Vis Comput
DOI 10.1007/s00371-016-1317-x

ORIGINAL ARTICLE

Real-time dissection of organs via hybrid coupling of geometric metaballs and physics-centric mesh-free method

Junjun Pan[1] · **Shizeng Yan**[1] · **Hong Qin**[2] · **Aimin Hao**[1]

Abstract This paper systematically describes a real-time dissection approach for digital organs by strong coupling of geometric metaballs and physically correct mesh-free method. For organ geometry, we employ a novel hybrid model comprising both inner metaballs and high-resolution surface mesh with texture information. Through the use of metaballs, the organ interior is geometrically simplified via a set of overlapping spheres with different radii. As for digital organ's physical representation, we systematically articulate a hybrid framework to interlink metaballs with physics-driven mesh-free method based on moving least squares (MLS) shape functions. MLS approach enables the direct and rapid transition from metaball geometry to local nodal formulations, which afford potential-energy-correct physical modeling and simulation over continuum domain with physical accuracy. For soft tissue dissection, the nature of our MLS-driven mesh-free method also facilitates adaptive topology modification and cutting surface reconstruction. To expedite simulation towards real-time performance, at the numerical level, we resort to position-based dynamics to simplify physical deformation to drive metaballs participating in the mesh-free formulation. Since nodal points participating in the physical process exist temporarily only in localized regions adjacent to the cutting path, our method could warrant accurate cutting surface without sacrificing real-time computational efficiency. This hybrid dissection technique has already been integrated into a VR-based laparoscopic surgery simulator with a haptic interface.

Keywords Metaballs · Mesh-free method · Digital organ · Physics-based deformation · Dissection

1 Introduction

In recent years, with the significant progresses of virtual reality technologies in medical fields, virtual surgery and VR-based medical simulators have attracted increasing interests from both academic researchers and industrial practitioners [1,2]. The popular research topics involve physical deformation of organs and simulation of the basic surgical procedures, such as dissection. For deformation and cutting simulation of soft tissue, a variety of approaches have been proposed. Technically speaking, they can be roughly classified into the mass-spring system, finite-element method (FEM), mesh-free method, etc. However, they all have their own limitations, such as low physical accuracy, and high computational cost leading to the real-time requirement loss [3]. Metaballs, as a special geometric modeling method based on implicit surfaces, can effectively express continuous, blob-like surfaces. Their modeling advantages for organs' shape geometry have been fully exploited since its inception [4,5].

In this paper, we propose a real-time-cutting simulation approach for digital organs by marrying the geometry of metaballs with the physically accurate mesh-free method. For organ geometry, we employ a novel hybrid model comprising both inner metaballs and fine surface mesh with texture.

Electronic supplementary material The online version of this article (doi:10.1007/s00371-016-1317-x) contains supplementary material, which is available to authorized users.

✉ Junjun Pan
pan_junjun@hotmail.com; junjunpan@gmail.com

Hong Qin
qin@cs.stonybrook.edu

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China

[2] Department of Computer Science, Stony Brook University (SUNY Stony Brook), Stony Brook, USA

Published online: 02 September 2016

Using metaballs, the organ interior can be approximated by a set of coarse, overlapping spheres with different radii. Moreover, if metaballs are equipped with any type of physical properties, the metaballs could also participate in the deformation of organ enabled by position-based dynamics (PBD) globally. For organs' physical representation in this paper, the mesh-free method is an ideal and natural choice, because other formulations are either less accurate (e.g., mass-spring system) or difficult to simulate due to complicated mesh modification (e.g., FEM), especially when our underlying geometry is solely dictated by metaballs serving as organs' interior structure. Therefore, we systematically articulate a hybrid framework to interlink metaballs with physics-driven mesh-free approach based on moving least squares (MLS) shape functions. It enables the direct, natural, and rapid transition from metaballs to local nodal formulations, which afford potential-energy-correct physical modeling and simulation over continuum domain with physical accuracy. For the coupling of local nodal points and metaballs nearby, we utilize the stretching potential energy and displacement to transfer the motion between these two different systems. In particular, a straightforward synchronization strategy is designed to handle the different convergence rates. For soft tissue dissection, the nature of our MLS-driven mesh-free method supports adaptive topology modification and cutting surface reconstruction. Once dissection is finished, the separated nodal points are converted back into the new overlapping spheres by the scheme of Centroid Voronoi Diagram. In addition, the cutting surface is reconstructed from the down-sampled nodal points by Constraint Delaunay Triangulation. Finally, undergoing the cutting behavior analysis through a mechanical model, we integrate this hybrid dissection simulation with a haptic device. This prototype system has already been adapted to a VR-based laparoscopic surgery simulator. Specifically, this paper includes the following innovative contributions:

- We systematically present a hybrid framework to interlink metaballs with a MLS-driven mesh-free method for cutting simulation. It enables the direct and rapid transition from metaballs to local nodal formulations, which afford potential-energy-correct physical modeling and simulation over continuum domain with high physical reality.
- At the numerical level, to tightly couple metaballs' motion with the mesh-free formulation, we propose to utilize PBD to approximate the entire physical deformation of digital organs. Another motivation for the utility of PBD is to expedite physics-driven simulation towards real-time performance. Moreover, nodal points only exist locally in the vicinity of the cutting path temporarily, our method provides an accurate cutting surface without sacrificing real-time computational efficiency.
- Using a mechanical model for cutting behavior analysis, we integrate this hybrid dissection simulation with a haptic device. At the system level, this novel cutting method is adapted to a VR-based laparoscopic surgery simulator.

2 Related work

Deformation and dissection of soft tissue are essential tasks in surgery. They require real-time geometry and topology modification, which form the most complex problems in virtual surgery. Despite earlier research endeavors, realistic and interactive dissection simulation remains a challenging topic and continues to attract a great deal of research efforts in recent years [3]. In principle, deformation and cutting simulation methods can be classified into three different categories.

Finite-element methods (FEM) FEM can perform accurate deformation result but with high computation cost. Wu et al. [7] presented a composite FEM simulation method for deformable objects. Cuetoet et al. [8] gave a survey of the latest progress of FEM research and their application in virtual surgery. In general, FEM-based dissection faces the challenges in realistic cutting surface reconstruction and dynamic topological update [9]. FEM subdivision-based method [3] is usually employed to cope with the topological update, but it can easily generate ill-conditioned elements.

Mass-spring-based methods This type of methods simplify the physical structure of objects as mass points and linked springs. It can achieve high computation efficiency, but with the sacrifice of physical accuracy. Pan et al. [10] presented a multi-layer mass-spring model to simulate the membrane tissue dissection in laparoscopic intestine surgery. Liu et al. [11] proposed a scheme of time integration for the standard mass-spring systems based on block coordinate descent. It provides a fast solution for classical linear springs.

Mesh-free methods Mesh-free methods are also called meshless methods, in which the physical behaviors are driven by material distance-based radial functions [12]. These approaches do not require any explicit topological construction. Thus, they can offer great flexibility in accommodating topological changes when confronting cutting simulation. Jones et al. [13] presented a mesh-free method to deform elastoplastic objects. Steinemann et al. [14] proposed a splitting method using a meshless discretization of the deformation field. Pietroni et al. [15] presented the splitting simulation based on a mesh-free technique, to handle the interactive virtual cutting on deformable objects. Pietroni et al. [16] presented a meshless animation framework for fracturing elastic and plastic solids. Adams and Wicke [17] comprehensively discussed the meshless methods for physics-based modeling and simulation. However, mesh-free methods exist some limitations. When the global arbitrary cutting of organs happens, its computation cost will drastically increase due to the time-consuming modification of the global material distance field.

Position-based dynamics (PBD) is another widely used method in physical deformation, due to its fast, robustness and position-based manipulation feature [18]. Pan et al. [19] presented an interactive dissection approach for hybrid soft tissue models, which contain both surface mesh and tetrahedra, by extended position-based dynamics. Müller et al. [20] applied the PBD framework to simulate soft objects in real-time environment.

Quite different from the above physical modeling methods, most existing metaballs-based approaches focus on the geometric modeling using implicit surface [4,21,22]. Pan et al. [6] presented a deformation approach for organs using metaballs. Its novelty hinges upon the integration of metaballs and PBD which enables metaballs-based organs to deform dynamically. Based on their research, our work is to interlink metaballs with a physically accurate mesh-free method, and present a hybrid framework for dissection simulation.

3 Method overview

We first present the framework overview, as shown in Fig. 1.

Hybrid geometric model A hybrid geometric model, including both surface mesh and the metaballs, is employed to represent the digital organs. The finer surface mesh with high-resolution geometric information and texture represents the boundary structure of organs. Through the use of metaballs, the organ interior is geometrically simplified via a set of coarse, overlapping spheres with different radii.

Global deformation Position-based dynamics is utilized to compute the position of sphere centers for global organ deformation. To preserve the local detail of the metaballs, we add the Laplacian coordinates constraints to update the position of sphere centers. Then, the local volume preservation is employed to adjust the radii of spheres. Finally, an automatic skinning algorithm is presented to map the deformation to the surface mesh in real time.

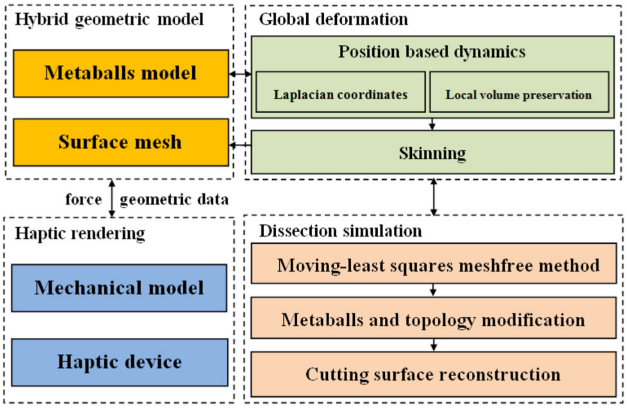


Fig. 1 Framework of deformation and dissection simulation via hybrid coupling of metaballs and mesh-free method

Dissection simulation Once the scalpel is touched with a metaball and starts cutting, the dissected spheres will be removed and converted into nodal points. The movements of these points are driven by moving least squares mesh-free method and coupled with the deformation of spheres nearby. When the scalpel leaves the soft tissue, these points will be separated and converted back into spheres with updated topology connection.

Haptic rendering Through the cutting behavior analysis by a mechanic model, the haptic interface is integrated in the simulation to feedback the force.

4 Global deformation of organs

In our technique, the geometric model of organs consists of both an exterior surface mesh and interior metaballs. Therefore, the first task is to fill an arbitrary object densely with a set of overlapping spheres and make the shape of metaballs best match the boundary of the mesh. As the preliminary research work for this paper, the technical detail of this section can be found in [6].

4.1 Construction of metaballs model

We use the Sphere Tree Construction Toolkit [23] to pack the spheres in the triangular mesh of the organ initially. The key strategy of this toolkit is to find the medial surface of an object using the Voronoi Diagram and pack spheres from the medial surface to approximate the objects roughly. Then, a series of optimization strategies are utilized to adjust the sphere radii, fill the vacant space, and merge the extremely overlapping spheres. Finally, a global electrostatic attraction model [6] is proposed to drive the metaballs best matching the boundary of the organ mesh. Figure 2b illustrates the metaballs construction result for a liver model.

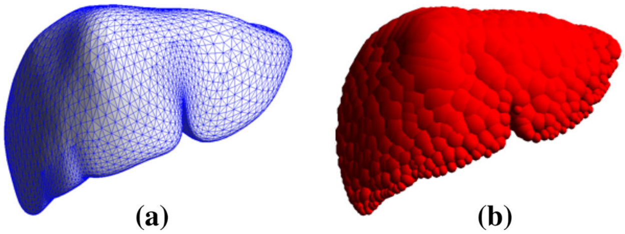


Fig. 2 Metaballs construction for a liver model. **a** Triangular mesh of liver, **b** metaballs model of liver

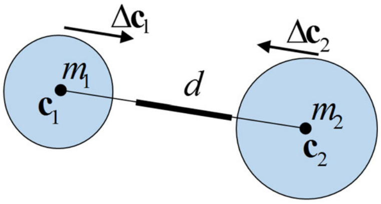


Fig. 3 Illustration of stretching constraint for two connected spheres

4.2 Deformation of metaballs

After the construction of metaballs topology, the position-based dynamics (PBD) is employed to compute the position of sphere centers. In general, PBD involves stretching constraints, bending constraints, and volume constraints [18]. Considering the irregular topology of the metaballs model, we cannot treat it as a polygon or polyhedron mesh. Here, we only apply stretching constraints for the deformation of metaballs. Figure 3 gives an example of stretching constraint for two connected spheres.

The constraint function about stretch can be described as follows:

$$C_{\text{stretching}}(\mathbf{c_1}, \mathbf{c_2}) = |\mathbf{c_1} - \mathbf{c_2}| - d, \tag{1}$$

where d is the initial distance between the sphere centers $\mathbf{c_1}$ and $\mathbf{c_2}$. m_1 and m_2 indicate the mass of spheres 1 and 2. Then, the displacement in each iteration can be computed by the following formula:

$$\Delta\mathbf{c_1} = -\frac{m_2}{m_1 + m_2}(|\mathbf{c_1} - \mathbf{c_2}| - d)\frac{\mathbf{c_1} - \mathbf{c_2}}{|\mathbf{c_1} - \mathbf{c_2}|}, \tag{2}$$

and

$$\Delta\mathbf{c_2} = \frac{m_1}{m_1 + m_2}(|\mathbf{c_1} - \mathbf{c_2}| - d)\frac{\mathbf{c_1} - \mathbf{c_2}}{|\mathbf{c_1} - \mathbf{c_2}|}. \tag{3}$$

Nevertheless, the stretching constraints in PBD only make 2D constraints and lack of 3D constraints. Therefore, we apply the Laplacian coordinates constraints [24] to preserve the local detail of the metaball shapes. Besides, we design a straightforward method to preserve the local volume of organs by adjusting the sphere radii in the area under surgical instruments interaction. Figure 4a illustrates the deformation of a metaballs model for the liver by a grasper in minimally invasive surgery.

4.3 Skinning

After the deformation of metaballs, the final task is transmitting this deformation to the exterior surface. Therefore, we need to construct the mapping between the metaballs model and surface mesh of organs. This process is very similar to the skinning technique in the skeleton driven animation [25]. Here, we treat the spheres as the "interior skeleton" and the polygon mesh as the "skin". In addition, an automatic algorithm based on distance field function [6] is designed to assign the weights for each vertex in the surface mesh. Figure 4b illustrates the final deformation result of the surface mesh.

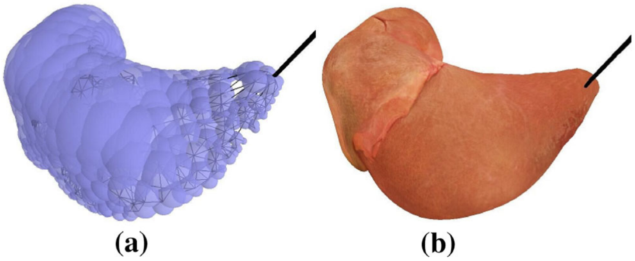


Fig. 4 Deformation of liver model by metaballs-based method. **a** Metaballs model, **b** surface mesh with texture

5 Dissection simulation of soft tissue

Collision detection is the first task that should be solved properly before the dissection process starts. We use the generalized cylinder-based collision detection [10] to handle the interactions between the surgical instruments and the soft tissue. After collision, if the scalpel moves across any triangle edge, the triangles in surface mesh will be subdivided. Meanwhile, as the interior structure of organs, the metaballs model should also be modified during the dissection. Considering the natural advantages of the mesh-free method in dissection simulation [26,27], we propose a hybrid approach to interlink metaballs model with the mesh-free method. When the scalpel moving across the spheres in the metaballs model, the dissected spheres will be removed and converted into nodal points. The movement of these points are driven by moving least squares shape functions, and it should also couple with the deformation of spheres nearby. When the scalpel leaves the soft tissue, these points will be separated and converted back into spheres. The whole process can be illustrated as shown in Fig. 5.

5.1 Mesh-free method based on moving least squares approximation

In cutting simulation, we use the nodal points in the mesh-free method to replace the dissected metaballs, as the local dynamic model. Once the scalpel touches a sphere in the metaballs model, this sphere will be removed and converted into points under uniform sampling. To guarantee the mass conservation, supposing m_i is the mass of sphere i and n is the number of nodal points converted from the sphere i, the mass of each converted point will be m_i/n.

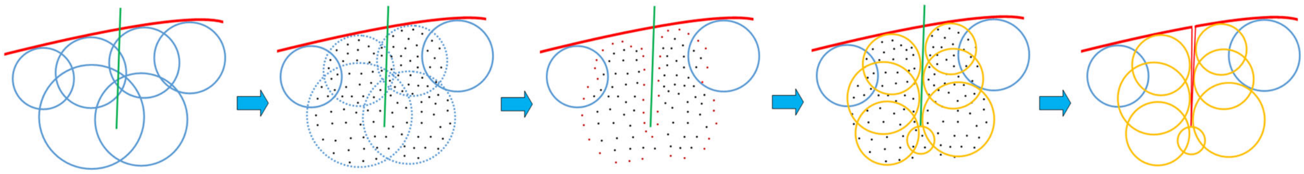


Fig. 5 Illustration of our hybrid-cutting simulation method

In continuum mechanics, the continuum elasticity equations describe how to compute the elastic stress inside a volumetric object by a given deformation field [27]. To employ these continuous elasticity equations in a numerical simulation, the volume is usually discretized into a large number of elements with finite size, for example, by the finite-element method (FEM). In mesh-free methods, the volume is discretized into a finite number of nodal points i without connectivity. All dynamic information, such as the position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, strain ϵ_i, stress σ_i, force $\mathbf{f}_i$, and deformation $\mathbf{u}_i$ can be computed by the physically correct mesh-free method.

The simulation loop can be described as follows. At the current time step t, from the displacement vectors $\mathbf{u}_i$, we approximate the nine spatial derivatives of three scalar fields which define the strain and stress tensors. The forces effected at the points are computed as the negative gradient of strain energy related to the displacements, and then integrated in time, yielding the updated displacements $\mathbf{u}_i'$ at the next time step $t + \Delta t$.

To measure the physical effect of a nodal point on its neighborhood, we use the following polynomial kernel [12] to compute the mass distribution of this point:

$$\omega_h(r) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \le r \le h \\ 0, & r > h \end{cases} \tag{4}$$

where r is the distance to the point and h is the support of the kernel.

To compute the strain, stress, and the elastic body forces, the spatial derivatives of the displacement field $\nabla \mathbf{u}$ are needed. These derivatives can be estimated from the displacement vectors of neighboring points. Here, we employ the moving least squares shape functions to approximate $\nabla \mathbf{u}$ with a linear basis. The spatial derivatives at the position of point $\mathbf{x}_i$ can be computed as Eq. (5):

$$\nabla u(\mathbf{x}_i) = \mathbf{A}_i^{-1} \sum_j (u_j - u_i)\omega_{ij}(\|\mathbf{x}_{ij}\|)\mathbf{x}_{ij}, \tag{5}$$

where $u(\mathbf{x}_i)$ is the x-component u of the displacement field $\mathbf{u} = (u, v, w)^T$. $\mathbf{x}_{ij}$ is the difference vector between point i and its neighboring points j. There is $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. ω_{ij} is the mass weight, which can be computed by Eq. (4), between point i and j. $\mathbf{A}_i$ is the moment matrix, which is only related to the distribution of point i and its neighboring points. $\mathbf{A}_i$ can be pre-computed by Eq. (6) and used for the computation of the derivative of v and w as well:

$$\mathbf{A}_i = \sum_j \omega_{ij}(\|x_{ij}\|)x_{ij}x_{ij}^T. \tag{6}$$

Using Eq. (5) and Green–Saint–Venant strain tensor [27], we can also compute the strain ϵ_i and the stress σ_i at point i by the following equations:

$$\epsilon_i = \mathbf{J}_i^T \mathbf{J}_i - \mathbf{I} = \nabla \mathbf{u}_i + \nabla \mathbf{u}_i^T + \nabla \mathbf{u}_i \nabla \mathbf{u}_i^T, \quad \sigma_i = \mathbf{C}\epsilon_i, \tag{7}$$

where $\mathbf{J}_i$ is the Jacobian matrix of point i. $\mathbf{C}$ is a rank four tensor, approximating the constitutive law of the material. Both ϵ_i and σ_i are symmetric 3 by 3 tensors. For an isotropic material, $\mathbf{C}$ is only related with Young's modulus and Poission's ratio.

According to the continuum mechanics theory, with ϵ_i and σ_i, we can estimate the strain energy U_i stored around point i by Eq. (8):

$$U_i = \int_\Omega u \, \mathrm{d}\Omega = \int_\Omega \frac{1}{2}(\epsilon_i \cdot \sigma_i)\mathrm{d}\Omega = \frac{1}{2}v_i(\epsilon_i \cdot \sigma_i), \tag{8}$$

where v_i is the rest volume of point i. The strain energy is a function of the displacement vector $\mathbf{u}_i$ for point i and the displacements $\mathbf{u}_j$ of all its neighbors. Taking the derivative with respect to these displacements, we can compute the force of each neighboring point j. In addition, the force acting on point i is the negative sum of all $\mathbf{f}_j$ acting on its neighboring points j:

$$\mathbf{f}_i = -\sum_j \mathbf{f}_j, \mathbf{f}_j = -\nabla_{\mathbf{u}_j} U_i = -v_i \sigma_i \nabla_{\mathbf{u}_j} \epsilon_i. \tag{9}$$

According to Newton's law, we can compute the acceleration $\mathbf{a}_i$ of point i by Eq. (10). Through time integration, we can get the new displacements $\mathbf{u}_i'$ of point i at the next time step. Here, we choose the LeapFrog explicit method [12] as the time integration scheme due to its high computation efficiency. In addition, we slightly reduce the time step of integration to ensure the stability of our mesh-free method:

$$\frac{\mathrm{d}^2 \mathbf{u}_i}{\mathrm{d}t^2} = \mathbf{a}_i = \mathbf{f}_i / m_i. \tag{10}$$

5.2 The coupling of metaballs model and mesh-free points

During dissection simulation, there are two different dynamic systems in the physical model. One is the PBD-based metaball model for the global deformation of digital organs. The other is the nodal points participating in the mesh-free method in the vicinity of a cutting path. One important task is to interlink the two different dynamic systems. Here, we use the stretching potential energy to transfer the motion of metaballs to point sets. This process can be illustrated as shown in Fig. 6a. In metaball model, the suppose sphere b is overlapped and connected with sphere a. After cutting, sphere b will be removed and converted into a number of nodal points. In addition, there must be some overlapping points (yellow points) inside sphere a. According to Hook's law, we use Eq. (11) to compute the stretching potential energy of sphere a:

$$E_a = \frac{1}{2}\sum_j \frac{m_j}{m_a + m_j} k(l'_{ja} - l_{ja})^2, \tag{11}$$

where E_a is the stretching potential energy of sphere a. Index j indicates all the neighboring spheres connected with sphere a. m_a, m_j are the masses of sphere a and j. l_{ja} is the rest length of spring ja. l'_{ja} is the current length of spring ja.

Then, E_a will be transferred by averaging to the overlapping points (Eq. (12)). U_i indicates the energy of overlapping point i in sphere a. n is the number of all the overlapping points. With Eqs. (9), (12), and (11), we can compute the displacement of nodal points by MLS approximation:

$$U_i = \frac{1}{n} E_a. \tag{12}$$

In addition, since the deformation of metaballs is driven by the PBD method, we directly use the displacement of overlapping points to transfer the motion of nodal points to metaballs. This process can be illustrated as shown in Fig. 6b. Supposing the displacement of sphere a is $\mathbf{u}_a$, $\mathbf{u}_a$ can be computed by Eq. (13), which is the average displacement of all the overlapping points inside sphere a:

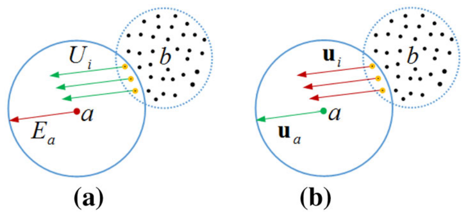


Fig. 6 Illustration of the coupling between metaballs and nodal points. **a** Energy transfer from metaballs to overlapping points, **b** displacement transfer from overlapping points to metaballs

$$\mathbf{u}_a = \frac{1}{n}\sum_i \mathbf{u}_i. \tag{13}$$

In Eq. (13), $\mathbf{u}_i$ indicates the displacement of overlapping point i. n is the number of all the overlapping points in sphere a.

5.3 Consistent material behavior preservation

In general, the computation speed of a physically mesh-free method is lower than of a metaballs model driven by a PBD method. It makes the stiffness of the metaballs model look higher than in the mesh-free method in visual performance. To ensure the deformation synchronization of these two dynamic systems, another task is the consistent material behavior preservation. Technically, the stiffness of the MLS-based deformation method depends on Young's modulus and Poisson ratio of soft tissue. These physical parameters can be measured by animal experiments [14]. In contrast, the stiffness of the PBD method depends on the time step size and the number of iterations. Here, we utilize a deformation synchronization strategy to adjust the material behavior for these two different dynamic systems. It is to lower the computation speed of the metaballs model driven by PBD and speedup the computation efficiency of the mesh-free method. One straightforward but effective way is to increase the number of PBD iteration steps and apply a GPU-based parallel acceleration by CUDA in the mesh-free method [30]. This has already been implemented in our system.

5.4 Metaballs and topology modification

When the scalpel leaves the surface mesh and cutting stops, the split points set should be converted back into metaballs with an updated topology connection. Our key strategy is to find the **medial points** in the split point set using the Centroid Voronoi Diagram-based method [6], and pack new spheres from these **medial points** to approximate the shape of the point set roughly. The whole process can be described as shown in Fig. 7. Nodal particle sets a and b are separated by the cutting plane P_0P_1 into two groups. Group 1 is the point set above P_0P_1. Group 2 is the point set under P_0P_1. In each group, we classify all the points into three categories. The first is **boundary points** (red points) which are near the boundary of a point group. The second is **medial points** (yellow points) which are near the medial surface of points in each group. The medial points approximately have the same distance from the boundary points in their local sphere region. They can be identified by the Centroid Voronoi Diagram-based method [6]. The third is **middle layer points** (black points) which are all other points in each group. Then, we equally down-sample the medial points as the center of new packing spheres (yellow circles in Fig. 7b) in each group. The

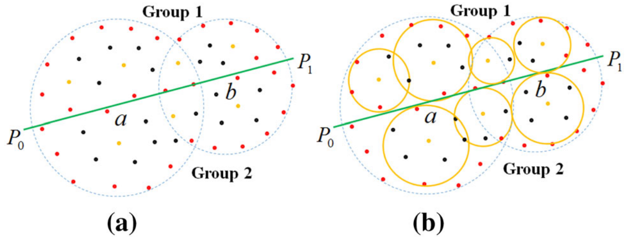


Fig. 7 Illustration of new sphere generation from split point sets. **a** Before the generation of new spheres, **b** after the generation of new spheres

radius of each sphere can be computed from the Euclidean distance between the sphere center and its nearest boundary point. Technically, the down-sampling density of medial points determines the number of newly generated spheres. When the sampling density decreases, the number of new spheres will increase. In addition, the updated metaballs model is more accurate, but with more computation cost. To keep the balance of accuracy and simulation speed, we employ Eq. (14) to adaptively control the down-sampling density for packing spheres:

$$d = \frac{1}{2} \max s(\mathbf{x}_i, \mathbf{p}), \tag{14}$$

where d is the distance between adjacent sphere centers in down-sampling. $\mathbf{x}_i$ are the 3d coordinates of medial point i. $\mathbf{p}$ is the cutting plane. $s(\mathbf{x}_i, \mathbf{p})$ indicates the projection distance between $\mathbf{x}_i$ and $\mathbf{p}$.

Algorithm 1 New sphere generation from the split point set

```
Input: point set (PS) in a group after dissection
Output: new sphere set
function NEW SPHERE GENERATION
    for i ∈ [1 to Number of PS] do
        search and mark boundary points from PS
    end for
    for i ∈ [1 to Number of rest points in PS] do
        search and mark medial points from PS
    end for
    for i ∈ [1 to Number of medial points] do
        for j ∈ [1 to Number of new generated spheres] do
            if distance between the center of sphere $j$ and
               medial points $i > d$ in (14) then
                compute the radius and pack a sphere at $\mathbf{x}_i$
                Number of spheres + 1
            end if
        end for
    end for
end function
```

Algorithm 1 describes the new sphere generation among a group of nodal points after dissection. The output is the newly generated spheres. For the topology construction among the newly generated spheres in each group, we use the scheme in [6] for sphere connection. These spheres only maintain the connectivity with the original spheres which are on the same side of the cutting plane as this new sphere. Figure 8 illustrates the result of cutting a liver model at different stages.

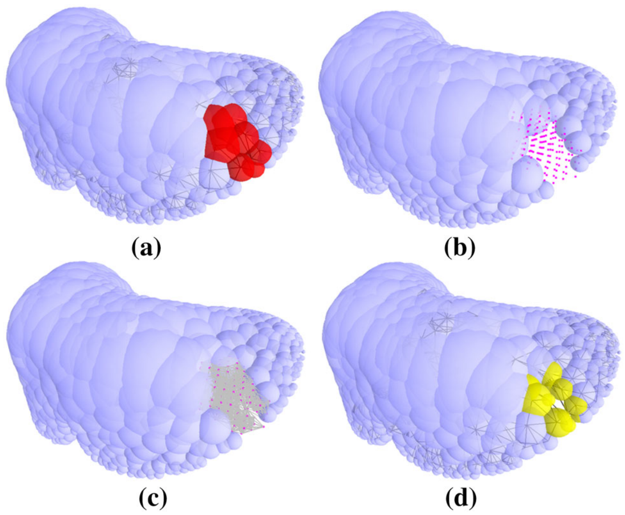


Fig. 8 Different stages of cutting a liver model. **a** Metaballs model before dissection, **b** metaballs model coupled with nodal points, **c** points with topology information after dissection, **d** updated metaballs model with connectivity after dissection

5.5 Cutting surface reconstruction

In the surgery simulation, all the display information of digital organs, such as texture and lighting, should be rendered on the surface mesh. Therefore, the last task for dissection simulation is the reconstruction of the cutting surface. As we use a mesh-free method for cutting, the nodal points near the sweep surface can be used to generate the surface mesh with more geometric detail. To lower the data size of new meshes, the points near the sweep surface should be down-sampled. Then, we employ Constraint Delaunay Triangulation [28] to generate cutting surface meshes for both sides. In addition, the approach in [29] is employed to generate the texture of a cutting surface. Finally, a mapping is built between the updated metaballs and the cutting surface mesh [6]. Figure 9 illustrates the cutting surface reconstruction result. The yellow spheres in Fig. 9a illustrate the modified metaballs after cutting.

6 Integration with haptic rendering

To apply our technique in virtual surgery training, we also integrate the dissection simulation with a haptic device. We choose Geomagic Touch (Phantom Omni), which can simulate the 6 DOF manipulation of scalpel, grasper or any other surgical tools in practical surgery.

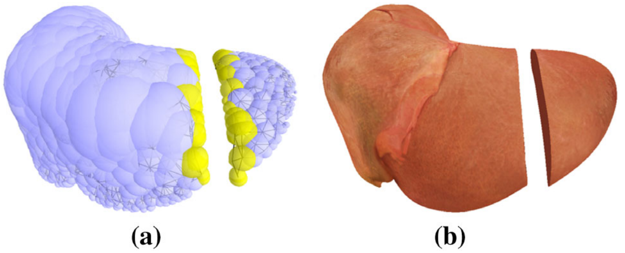


Fig. 9 Cutting surface reconstruction for a liver model. **a** Modified metaballs model after dissection, **b** surface mesh with texture

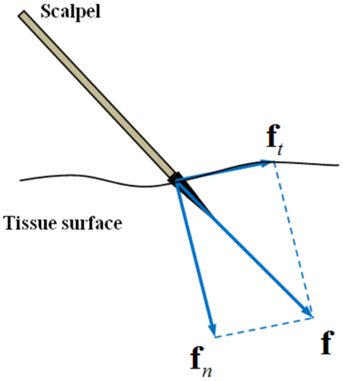


Fig. 10 Decomposition of force during cutting

We use the following mechanical model (Fig. 10) to describe the dissection behavior:

$$\mathbf{f} = \mathbf{f}_t + \mathbf{f}_n, \tag{15}$$

where **f** is the external force on the soft tissue surface. **f** can be decomposed into a normal force $\mathbf{f}_n$ perpendicular to the tissue surface and a tangential force $\mathbf{f}_t$. $\mathbf{f}_t$ is responsible for the incision movement of the scalpel; $\mathbf{f}_n$ controls the pressing force of the scalpel on the tissue surface. In our haptic rendering algorithm, the following criterion is used to determine whether the dissection starts or not:

$$|\mathbf{f}_n| > F_p, \quad |\mathbf{f}_t| > F_s. \tag{16}$$

In (16), the instrument will open the tissue only if the magnitude of $\mathbf{f}_n$ is larger than a given tissue dependent threshold F_p. Then, if the magnitude of $\mathbf{f}_t$ exceeds the value of static friction of tissue: F_s, the instrument will start dissection. During cutting, the system will monitor and analyze the value of **f** in each haptic frame. The scalpel will stop when $\mathbf{f}_t$ is smaller than the value of dynamic friction of tissue: F_d. In the cutting process, all parameters: F_p, F_s and F_d can be based on biomechanical experiments. This mechanical model can also be utilized to output the intuitive force during dissection simulation. The direction of the output friction force is opposite to the direction of scalpel movement. The force magnitude is proportional to the depth of the scalpel edge inserted into the tissue.

7 Experiments and application

We have implemented our dissection simulation technique using OpenGL, CUDA, and OpenHaptics. All the experiments run on a desktop with NVIDIA GeForceGT 630, Intel(R) Core(TM) i7-4790 CPU (3.60 GHz, 8 cores), and 8G RAM. The haptic rendering loop is running on a separate thread, and the updating rate is guaranteed around 1 kHz. Here, we have designed two groups of experiments.

The first experiment is cutting three different abdominal organs: liver, spleen, and gallbladder. These organs are also dragged and deformed by a grasper after dissection. Figure 11 illustrates the simulation results at different stages for

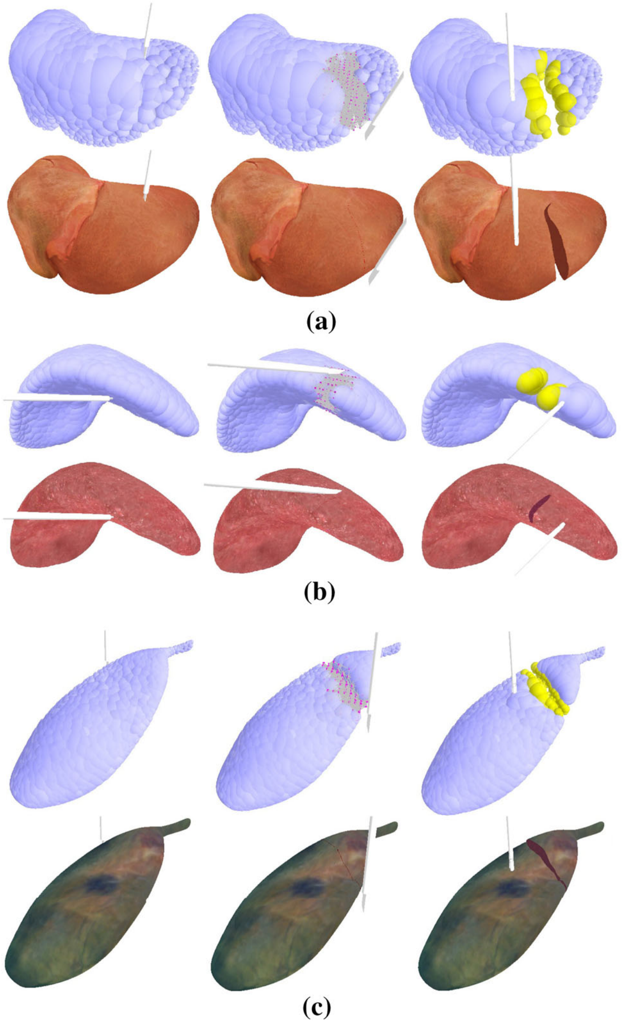


Fig. 11 Cutting simulation result of three abdominal organs at different stages. **a** Liver, **b** spleen, **c** gallbladder

Table 1 Data size and time performance of our dissection simulation method for three abdominal organs

Model	Number of vertices	Number of spheres	Time for metaballs construction (s)	Time for deformation (ms)	Time for cutting (ms)
Liver	8128	570	1884	6.25	9.52
Spleen	1415	452	1625	2.65	3.33
Gallbladder	1480	466	1572	2.14	2.58

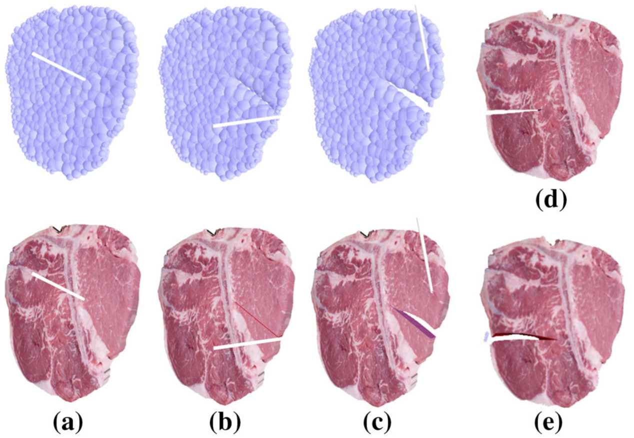


Fig. 12 Comparison of our method with the latest FEM- and PBD-based approaches in cutting a steak model. **a**–**c** Our cutting simulation results at different stages, **d** result in [30], **e** result in [19]

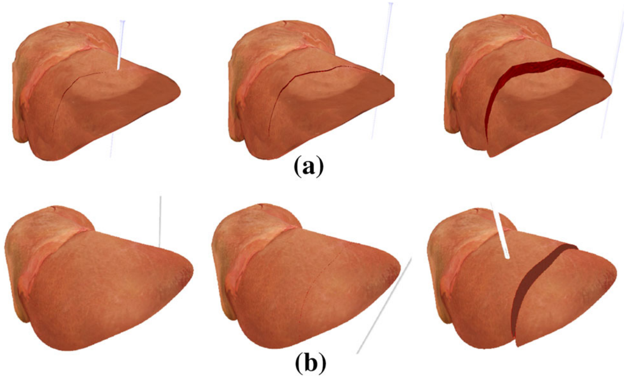


Fig. 13 Comparison of our method with a PBD-based approach in cutting a liver model. **a** Result in [19], **b** our result

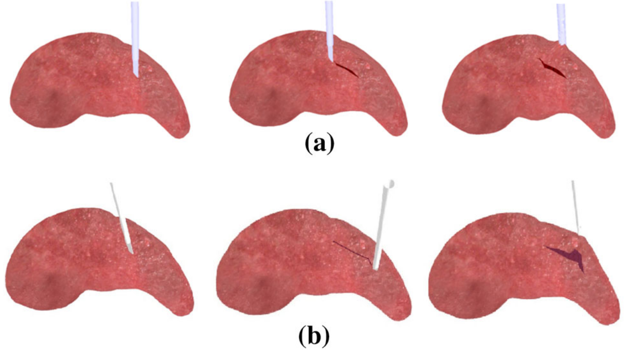


Fig. 14 Comparison of our method with a PBD-based approach in cutting a spleen model. **a** Result in [19], **b** our result

both the metaballs and surface mesh of organs. Our method supports curve cutting. Table 1 lists the data size and the computation time in the metaball construction, deformation, and cutting simulation. The metaball construction is processed offline, and it will not effect the real-time simulation performance of our method.

In the second experiment, we compare our method with the latest FEM and PBD-based dissection methods [19,30]. We cut the same steak, liver, and spleen model and record three key frames in Figs. 12, 13, and 14. Table 2 documents the number of tetrahedra or spheres and the computation time. Compared with the FEM-based method [30], our method can use far less elements and computation time to obtain almost the same visual performance. Meanwhile, since our method utilizes a certain number of points to generate the cutting surface, the geometry is more smooth and accurate than PBD-based method [19].

Our ultimate goal is to apply this novel dissection method to a VR-based medical-training system. To validate our techniques, we have incorporated it into a VR laparoscopic surgery simulator we developed. Figure 15 shows the interface of this VR simulation system. This prototyped medical simulator has been equipped with the following essential functionalities in laparoscopic surgery training:

Table 2 Comparison of three soft objects with FEM- and PBD-based-cutting simulation methods in data size and time cost (millisecond)

Model	Method	Number of elements	Deformation	Cutting
Steak	[30]	6272	25.2	8.6
	[19]	1477	1.7	6.5
	Ours	539	1.8	2.4
Liver	[19]	4079	4.1	15.2
	Ours	570	6.2	9.5
Spleen	[19]	2385	2.9	5.2
	Ours	466	2.1	2.6

- The laparoscope navigation and orientation in the virtual abdominal cavity.

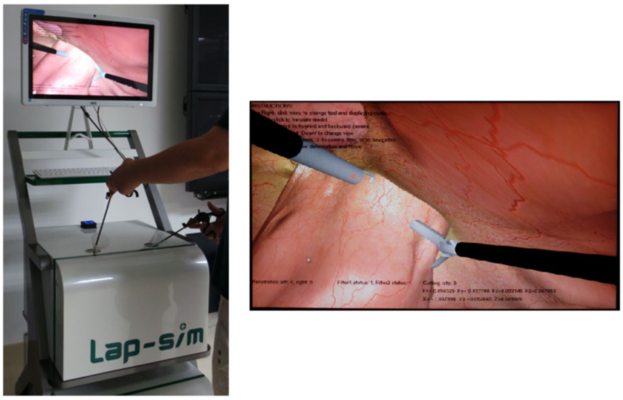


Fig. 15 Interface of our prototyped VR laparoscopic surgery simulator

- Laparoscopic real-time graphic rendering and deformation of soft tissue.
- The manipulation of a number of basic instruments, such as grasper, scalpel, scissor, and hook cautery.
- The force feedback control in haptic rendering.
- The simulation of basic procedures, such as grasping and cutting of soft tissue.
- Multimedia special effects, such as sound, and smoke generation in electrosurgical dissection.

8 Conclusion and future work

In this paper, we have detailed a real-time dissection approach for digital organs by tight coupling of geometric metaballs with a physically correct mesh-free method. The shape geometry of digital organ consists of both surface mesh and the metaballs, where the fine surface mesh is necessary for the exterior structure of organs, and the interior structure is filled by a set of overlapping spheres. At the physical level, we have demonstrated a hybrid framework that interlinks metaballs with a physics-driven mesh-free method based on MLS shape functions. The MLS approach enables the direct and rapid transformation from metaball geometry to local nodal formulations, participating in physical modeling and simulation over a continuum domain with a high-fidelity physical realm. For soft tissue dissection, the nature of our MLS-driven mesh-free method also facilitates adaptive topology modification and cutting surface reconstruction. Through physical deformation (involving dissection), we have employed PBD to expedite simulation towards real-time performance while driving metaballs to participate in the mesh-free formulation with dynamic updates of metaballs. Finally, we have migrated this technique into a prototype VR laparoscopic surgery simulator with a haptic interface.

Nevertheless, our hybrid dissection method still has some limitations. If the cutting path is long, the sweep surface may cut across quite a few spheres. It will generate a large number of nodal points, which may affect the real-time simulation performance. Fortunately, in real minimally invasive surgery settings, surgeons usually move the cautery hook in a short steps but with high frequency [1]. Hence, our cutting simulation satisfies the practical manipulation in a clinic. Besides, after many dissections, the metaballs usually generate quite a number of small spheres. This would seriously reduce the computational efficiency. In the near future, we plan to design a new scheme to optimize the ill-conditioned metaballs' model by merging too small spheres. Moreover, since mesh-free methods support material heterogeneities, we also plan to extend our dissection method to soft tissue with different material properties.

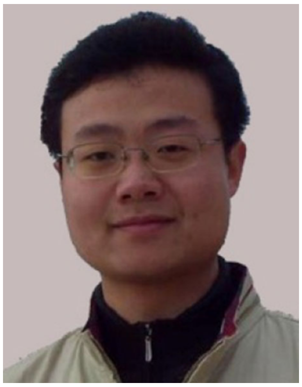
Acknowledgments This work was supported by the National Natural Science Foundation of China (Nos. 61402025, 61532002 and 61672149), the National Science Foundation of USA (Nos. IIS-0949467, 1047715, and 1049448), and the Fundamental Research Funds for the Central Universities.

References

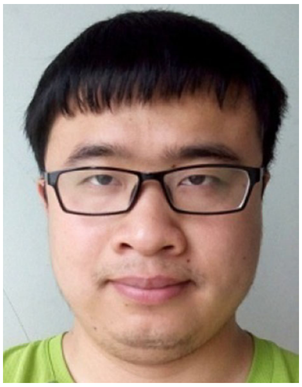
1. Simbionix. http://simbionix.com/simulators/lap-mentor/
2. Mentice. http://www.mentice.com/
3. Wu, J., Dick, C., Westerman, R.: Physically-based simulation of cuts in deformable bodies: a survey. Comput. Graph. Forum **34**, 161–187 (2015)
4. Wei, Y., Cotin, S., Dequidt, J.: A (near) real-time simulation method of aneurysm coil embolization. Aneurysm **8**(29), 223–248 (2012)
5. Gianluca, D.N., Melchiorri, C.: Surgery simulations and haptic feedback: a new approach for local interaction using implicit surfaces. International Conference on Applied Bionics and Biomechanics, Venice, October, 23–28 (2010)
6. Pan, J., Zhao, C., Zhao, X., Hao, A., Qin, H.: Metaballs-based physical modeling and deformation of organs for virtual surgery. Vis. Comput. **31**(6), 947–957 (2015)
7. Wu, J., Dick, C., Westermann, R.: Efficient collision detection for composite finite element simulation of cuts in deformable bodies. Vis. Comput. **29**(6–8), 739–749 (2013)
8. Cueto, E., Chinesta, F.: Real time simulation for computational surgery: a review. Adv. Model. Simul. Eng. Sci. **1**(11), 1–18 (2014)
9. Jeřábková, L., Bousquet, G., Barbier, S., Faure, F., Allard, J.: Volumetric modeling and interactive cutting of deformable bodies. Prog. Biophys. Mol. Biol. **103**(2–3), 217–224 (2010)
10. Pan, J., Chang, J., Yang, X., Liang, H., Zhang, J., Qureshi, T., Howell, R., Hickish, T.: Virtual reality training and assessment in laparoscopic rectum surgery. Int. J. Med. Robot. Comput. Assist. Surg. **11**(2), 194–209 (2015)
11. Liu, T., Bargteil, A.W., O'Brien, J.F., Kavan, L.: Fast simulation of mass-spring systems. ACM Trans. Graph. **32**(6), 1–7 (2013)
12. Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., Alexa, M.: Point based animation of elastic, plastic and melting objects. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 141–151 (2004)
13. Jones, B., Ward, S., Jallepalli, A., Perenia, J., Bargteil, A.W.: Deformation embedding for point-based elastoplastic simulation. ACM Trans. Graph. **33**(2), 1–9 (2014)
14. Steinemann, D., Miguel, A.O., Gross, M.: Fast arbitrary splitting of deforming objects. In: Proceedings of the 2006 ACM SIG-

GRAPH/Eurographics Symposium on Computer Animation, Sep 10, pp. 63–72 (2006)

15. Pietroni, N., Ganovelli, F., Cignoni, P., Scopigno, R.: Splitting cubes: a fast and robust technique for virtual cutting. Vis. Comput. **25**(3), 227–289 (2009)
16. Pauly, M., Keiser, R., Adams, B., Gross, M., Guibas, L.J.: Meshless animation of fracturing solids. ACM Trans. Graph. **24**(3), 957–964 (2005)
17. Adams, B., Wicke, M.: Meshless Approximation Methods and Applications in Physics Based Modeling and Animation. Eurographics 2009, Tutorial (2009)
18. Bender, J., Müller, M., Teschner, M., Macklin, M.: A survey on position based simulation methods in computer graphics. Comput. Graph. Forum **33**(6), 228–251 (2014)
19. Pan, J., Bai, J., Zhao, X., Hao, A., Qin, H.: Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics. Comput. Animat. Virtual Worlds **6**, 321–335 (2015)
20. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-rime applications. ACM Trans. Graph. **33**(4), 1–10 (2014)
21. France, L., Angelidis, A., Meseure, P., Cani, M.P., Lenoir, J., Faure, F., Chaillou, C.: Implicit Representations of the Human Intestines for Surgery Simulation. ESAIM: Proceedings, November 12, pp. 42–47 (2002)
22. Rivera-Rovelo, J., Bayro-Corrochano, E.: Medical image segmentation, volume representation and registration using spheres in the geometric algebra framework. Pattern Recognit. **40**, 171–188 (2007)
23. Bradshaw, G., Sullivan, C.O.: Sphere-tree construction using dynamic medial axis approximation. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 33–40 (2002)
24. Sorkine-Hornung, O., Cohen-Or, D., Lipman, Y., Alexa, M., Roessl, C., Seidel, H.-P.: Laplacian Surface Editing. Eurographics Symposium on Geometry Processing, pp. 1–10 (2004)
25. Pan, J., Yang, X., Xie, X., Willis, P., Zhang, J.: Automatic rigging for animation characters with 3D silhouette. Comput. Animat. Virtual Worlds **20**(2–3), 121–131 (2009)
26. Eberhard, P., Gaugele, T.: Simulation of cutting processes using mesh-free Lagrangian particle methods. Comput. Mech. **51**(3), 261–278 (2013)
27. Chung, T.J.: Applied Continuum Mechanics. Cambridge University Press, NY (1996)
28. Kallmann, M., Bieri, H., Thalmann, D.: Fully dynamic constrained delaunay triangulations. In: Kallmann, M., Bieri, H., Thalmann, D (eds.) Geometric Modeling for Scientific Visualization (Part of the series Mathematics and Visualization). Springer, Berlin, Heidelberg, pp. 241–257 (2011)
29. Li, X., Guo, X., Wang, H., He, Y., Gu, X., Qin, H.: Meshless harmonic volumetric mapping using fundamental solution methods. IEEE Trans. Autom. Sci. Eng. **6**(3), 409–422 (2009)
30. Yang, C., Li, S., Wang, L., Hao, A., Qin, H.: Real-time physical deformation and cutting of heterogeneous objects via hybrid coupling of meshless approach and finite element method. Comput. Animat. Virtual Worlds **25**(3–4), 423–435 (2014)



Junjun Pan received both BSc and MSc degree in School of Computer Science, Northwestern Polytechnical University, China. In 2006, he studied in National Centre for Computer Animation (NCCA), Bournemouth University, UK as PhD candidate with full scholarship. In 2010, he received PhD degree and worked in NCCA as Postdoctoral Research Fellow. From 2012 to 2013, he worked as a Research Associate in Center for Modeling, Simulation and Imaging in Medicine, Rensselaer Polytechnic Institute, USA. In Nov 2013, he was appointed as Associate Professor in School of Computer Science, Beihang University, China. His research interests include virtual surgery and computer animation.



Shizeng Yan is a master student in Beihang University. From 2014, he studied in the State Key Laboratory of Virtual Reality Technology and Systems in China. His research interests include virtual surgery and 3D visualization.

Hong Qin is a full professor of Computer Science in the Department of Computer Science at Stony Brook University (SUNY). He received his BS and his MS in Computer Science from Peking University, China. He received his PhD in Computer Science from the University of Toronto. Currently, he serves as an associate editor for The Visual Computer, Graphical Models, and Journal of Computer Science and Technology. His research interests include geometric and solid modeling, graphics, physics based modeling and simulation, computer aided geometric design, human–computer interaction, visualization, and scientific computing.

Aimin Hao is a professor in the Computer Science School and the Associate Director of the State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. He got his BS, MS, and PhD in Computer Science at Beihang University. His research interests are on virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.