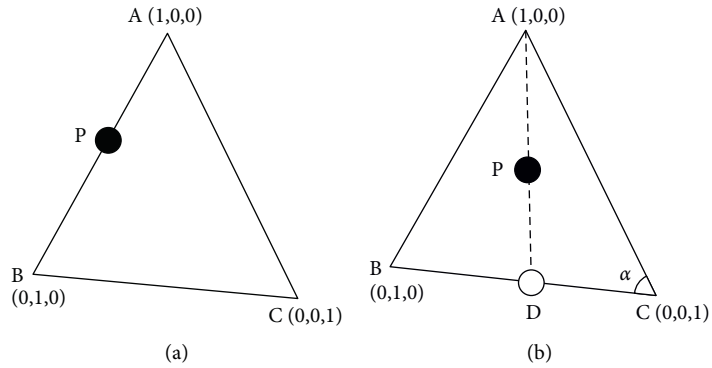


Figure 8: Generation of new stretch constraint: (a) intersection P on edge; (b) intersection P on facet.

$$\cos\alpha = \frac{l_{BC}^2 + l_{AC}^2 - l_{AB}^2}{2l_{BC}l_{AC}}, \tag{20}$$

where l_{AB}, l_{BC}, and l_{AC} are the initial lengths in original constraints AB, BC, and AC, respectively. The initial length of DC l_{DC} is

$$l_{DC} = \frac{\lambda_2}{\lambda_2 + \lambda_3} l_{BC}. \tag{21}$$

According to the law of cosine, the initial length of AD l_{AD} is

$$l_{AD} = \sqrt{l_{DC}^2 + l_{AC}^2 - 2l_{DC}l_{AC}\cos\alpha}. \tag{22}$$

The initial length of AP l_{AP} is

$$l_{AP} = (\lambda_2 + \lambda_3) l_{AD}. \tag{23}$$

For the bending constraint, the subdivision of triangle elements does not change the initial angle of two original triangles. Thus, the initial angle of triangular subelements remains unchanged. Since a control point is in the intersection segment, the control point P is duplicated due to the element subdivision. These two newly duplicated points are no longer control points.

8. Experiments and Results

8.1. Experimental Environment. We conduct our experiments on a computer using Intel Core i3 (2.7 GHz), 4 GB RAM, HD530 integrated graphics, and Windows 10. The software platform consists of VC++2019, open-source SOFA framework, and OpenGL graphics library.

8.2. Deformation Simulation. In order to present the mesh deformation effect, we implemented our proposed method on a simple deformation simulation.

In simulation, we use a pin-head to apply the external force $\mathbf{f}$ on the mesh. As Figure 9(a) shows, at the beginning, the mouse cursor controls the pin-head. When the cursor (pin-head) does not touch the mesh, the position of the pin-head is consistent with the position of the cursor. Once the cursor gets close to the node of mesh on the surface, as Figure 9(b) shows, the pin-head snaps to the closest node if the left mouse button is pressed down. In this situation, the position of pin-head is consistent with this node, not the cursor, and it becomes the external force-applied node. Then, as Figure 9(c) shows, move the cursor to generate external force $\mathbf{f}$.

$$\mathbf{f} = k(\mathbf{x} - \mathbf{x}_0). \tag{24}$$

$\mathbf{x}$ is the current position of the cursor. $\mathbf{x}_0$ is the original cursor position when the pin-head snaps to the mesh node. The coefficient k can be set by user's preference. The $\mathbf{f}$ is provided to the deformation algorithm. If the left mouse button is released, the external force is withdrawn and the position of pin-head is consistent with the cursor again. The cursor is not rendered in graphics. The flow chart is shown in Figure 10.

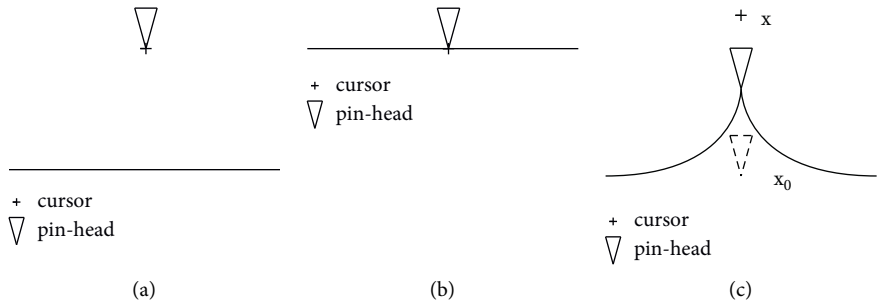


Figure 9: Diagram of deformation simulation: (a) pin-head untouched; (b) pin-head touched; (c) deformation.

Figure 11 shows the deformation effect on two shape models (cube and sphere) and two real mesh models (liver and steak) with tension and pressure. The two real mesh models are from turbosquid.com. We set the material properties as follows: Young's modulus: 5.5×10^3 Pa; Poisson's ratio: 0.3. By observing the four models, the surface deformation effects are realistic and the graphic rendering is good.

8.3. Surface Deformation Performance

8.3.1. Accuracy Verification. In order to show the accuracy of our proposed method, we compare the deformation result of liver and steak mesh simulated by our method with FEM [10], PBD [14], and MSM [17]. The mesh used in FEM is a fine tetrahedral volume mesh, which is directly generated by M_S through the Delaunay algorithm in Tetgen. The reason we use fine volume mesh in FEM is that the deformation experimental results will be used as the reference in experiment for its high accuracy. The mesh used in PBD and MSM is the same surface fine mesh model as the M_S in our hybrid mesh. The information of cube, sphere, liver, and steak mesh models used in different methods is listed in Table 1.

The specific process is as follows: First, we directly applied vertical upward tension and vertical downward pressure on the liver and steak mesh. The magnitudes are 0.5 N, 1.0 N, and 2.0 N for small deformation and 5.0 N, 10.0 N, and 20.0 N for large deformation. Second, we choose 10 points near the force-applied points of the mesh as sample points. Finally, the positions of the sample points after deformation of two shape models (cube and sphere) and two real mesh models (liver and steak) are recorded, and the deformation displacement d are calculated by using

$$d = \left|\mathbf{x}_s^i - \mathbf{x}_0^i\right|, \tag{25}$$

where $\mathbf{x}_s^i$ represents the position of the sample points i after the deformation and $\mathbf{x}_0^i$ represents the initial position of the sample points i. The deformation displacement radar diagrams of cube, sphere, liver, and steak models are shown in Figures 12–15 respectively.

Meanwhile, we calculate the RMSE (root mean square error) of the deformation displacement of our method, PBD, and MSM. Note that the RMSEs are obtained based on the FEM as reference:

$$r = \sqrt{\frac{\sum_{i=1}^{n} \left|\mathbf{x}_s^i - \mathbf{x}_f^i\right|^2}{n}}, \tag{26}$$

where r represents the RMSE value, n represents the number of the deformation sample points, $\mathbf{x}_s^i$ represents the position of the deformation sample points i after the deformation of our method, PBD, and MSM, respectively, and $\mathbf{x}_f^i$ represents the position of the deformation sample points i after the deformation of FEM as reference. For RMSE calculation, we choose 50 points nearest to the load-applied point (include the load-applied point) as sample points. Table 2 shows RMSE of deformation displacement of our method, PBD, and MSM on liver and steak mesh models. Table 3 shows RMSE of deformation displacement of our method, PBD, and MSM on cube and sphere shapes.

From the diagrams shown in Figures 12 and 13, for both small deformation (0.5 N, 1.0 N, 2.0 N) and large deformation (5.0 N, 10.0 N, 20.0 N), the deformation displacement curve of our method is closer to that of FEM than PBD and MSM. According to the definition of RMSE, the closer the RMSE value is to zero, the closer the deformation displacements is to the value of FEM. The RMSE value of our method is closer to zero than the value of PBD and MSM on both small deformation and large deformation. The above results verify the accuracy of our method.

8.3.2. Time Verification. For time verification, we compared the consumed time of our method with that of PBD, MSM, and FEM in calculating the deformation results. As shown in Tables 4 and 5, the time consumed by our method, PBD, MSM, and FEM under different forces is recorded on two shape models (cube and sphere) and two real mesh models.

From the result shown in Tables 3 and 4, on the one hand, compared with PBD and MSM, our method costs more computation time due to the addition of the time consumed for M_V deformation, but this time increment is not too much. On the other hand, compared with FEM, our method reduces large computation cost, since we use coarse volume mesh instead of fine volume mesh in FEM for deformation calculation. Although the study in [18] has reduced some computational cost of FEM by optimizing the numerical schemes of dealing with the stretch part of deformation energy, it needs more elements to present the accurate surface deformation effect than ours. Our proposed

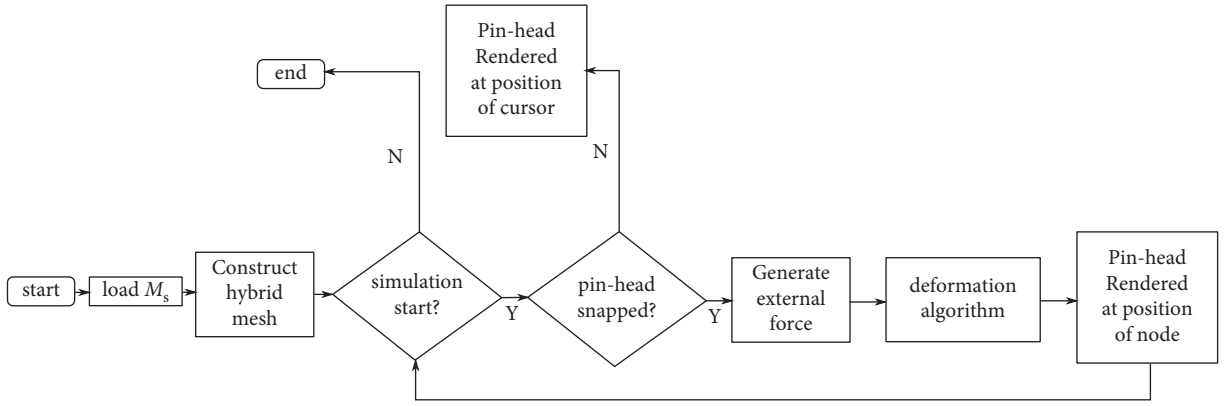


Figure 10: Flowchart of the simple deformation simulation.

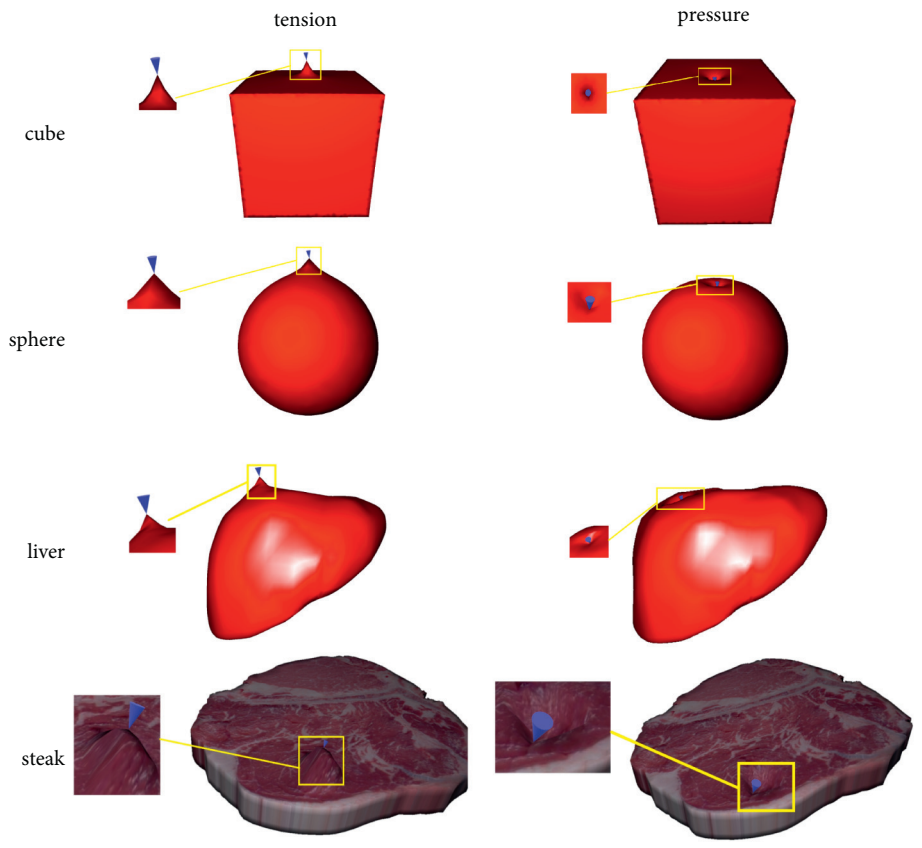


Figure 11: The surface deformation effects of four models in two load cases. Models (from top to bottom): cube, sphere, liver, and steak. Load cases: left, tension; right, pressure.

Table 1: The information of four models.

Models	Fine surface mesh (M_S)			Coarse volume mesh (M_V)			Fine volume mesh		
	#node	#elem	Size (kB)	#node	#elem	Size (kB)	#node	#elem	Size (kB)
Cube	1291	2888	106	436	1220	45	3028	14874	465
Sphere	1402	3158	115	587	1631	60	3441	17821	538
Liver	1379	3122	114	543	1592	58	3357	16985	526
Steak	1543	3378	126	654	1719	65	3865	20898	623

#node: number of nodes; #elem: number of elements; size is expressed in kB. Our method: $M_S + M_V$; PBD and MSM: M_S; FEM: fine volume mesh generated by M_S.

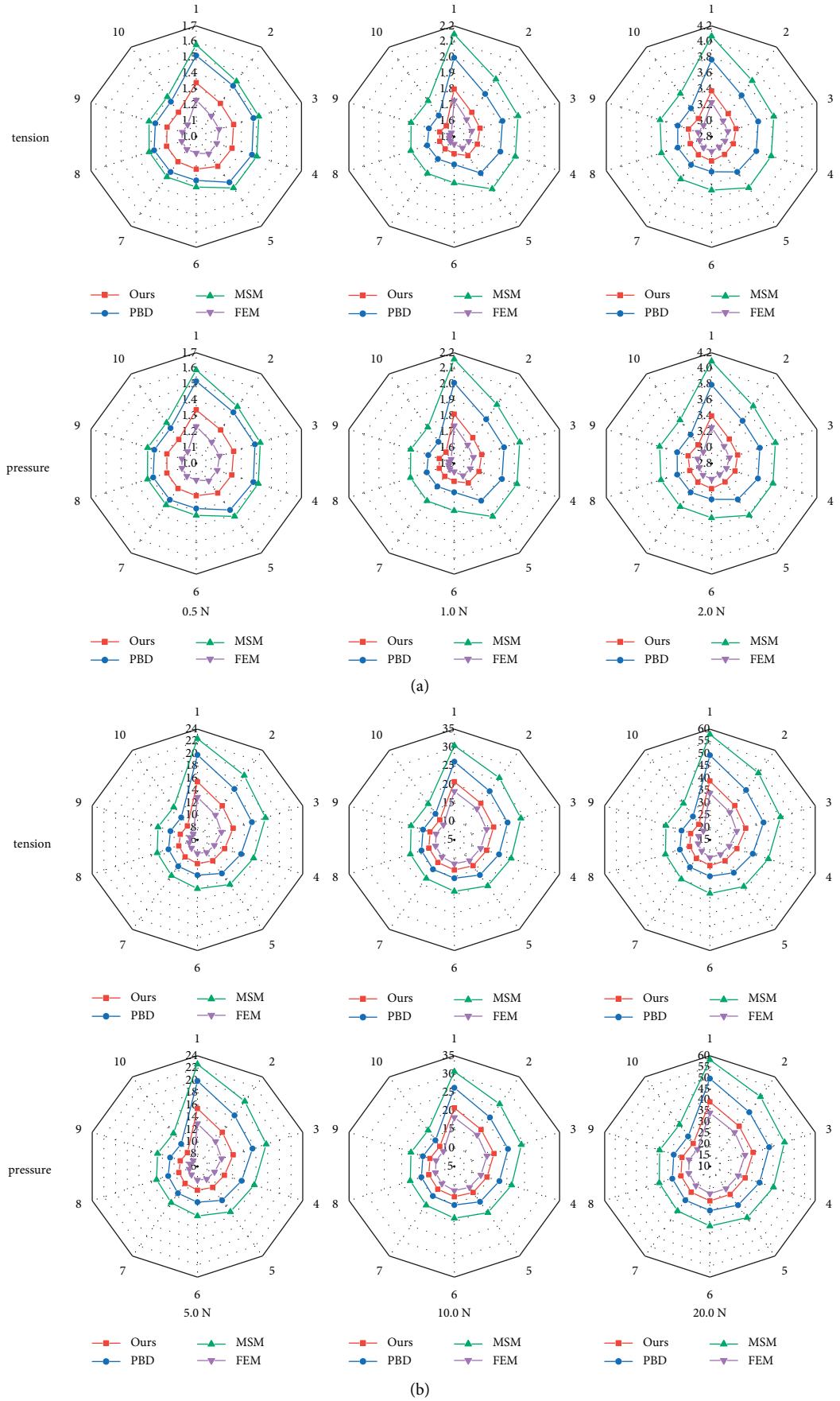


Figure 12: The surface deformation displacement with respect to the sample nodes of the cube model. (a) Small deformation: from left to right are 0.5 N, 1.0 N, and 2.0 N. Top indicates tension and bottom indicates pressure. (b) Large deformation: from left to right are 5.0 N, 10.0 N, and 20.0 N. Top indicates tension and bottom indicates pressure.

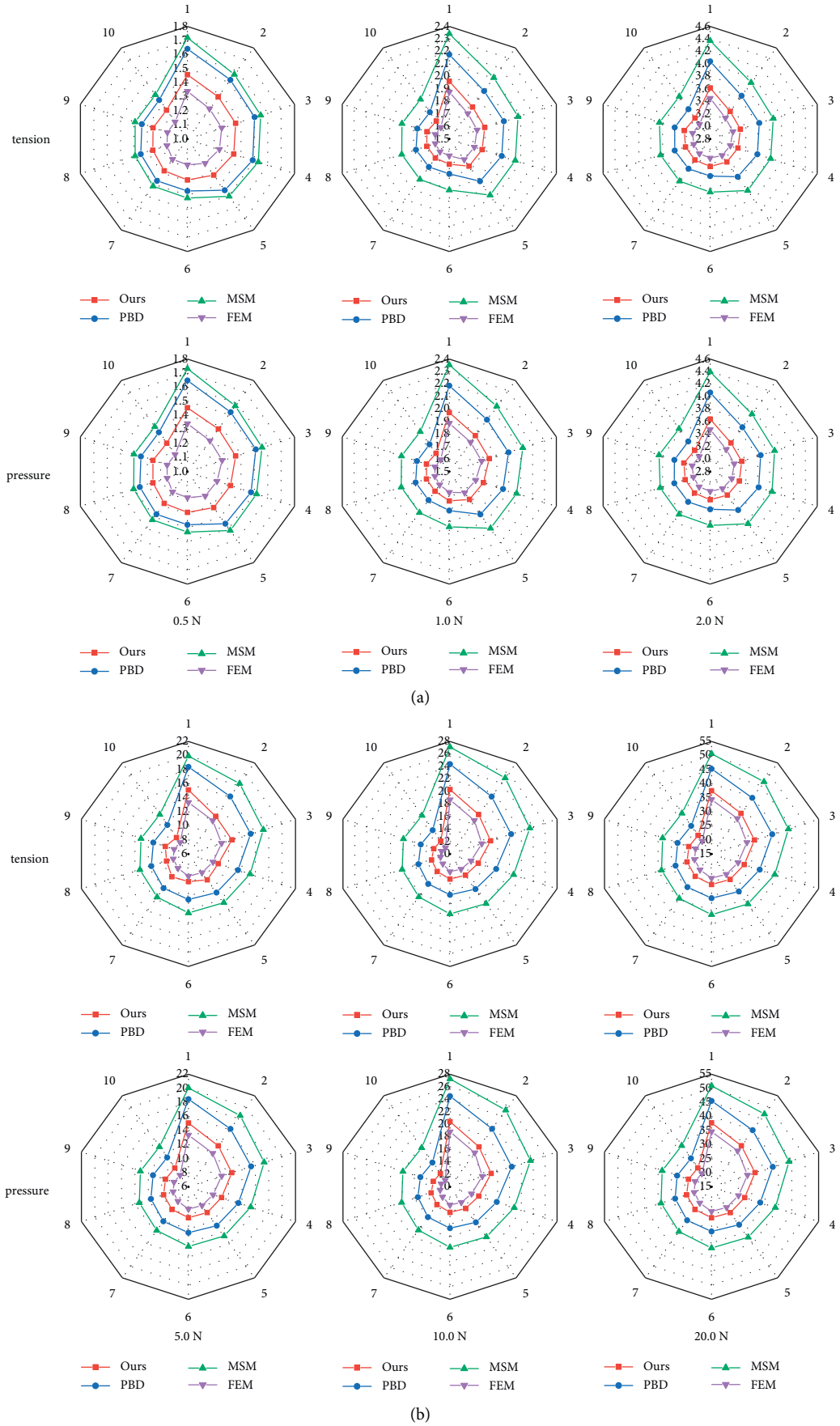
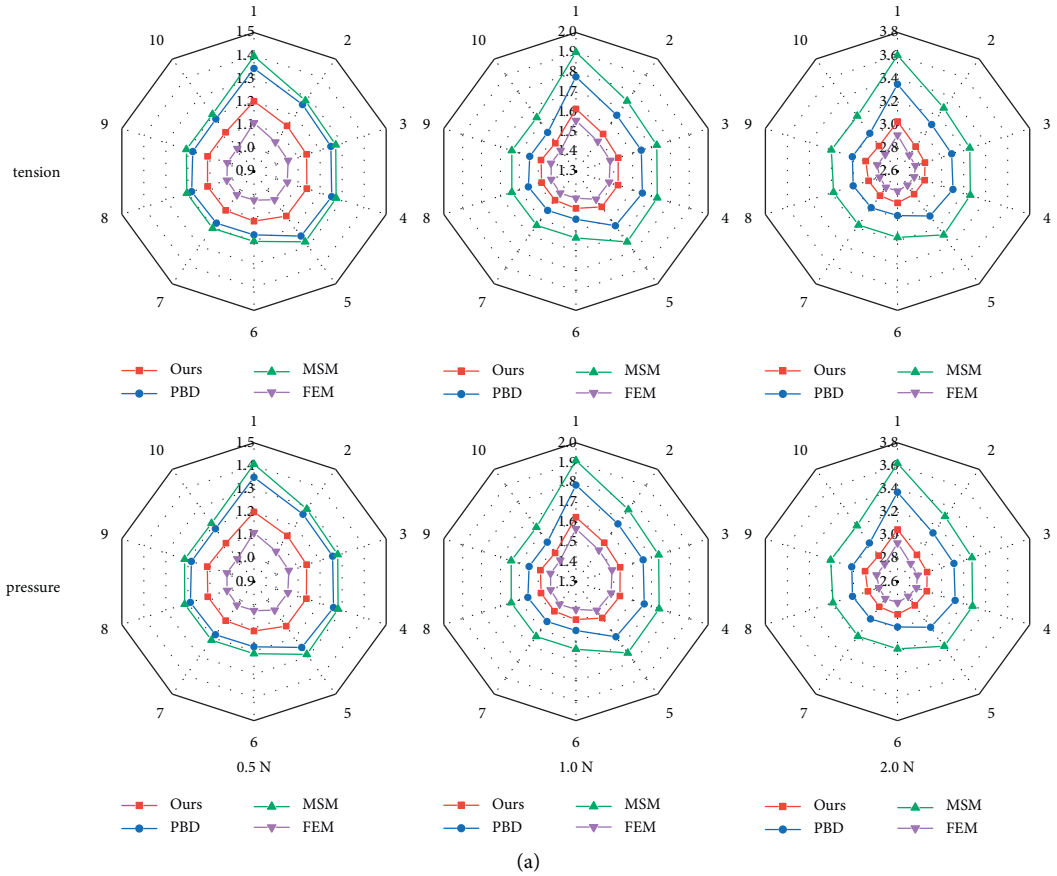
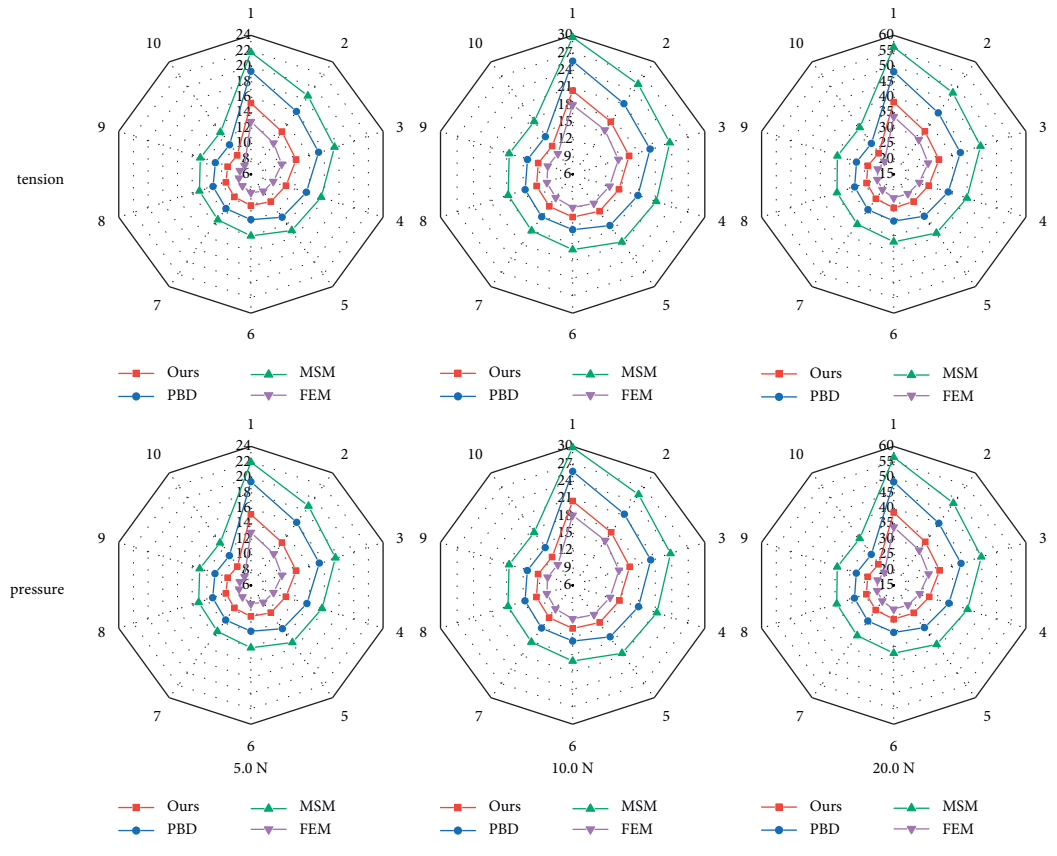


Figure 13: The surface deformation displacement with respect to the sample nodes of the sphere model. (a) Small deformation: from left to right are 0.5 N, 1.0 N, and 2.0 N. Top indicates tension and bottom indicates pressure. (b) Large deformation: from left to right are 5.0 N, 10.0 N, and 20.0 N. Top indicates tension and bottom indicates pressure.

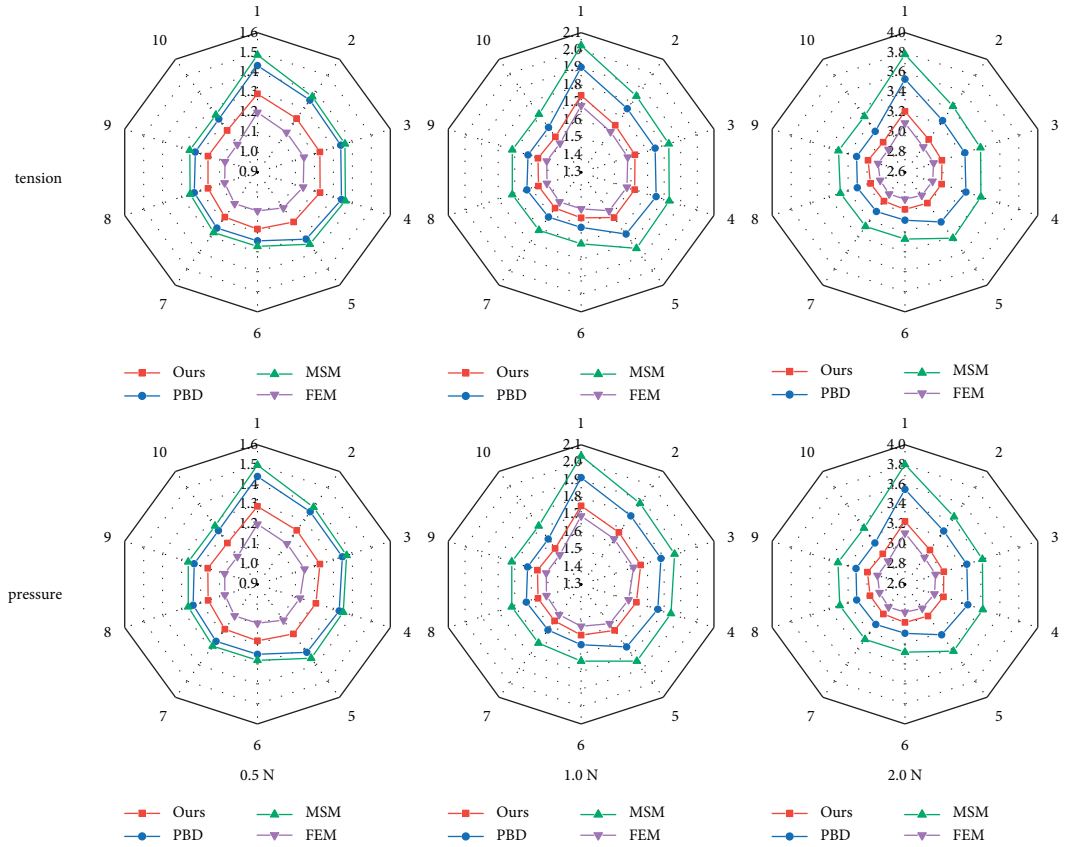


(a)

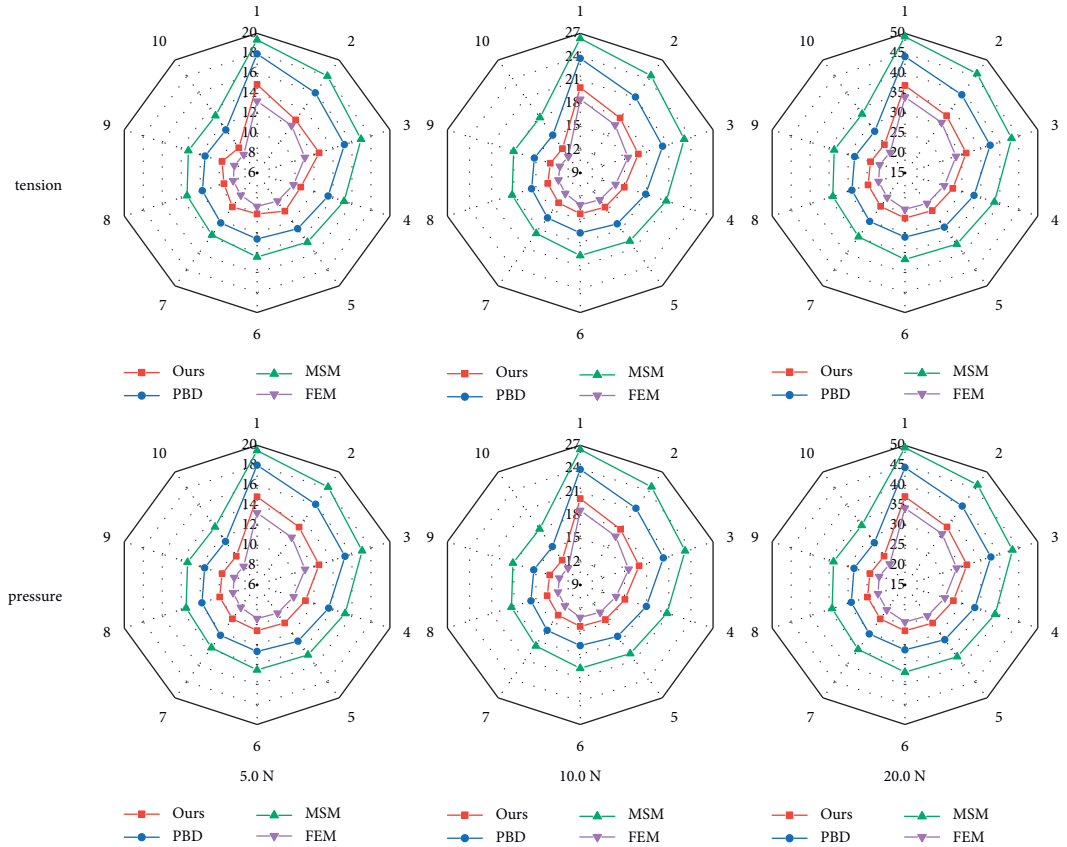


(b)

Figure 14: The surface deformation displacement with respect to the sample nodes of the liver model. (a) Small deformation: from left to right are 0.5 N, 1.0 N, and 2.0 N. Top indicates tension and bottom indicates pressure. (b) Large deformation: from left to right are in 5.0 N, 10.0 N, and 20.0 N. Top indicates tension and bottom indicates pressure.



(a)



(b)

FIGURE 15: The surface deformation displacement with respect to the sample nodes of the steak model. (a) Small deformation: from left to right are 0.5 N, 1.0 N, and 2.0 N. Top indicates tension and bottom indicates pressure. (b) Large deformation: from left to right are 5.0 N, 10.0 N, and 20.0 N. Top indicates tension and bottom indicates pressure.

Table 2: RMSE of deformation displacement of our method, PBD, and MSM on two real mesh models.

RMSE (mm)		Liver			Steak		
		Ours	PBD	MSM	Ours	PBD	MSM
Tension	0.5 N	0.0850	0.1784	0.2067	0.0878	0.1801	0.2080
	1.0 N	0.0476	0.1366	0.2271	0.0487	0.1450	0.2388
	2.0 N	0.0952	0.2732	0.4542	0.0974	0.2899	0.4776
	5.0 N	0.2114	0.8580	1.0708	0.2595	0.8881	1.0245
	10.0 N	0.5288	1.8045	2.2797	0.5771	1.7984	2.3723
	20.0 N	1.0220	4.0798	5.5188	1.0382	4.0898	5.4101
Pressure	0.5 N	0.0476	0.1366	0.2271	0.0854	0.1845	0.2156
	1.0 N	0.0572	0.1740	0.2991	0.0511	0.1650	0.2660
	2.0 N	0.0952	0.2732	0.4542	0.0972	0.3299	0.4920
	5.0 N	0.2887	0.8876	1.0690	0.2575	0.8765	1.0077
	10.0 N	0.5347	1.7992	2.3037	0.5817	1.8091	2.2516
	20.0 N	1.0340	3.9092	5.2667	1.0342	3.9090	5.3660

Table 3: RMSE of deformation displacement of our method, PBD, and MSM on two shape models.

RMSE (mm)		Cube			Sphere		
		Ours	PBD	MSM	Ours	PBD	MSM
Tension	0.5 N	0.0295	0.1004	0.1377	0.0302	0.1021	0.1455
	1.0 N	0.0462	0.1613	0.2061	0.0493	0.1432	0.2474
	2.0 N	0.0999	0.2948	0.4819	0.0957	0.2618	0.511
	5.0 N	0.2224	0.8614	0.9809	0.2266	0.8437	0.9364
	10.0 N	0.569	1.8081	2.4256	0.5355	1.6941	2.1825
	20.0 N	0.9545	4.1777	5.8886	1.0372	4.3188	5.4804
Pressure	0.5 N	0.0305	0.1015	0.1387	0.0281	0.1163	0.1327
	1.0 N	0.0472	0.1702	0.2119	0.0462	0.1811	0.2283
	2.0 N	0.0962	0.2967	0.4922	0.1009	0.3099	0.5087
	5.0 N	0.2161	0.8842	0.9974	0.2207	0.8775	1.0279
	10.0 N	0.5734	1.8164	2.5253	0.5817	1.8272	2.5137
	20.0 N	1.0702	4.0241	5.6092	1.1164	4.0482	5.7581

Table 4: Deformation calculation time consumption of our method, PBD, MSM, and FEM on two shape models.

Time (ms)		Cube				Sphere			
		Ours	PBD	MSM	FEM	Ours	PBD	MSM	FEM
Tension	0.5 N	3.8	2.4	1.9	11.6	3.9	2.6	2.2	10.9
	1.0 N	3.9	2.2	2.0	12.5	3.8	2.1	2.2	11.8
	2.0 N	3.6	2.3	2.2	12.6	3.9	2.1	2.3	12.1
	5.0 N	3.7	2.3	1.9	13.6	3.8	2.3	2.4	13.5
	10.0 N	3.8	2.4	2.2	13.0	3.7	2.5	2.1	13.3
	20.0 N	4.0	2.6	2.2	13.7	3.9	2.3	2.1	13.9
Pressure	0.5 N	3.6	2.3	2.1	11.1	3.7	2.1	2.4	11.1
	1.0 N	3.7	2.2	2.3	12.6	3.8	2.3	2.2	12.1
	2.0 N	3.8	2.3	2.1	13.3	3.9	2.3	2.4	12.7
	5.0 N	3.9	2.5	2.4	13.3	3.7	2.7	2.4	13.1
	10.0 N	3.9	2.5	2.3	14.4	3.8	2.5	2.2	13.3
	20.0 N	4.0	2.6	2.2	14.9	4.0	2.3	2.0	13.2

method needs fewer parameters of volume element and surface element to present the effect. The real-time performance of our proposed method is guaranteed.

8.3.3. Discussion of the Trade-Off between Accuracy and Time Consumption. In accuracy verification, by comparing the deformation displacement and RMSE of our method with the ones of PBD, MSM, and FEM, the results indicates that the deformation of our proposed method is similar to the one of FEM, which we take as reference to evaluate the accuracy. The deformation result of our method is more accurate, due to the constraint on M_S from the displacement of M_V. For the computation time, by comparing the consumed time of our methods with that of PBD, MSM, and FEM, the result indicates that our method has lower time consumption than FEM. Although our method has more time consumption than PBD and MSM, the time increment

Table 5: Deformation calculation time consumption of our method, PBD, MSM, and FEM on two real mesh models.

Time (ms)		Liver				Steak			
		Ours	PBD	MSM	FEM	Ours	PBD	MSM	FEM
Tension	0.5 N	3.8	2.2	2.1	8.2	3.6	2.4	2.3	16.8
	1.0 N	3.7	2.3	2.1	9.3	3.6	2.2	2.2	17.2
	2.0 N	3.9	2.2	2.2	10.3	3.8	2.3	2.3	17.9
	5.0 N	3.6	2.1	2.1	12.5	3.5	2.3	2.2	19.6
	10.0 N	3.8	2.3	2.1	12.1	3.7	2.4	2.1	20.7
	20.0 N	3.7	2.4	2.3	13.8	3.9	2.2	2.1	22.8
Pressure	0.5 N	3.7	2.2	2.1	7.9	3.6	2.3	2.3	18.4
	1.0 N	3.6	2.3	2.2	9.1	3.7	2.4	2.2	18.8
	2.0 N	3.8	2.2	2.1	9.9	3.9	2.5	2.3	19.3
	5.0 N	3.9	2.4	2.2	11.4	3.7	2.6	2.3	20.1
	10.0 N	3.6	2.3	2.1	12.2	3.8	2.3	2.1	21.4
	20.0 N	3.8	2.4	2.1	13.1	3.7	2.5	2.2	23.4

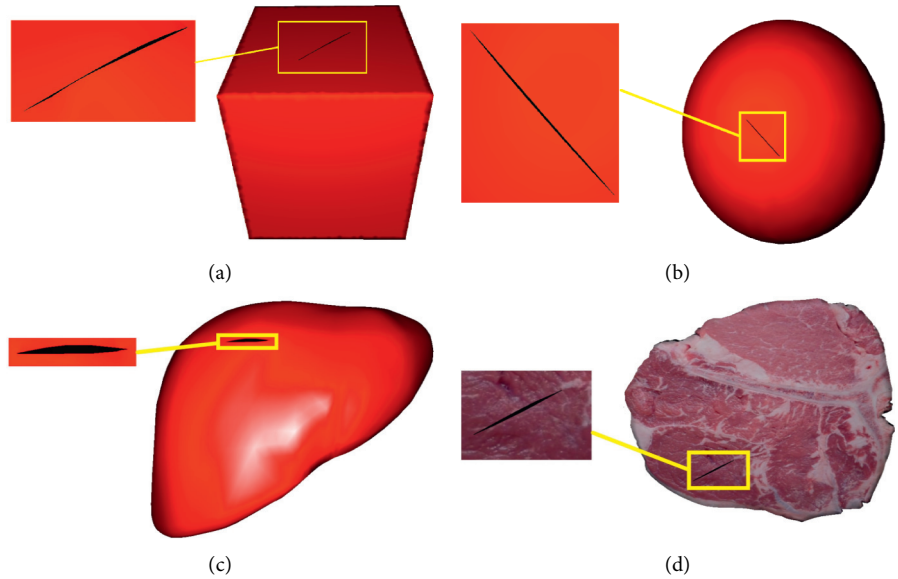


Figure 16: The small incision effects of two shape models and two real mesh models. (a) Cube, (b) sphere, (c) liver, and (d) steak.

is not too much since computing the deformation for M_V does not take too much time. Comprehensively considering the deformation result and time consumption, our proposed method achieves the trade-off between the high accuracy and low time consumption.

8.4. Small Incision Effect. We also developed a small incision simulation using our hybrid mesh model. We use a pin-head as a virtual cutting tool. If the pin-head is close to the mesh surface and the left mouse button is pressed down, the incision operation begins. As the pin-head moves, the intersection happens on the mesh surface. When the left mouse button is released, the incision operation is finished. Figure 16 shows the small incision simulation effects on the liver and steak mesh model. For the incision to be visible, we applied a tiny force on a node near the incision on the surface mesh. From Figure 16, we can see that the incisions of these two shape models and two real mesh models are smooth. The incision processing is compatible with our proposed method.

9. Conclusions

In this paper, we propose a deformation method based on the hybrid mesh model which achieves the trade-off between the high accuracy and low time consumption. First, we proposed the construction of the hybrid mesh from fine surface triangular mesh. Its most important step is the simplification process of the fine surface mesh. Second, we combined the FEM for coarse volume mesh deformation and PBD deformation method for fine surface mesh deformation and proposed the algorithm flow based on the hybrid mesh model. Third, we implemented our method on a simple deformation simulation. The experimental result of the deformation algorithm is presented. Compared with the PBD, MSM, and FEM, our method verifies the high accuracy and low time consumption. Finally, we developed an incision simulation which is compatible with our proposed method.

In the future, our proposed method can be improved further in discontinuity processing. The cutting on the hybrid mesh model is a major problem. It involves the

subdivision of M_V, the way to calculate the deformation of the subdivided M_V, and how the surface node is constrained on the subdivided M_V. When a large cut happens, the graphic rendering of incision on M_V is a problem. It is a challenging task to maintain the trade-off between high accuracy and low time consumption in deformation simulation with subdivision of M_S and M_V.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the NSFC Joint Fund under Grants GG2090090072, U1332130, and U1713206; 111 Projects under Grant B07033; 973 Project under Grant 2014CB931804; and Key Research and Development Plan of Anhui Province under Grant 1704a0902051.

References

[1] J. Wu, R. . Westermann, and C. Dick, "A survey of physically based simulation of cuts in deformable bodies," *Computer Graphics Forum*, vol. 34, no. 6, pp. 161–187, 2015.

[2] D. Thanoon, M. Garbey, and B. L. Bass, "Deriving indicators for breast conserving surgery using finite element analysis," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 5, no. 18, pp. 533–544, 2015.

[3] Z. Liao, L. Kong, Z. Peng, and X. Huang, "New tetrahedron cutting method on tetrahedron subdivision," *Application Research of Computers*, vol. 12, pp. 3834–3836, 2015.

[4] V. B. Shim, M. Battley, I. A. Anderson, and J. T. Munro, "Validation of an efficient method of assigning material properties in finite element analysis of pelvic bone," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 18, no. 14, pp. 1495–1499, 2015.

[5] J. Zhen, Y. Yang, S. Lou, D. Zhang, and S. Liao, "Construction and validation of a three-dimensional finite element model of degenerative scoliosis," *Journal of Orthopaedic Surgery and Research*, vol. 10, no. 189, 2015.

[6] C. J. Paulus, L. Untereiner, H. Courtecuisse, S. Cotin, and D. Cazier, "Virtual cutting of deformable objects based on efficient topological operations," *The Visual Computer*, vol. 31, no. 6-8, pp. 831–841, 2015.

[7] H. Courtecuisse, J. Allard, P. Kerfriden, S. Bordas, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, no. 2, pp. 394–410, 2013.

[8] N. Haouchine, S. Cotin, I. Peterlik, J. Dequidt, and M. S. Lopez, "Impact of soft tissue heterogeneity on augmented reality for liver surgery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 5, pp. 584–597, 2015.

[9] W. Tang and T. R. Wan, "Constraint-based soft tissue simulation for virtual surgical training," *IEEE Transactions on Bio-Medical Engineering*, vol. 61, no. 11, pp. 2698–2706, 2014.

[10] T. Kugelstadt, D. Koschier, and J. Bender, "Fast corotated FEM using operator splitting," *Computer Graphics Forum*, vol. 37, no. 8, pp. 149–160, 2018.

[11] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin, "A survey on position-based simulation methods in computer graphics," *Computer Graphics Forum*, vol. 33, no. 6, pp. 228–251, 2014.

[12] B. Kubiak, N. Pietroni, F. Ganovelli, and M. Fratarcangeli, "A robust method for real-time thread simulation," in *Proceedings of the 2007 ACM symposium on Virtual reality software and technology. VRST*, Newport Beach, CA, USA, November 2007.

[13] 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.

[14] M. Camara, E. Mayer, A. Darzi, and P. Pratt, "Soft tissue deformation for surgical simulation: a position-based dynamics approach," *International Journal of Computer Assisted Radiology & Surgery*, vol. 11, no. 6, pp. 919–928, 2016.

[15] J. Pan, J. Bai, Z. Xin, A. Hao, and Q. Hong, "Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics," *Computer Animations and Virtual Worlds*, vol. 26, no. 3-4, pp. 321–335, 2015.

[16] Y. Liu, C. Guan, and J. Li, "Xueli yu and shuang zhang "the PBD model based simulation for soft tissue deformation in virtual surgery"," *Journal of Physics: Conference Series*, vol. 1621, Article ID 012043, 2020.

[17] Y. Duan, W. Huang, H. Chang et al., "Volume preserved mass–spring model with novel constraints for soft tissue deformation," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 268–280, 2016.

[18] S. Skornitzke, G. Schummers, M. Schreckenberg et al., "Mass-spring systems for simulating mitral valve repair using 3D ultrasound images," *Computerized Medical Imaging and Graphics*, vol. 45, pp. 26–35, 2015.

[19] W. Gao, L. Chu, Y. Fu, and S. Wang, "A non-linear, anisotropic mass spring model-based simulation for soft tissue deformation," in *Proceedings of the 11th lnternational Conference on Ubiquitous Robots and Ambient lntelligence*, pp. 7–10, Kuala Lumpur, Malaysia, November 2014.

[20] X. Zhang, J. Duan, W. Sun, T. Xu, and S. K. Jha, "A three-stage cutting simulation system based on mass-spring model," *CMES-Computer Modeling in Engineering & Sciences*, vol. 127, no. 1, pp. 117–133, 2021.

[21] H. Si, "Tetgen, a delaunay-based quality tetrahedral mesh generator," *ACM Transactions on Mathematical Software*, vol. 41, no. 2, pp. 1–36, 2015.

[22] C. H. Papadimitriou, "NP-complete problems," in *Computational Complexity*, pp. 181–207, Addison-Wesley, Boston USA, 1st edition, 1993.

[23] P.-L. Manteaux, W.-L. Sun, F. Faure, M.-P. Cani, and J. F . O'Brien, "Interactive detailed cutting of thin sheets," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pp. 125–132, Paris, France, November 2015.