Vis Comput (2015) 31:947–957
DOI 10.1007/s00371-015-1106-y

ORIGINAL ARTICLE

Metaballs-based physical modeling and deformation of organs for virtual surgery

Junjun Pan[1] · Chengkai Zhao[1] · Xin Zhao[1] · Aimin Hao[1] · Hong Qin[2]

Published online: 1 May 2015

Abstract Prior research on metaballs-based modeling solely focuses on shape geometry and its processing for organic objects. This paper takes a different approach by exploring a new metaballs-based physical modeling method for digital organs that are imperative to support virtual surgery. We propose a novel hybrid physical model comprising both surface mesh and the metaballs which occupy organs' interior. The finer surface mesh with high-precision geometric information and texture is necessary to represent the boundary structure of organs. Through the use of metaballs, the organ interior is geometrically simplified via a set of overlapping spheres with different radii. This work's novelty hinges upon the integration of metaballs and position-based dynamics (PBD) which enables metaballs-based organs to serve as physical models and participate in dynamic simulation. For the metaballs construction, we develop an adaptive approach based on Voronoi Diagram for model initialization. Using global optimization, an electrostatic attraction model is proposed to drive the metaballs to best match with the organ's boundary. Using PBD, we devise a novel metaballs-based deformation algorithm, which preserves two local shape properties via constraints on Laplacian coordinates and local volume. To retain the organ's smooth deformation, we propose a new skinning method based on distance field, and it is employed to build the mapping between the metaballs and organ boundary. This metaballs-based deformation technique has already been integrated into a VR-based laparoscopic surgery simulator.

Keywords Metaballs · Optimization · Organ · Deformation · Skinning

1 Introduction

With the ever-growing significance of virtual reality in medical fields, virtual surgery techniques and VR-based medical simulators have attracted increasing interests from both academic researchers and practitioners in industry. In recent years, there have been remarkable progresses in the research and application of virtual surgery. And several commercial VR surgical simulators have been developed with great success [1,2]. The core research content in virtual surgery is the soft tissue modeling, deformation, and the simulation of basic surgery procedure. For soft tissue deformation, varieties of approaches have been proposed. In technical essence, they can be classified into the finite element method (FEM), mass-spring model, meshfree method, etc. However, they all have their own limitations, such as weak performance not matching with the real-time requirement, unstable, noticeable geometry distortion, etc [3]. Metaballs, as a special and unique modeling method based on implicit surfaces, can effectively express continuous, blobby-like surfaces that could be of arbitrary topology. Its modeling advantage for shape geometry has been greatly exploited ever since its inception in the earlier 90's [4]. So far, most existing research of metaballs focuses on the geometric modeling and processing [5,6] with considerable success. In this paper, we propose to utilize the metaballs in a physically meaningful sense by marrying the geometry of metaballs with physical modeling and simulation for digital organs. Unlike popular FEM or

✉ Junjun Pan
pan_junjun@hotmail.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), New York, USA

meshfree methods, which have to construct shape geometry using a large number of small physical elements (hexahedra or particles), the metaballs could simplifies the physical structure with just a very small number of overlapping spheres. In particular, we propose a hybrid physical modeling approach for digital organs that have both mesh and metaballs for shape geometry. The finer surface mesh with high-precision geometric structure and texture at the detailed level is employed to represent the boundary structure of organs. Meanwhile, the interior structure of soft tissues is simplified by a number of coarse, overlapping spheres with different radii. The metaballs-based digital organs could directly participate in dynamic simulation, and this physical modeling capability is enabled by position-based dynamics (PBD). The simplified geometry of metaballs affords far less details for its interior structure, nonetheless, it could effectively reduce the computational cost during dynamic simulation. In our framework, we treat the centers of all participating balls as particles whose physical behavior is solely dictated by PBD (a much more simplified physical model in comparison with FEM and/or meshfree methods). This framework has already been adapted to a VR-based laparoscopic surgery simulator, in addition there are salient innovative contributions in our system.

- We propose an adaptive approach based on Voronoi Diagram to convert polygonal mesh to metaballs for model initialization. After obtaining a coarse model of metaballs, we utilize local optimization to adjust their radii, fill vacant space with more balls, and perform spheres merging for model refinement.
- To ensure the collision detection and deformation to be more accurate, we propose a novel global optimization method for the metaballs based on an electrostatic attraction model. It can drive the metaballs to achieve their best matches with the organ's boundary mesh.
- For physical modeling, we design a novel deformation algorithm of metaballs using position-based dynamics (PBD). This method can preserve two local shape properties via additional constraints: Laplacian coordinates and local volume.
- Based on distance field, we design an automatic skinning algorithm to assign different weights. This way we are able to generate the smooth deformation of organ surface, while metaballs and their connectivity are treated as the "interior skeleton" and the polygonal mesh serves as the "skin".

2 Related work

Physical modeling of organs is one of the essential tasks in virtual surgery. It has to handle the realistic deformation of soft tissues in real time. In principle, these deformation approaches can be classified into three different categories.

Finite element methods (FEM) FEM can perform accurate deformation result but with the cost of high computation time. Cuetoet et al. [7] gave a survey of the latest progress of FEM research and their application in virtual surgery. Wu et al. [8] presented a composite finite element-based simulation method for deformable objects. Jeřábková [9] designed a XFEM-based strategy to handle the deformation and cutting simulation of soft tissue in virtual surgery.

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. Liu et al. [10] proposed a scheme of time integration for standard mass-spring systems based on block coordinate descent. It can provide a fast solution for classical linear springs. Pan et al. [11] presented a hybrid dynamic model, which combines mass-spring and cosserat rod, to handle the large deformation of intestine.

Meshfree methods Meshfree methods are also called meshless methods. Without the topology constraints among vertices, this approach can simulate super soft object and fluid. Jones et al. [12] presented a meshfree method to deform elastoplastic object. Steinemann et al. [13] proposed a splitting method using a meshless discretization of the deformation field. Pietroni et al. [14] presented the splitting simulation based on a meshfree technique, to handle the interactive virtual cutting on deformable objects.

Position-based dynamics (PBD) is another fierce research topic in physical modeling, due to its robustness and position-based manipulation feature [15]. Most recently, Müller et al. [16] applied the PBD framework to simulate fluid and soft object in virtual real-time environment.

Quite different from the above physical modeling methods, most existing metaballs-based approaches focus on the geometric modeling using implicit surface [4,5,17]. Suzuki et al. [18] filled a large number of unified spheres in the liver model and deform the liver tissue by moving spheres. This method can simulate simple deformation and dissection in liver surgery. However, it hardly handles the large deformation of organs, especially for intestine. To the best of our current knowledge, until now there is little research work in the utility of metaballs as the physical model of organs in virtual surgery. In this paper, we proposed a strategy to deform the organs using a hybrid physical model which comprises both metaballs and surface mesh. Finally, we integrated this technique into a developed VR laparoscopic surgery simulator.

3 Framework overview

As shown in Fig. 1, we first present the framework overview as follows.

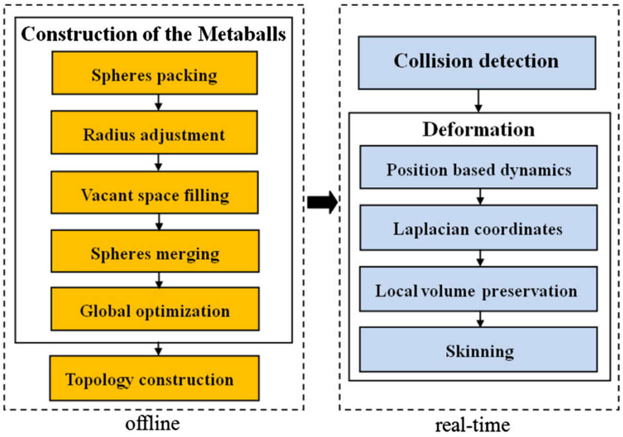


Fig. 1 The framework of our metaballs-based physical modeling and deformation method

Construction of the metaballs We first develop an adaptive approach based on Voronoi Diagram to generate the coarse metaballs model from organ surface mesh. Then a local optimization strategy is utilized by adjusting the spheres radii, filling the vacant space, and merging the extremely overlapping spheres. Finally, a global optimization is conducted by an electrostatic attraction model to drive the metaballs best matching the shape of mesh. Before real-time deformation computation, we construct the topology of metaballs model by determining the adjacent spheres for each sphere.

Deformation After collision detection, a novel deformation algorithm of metaballs is proposed. This method preserves two local shape properties: Laplacian coordinates constraint and local volume. During deformation, PBD is employed to compute the position of sphere centers first. To preserve the local detail of the metaballs, we use 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, we treat the deformed metablls as the "interior skeleton" and the polygon mesh as the "skin". An automatic skinning algorithm is presented to map the deformation to the surface mesh in real time.

4 Construction of the metaballs

In our technique, the hybrid physical model of organs consists of both exterior surface mesh and interior metaballs. So our 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 mesh. Besides, to reduce the computational cost during simulation, we expect the number of spheres could be as small as possible.

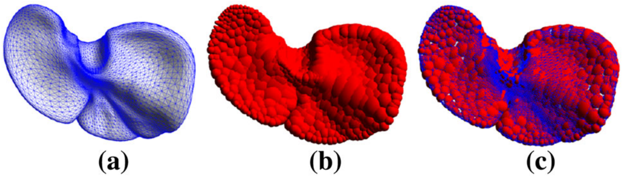


Fig. 2 The initial spheres packing for liver model. **a** The triangular mesh of liver. **b** Initial metaballs model of liver after spheres packing. **c** Liver model contained both mesh and metaballs

4.1 Spheres packing

Metaballs are extremely appropriate for geometrical modeling of organs as it can express continuous, blobby-like surfaces. Many researches have been carried out in this area. Bradshaw et al. [19,20] extended Hubbard's theory [21] and proposed several easy-to-use methods to pack spheres in polygon mesh. Here we use Sphere Tree Construction Toolkit [20] to pack the spheres in the triangular mesh of 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. Particularly, we generate a hierarchical sphere tree with eight children per node and we only leave the leaf nodes. Figure 2 illustrates the spheres packing result for a liver model. The spheres of leaf nodes approximate the object. However, the initial metaballs model after spheres packing cannot match the mesh boundary very well. There are large parts of spheres outside the surface of the object (Fig. 2c). So a radius adjustment process is needed for the next step.

4.2 Radius adjustment

A distance field is constructed for each sphere center. We compute its corresponding shortest distance to the surface. As we use a fine triangular mesh for the exterior structure of an object, the exact distance can be determined briefly by following formula:

$$D(\mathbf{c}) = \min d(\mathbf{c}, \mathbf{Tri}_i), \tag{1}$$

where $\mathbf{c}$ is the sphere center, $\mathbf{Tri}_i$ is the triangle in index i, and $d(\mathbf{c}, \mathbf{Tri}_i)$ indicates the shortest distance between $\mathbf{c}$ and triangle i. Then we adjust the sphere radius according to the distance $D(\mathbf{c})$, to ensure that the exterior parts of the sphere shrink inwards and contact with the surface boundary accurately. Figure 3a illustrates the result after radius adjustment. In Fig. 3a, we can find that small vacant space between spheres remains in some area. So next we will refine the metaballs model by a local optimization which involves vacant space filling and spheres merge.

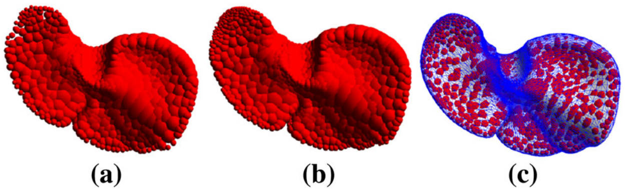


Fig. 3 The optimization of metaballs model for liver. **a** The result of radius adjustment. **b** The result of the global optimization. **c** Liver model contained both mesh and metaballs after optimization

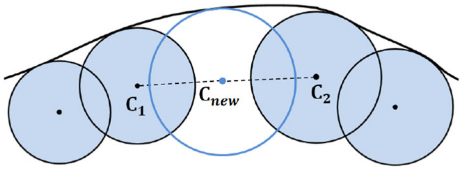


Fig. 4 The illustration of vacant space filling

4.3 Vacant space filling

The strategy of vacant space filling is the "interpolation", which is based on k-nearest neighbors algorithm. We search the vacant space around a sphere and fill it by adding new spheres (Fig. 4). The position of new sphere can be determined by following formula.

$$\mathbf{c}_{\text{new}} = \frac{d(\mathbf{c}_1, \mathbf{c}_2) + r_2 - r_1}{2d(\mathbf{c}_1, \mathbf{c}_2)}\mathbf{c}_1 + \frac{d(\mathbf{c}_1, \mathbf{c}_2) + r_1 - r_2}{2d(\mathbf{c}_1, \mathbf{c}_2)}\mathbf{c}_2, \quad (2)$$

where $\mathbf{c}_1 and \mathbf{c}_2$ are the coordinates of center for spheres 1 and 2 respectively, and $r_1 and r_2$ are radii for spheres 1 and 2 respectively. $d(\mathbf{c}_1, \mathbf{c}_2)$ is the distance between $\mathbf{c}_1$ and $\mathbf{c}_2$. The radius of new sphere is the shortest distance between $\mathbf{c}_{\text{new}}$ and mesh surface. $\mathbf{r}_{\text{new}}$ can be computed by (1).

Algorithm 1 outlines the vacant space filling among the sphere set. The output is the new set of spheres, which contains the original spheres and new added ones.

Algorithm 1 Vacant space filling among the sphere set

```
Require: sphere set (SS) of a metaballs model
Ensure: new sphere set
function VACANT SPACE FILLING
    queue $Q \leftarrow SS$
    while $Q$ is not empty do
        pop($\mathbf{s}$) from $Q$
        array $Neighbor \leftarrow$ search Neighbors($\mathbf{s}$)
        for i $\in$ [1 $to$ Number of $Neighbor$] do
            compute $\mathbf{c}_{new}$ from $\mathbf{s}$ and $Neighbor_i$ in(2)
            compute $r_{new}$ from $\mathbf{s}$ and $Neighbor_i$ in(1)
            if $\mathbf{c}_{new}$ is not inside any sphere in $Q$ and
                $\mathbf{c}_{new}$ is inside the mesh then
                push($Q$,$s_{new}(\mathbf{c}_{new}, r_{new})$)
            end if
        end for
    end while
end function
```

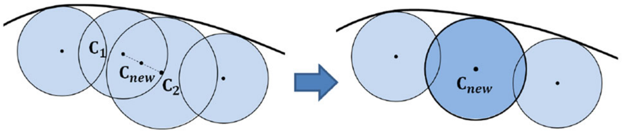


Fig. 5 The illustration of merging two extremely overlapping spheres

4.4 Spheres merging

Generally the packed spheres are not distributed equally, we need to merge the extremely overlapping spheres to simplify the metaballs model. If the distance between two sphere centers is less than the radius of the smaller sphere, these two spheres will be merged into single one. The center of this new sphere is located at the middle point of the center line between two original spheres. To keep the connection of topology, the radius can be determined by following formula:

$$r_{\text{new}} = \min(r, D(\mathbf{c}_{\text{new}})), \quad (3)$$

where r is the radius of the bigger sphere in the original spheres pair, and $D(\mathbf{c}_{\text{new}})$ represents the shortest distance between the new sphere center and mesh surface. $D(\mathbf{c}_{\text{new}})$ can be computed by (1). Fig. 5 shows an example of spheres merging.

4.5 Global optimization by the electrostatic attraction model

The ultimate aim of optimization is to make the shape of constructed metaballs best match with the boundary mesh. This criterion can make the collision detection and deformation (Sect. 5) more accurate. Since all spheres are restricted inside the mesh after radius adjustment, this criterion implies the vacant space between the metaballs and polygon mesh should be minimum. Here we voxelize the organ model at first. Then locate all the voxels which are not inside any sphere. We name this kind of voxel as V_{hollow} and compute the sum of V_{hollow}. Our objective function of optimization can be expressed as (4)

$$\min sum(V_{\text{hollow}}), \quad (4)$$

Supposing each sphere can move freely in a local space inside the mesh and its radius is fixed. An electrostatic attraction model is proposed to solve this optimization problem (Fig. 6). We treat each V_{hollow} in this voxelized organ model as a unit of "positive charge" and each sphere as the "negative charge". The quantity of negative charge is proportional to the volume of this sphere. To simplify the problem, we restrict each V_{hollow} can only attract the spheres in its neighboring space. The attracting force can be computed by Coulomb's law. And the resultant of attracting forces will drive each sphere

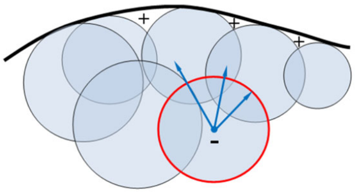


Fig. 6 The illustration of the electrostatic attraction model

move along its direction. Then this computation model can be described as Algorithm 2:

Algorithm 2 Global Optimization

```
voxelize model
array $Vhollow$, $AttractingForces$, set $AttractedSpheres$
initial $Vhollow$
compute $sum(Vhollow)$
while $true$ do
    for i ∈ [1 $to$ Number of $Vhollow$] do
        find all $neighboring\ sphere\ indexes$ of $Vhollow_i$
        for all index j do
            compute attracting force f
            $AttractingForces_j$ ← compute
              ResultantForce($AttractingForces_j$,f)
            push($AttractedSpheres$, j)
        end for
    end for
    for all k ∈ $AttractedSpheres$ do
        ResultantForce f ← $Attracting$
          $Forces[AttractedSpheres_k]$
        move $sphere[AttractedSpheres_k]$ along the
          direction of f for one voxel
        update $Vhollow$
        compute $sum'(Vhollow)$
        if $sum'(Vhollow) < sum(Vhollow)$ then
            $sum(Vhollow) \leftarrow sum'(Vhollow)$
            break
        end if
    end for
    if k == Number of $AttractedSpheres$+1 then
        break
    end if
end while
```

In each step, update the position of attracted spheres and V_{hollow}. If $sum(V_{\text{hollow}})$ decreases, save the current status of all spheres position. The iteration will end until $sum(V_{\text{hollow}})$ stop deceasing. The final distribution of spheres is the optimized metaballs model (Fig. 3b).

5 Physical modeling and deformation

Physical modeling and deformation of soft tissue is an essential task in virtual surgery. For our hybrid physical modeling approach, we use the constructed metaballs as the "interior skeleton" to deform the polygon mesh as "skin". The whole process involves the construction of topology, deformation of metaballs and skinning.

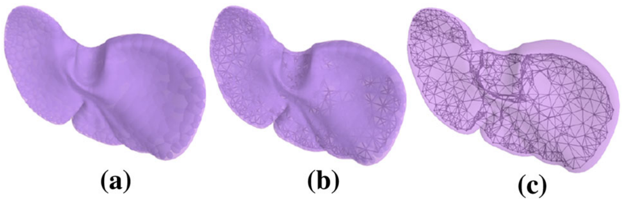


Fig. 7 The topology construction of the liver metaballs model. **a** The original metaballs model. **b** The metaballs model with connectivity. **c** The topology of connected sphere centers

5.1 Construction of topology

In constructed metaballs model, for a sphere i, whose center is $\mathbf{c}_i$, we select its adjacent connected spheres among all its overlapping spheres for topology connection. The selection criterion can be described as follows:

$$\text{Topo}(i) = \begin{cases} L(\mathbf{c}_i), & \text{num}(i) \leqslant N \\ L'(\mathbf{c}_i), & \text{num}(i) > N \end{cases} \tag{5}$$

where N is the threshold value, and $\text{num}(i)$ is the number of spheres overlapping with sphere i. $\text{Topo}(i)$ means the topology connection of sphere i. Function $L(i)$ indicates linking all the overlapping spheres center with $\mathbf{c}_i$. Function $L'(i)$ indicates linking N overlapping spheres center, which has the shortest distance between $\mathbf{c}_i$. Function $L'(i)$ can prevent each sphere connected with too many surrounding spheres and simplify the topology of metaballs. According to experimental experience, we usually set N as 9. Figure 7 illustrates the topology construction result of the metaballs model for liver. From Fig. 7c, we can find the topology of metaballs is like an irregular surface mesh.

5.2 Collision detection

Before interacting with organs, the collision detection is the first task that must be handled properly. Compared with mesh model, one significant advantage of the metaballs is the fast computation in collision detection. Since the shape of organ model can be approximated by the spheres set in metaballs, here we use a straightforward but effective strategy to handle the collision detection between metaballs and surgical instrument. During simulation, we first search the nearest sphere between the contact point of surgical instrument. Then compute the distance between the nearest sphere center and contact point. If the distance is smaller than the radius of this nearest sphere, the collision happens. Meanwhile, we use the method in [11] to solve the self-collision detection of organs.

5.3 Deformation of metaballs

The position-based dynamics (PBD) is employed to compute the position of sphere centers at first. To preserve the local detail of the metaballs shape, we use the Laplacian coordinates to update the position of sphere centers. Finally, the local volume preservation is employed to adjust the radii of spheres, in the area under surgical instruments interaction.

Position-based dynamics We choose PBD to deform the metaballs due to its robustness and position based manipulation feature. These advantages make PBD quite popular in the game industry and VR surgical simulator development [22]. Algorithm 3 outlines our PBD method. Here $\mathbf{c}_i$ stands for the center position of sphere i. $\mathbf{v}_i$ stands for the velocity of sphere i. m_i represents the mass of sphere i.

Algorithm 3 Our PBD method

```
$int\ N \leftarrow number\ of\ Spheres$
for i ∈ [1 $to$ $N$] do
    initialize $\mathbf{c}_i \leftarrow \mathbf{c}i^0$
    $\mathbf{v}_i \leftarrow \mathbf{v}i^0$
    $w_i \leftarrow 1/m_i$
end for
loop
    for i ∈ [1 $to$ $N$] do
        $\mathbf{v}_i^{new} \leftarrow \mathbf{v}_i + \mathbf{f}\Delta t w_i + Damp(\mathbf{v}_i)$
    end for
    for i ∈ [1 $to$ $N$] do
        $\mathbf{c}_i^{new} \leftarrow \mathbf{c}_i + \mathbf{v}_i^{new}\Delta t$
    end for
    loop $solveriterations\ times$
        stretching constraints($\mathbf{c}_1$,...,$\mathbf{c}_N$)
        Laplacian coordinates constraints($\mathbf{c}_1$,...,$\mathbf{c}_N$)
    end loop
    for i ∈ [1 $to$ $N$] do
        $\mathbf{c}_i^{fin} \leftarrow \mathbf{c}_i^{sol}$
        $\mathbf{v}_i^{fin} \leftarrow (\mathbf{c}_i^{sol} - \mathbf{c}_i)/\Delta t$
    end for
end loop
```

Generally, PBD involves stretching constraints, bend constraints, and volume constraints [15]. Considering the irregular topology of metaballs model, we cannot treat it as the polygon or polyhedron mesh. Here we only apply stretching constraints for the deformation of metaballs. Figure 8 gives an example of stretching constraint for two connected spheres. The constraint function about stretch can be described as follows:

$$C_{\text{stretching}}(\mathbf{c_1}, \mathbf{c_2}) = |\mathbf{c_1} - \mathbf{c_2}| - d, \tag{6}$$

where d is the initial distance between sphere centers $\mathbf{c_1}$ and $\mathbf{c_2}$ before simulation. The displacement in each iteration can be computed by the following formula:

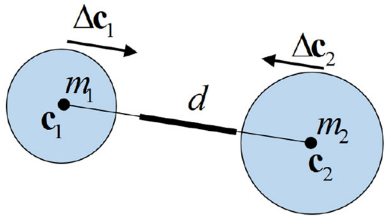


Fig. 8 The illustration of stretching constraint for two connected spheres

$$\Delta\mathbf{c_1} = -\frac{w_1}{w_1 + w_2}(|\mathbf{c_1} - \mathbf{c_2}| - d)\frac{\mathbf{c_1} - \mathbf{c_2}}{|\mathbf{c_1} - \mathbf{c_2}|}, \tag{7}$$

and

$$\Delta\mathbf{c_2} = \frac{w_2}{w_1 + w_2}(|\mathbf{c_1} - \mathbf{c_2}| - d)\frac{\mathbf{c_1} - \mathbf{c_2}}{|\mathbf{c_1} - \mathbf{c_2}|}. \tag{8}$$

Laplacian coordinates constraint Stretching constraints in PBD have an obvious limit. It only makes two-dimensional constraints and lack of constraints for three-dimensional space. Here we introduce the Laplacian coordinates constraint [23] to preserve the local detail of the metaballs shape. The method can be described as follows:

For any sphere m whose center is $\mathbf{c}_m$, $\mathbf{c}_i$ indicates the center of spheres which are connected with the sphere m in topology. The number of these connected spheres is n. So the center coordinates of these adjacent spheres center can be computed by the following formula.

$$\mathbf{c}_{\text{center}} = \sum_{i}^{n} \frac{\mathbf{c}_i}{n}, \tag{9}$$

Before the simulation loop, we can initialize the Laplacian coordinate of $\mathbf{c}_m$ by (10):

$$\mathbf{L}_m = \mathbf{c}_m - \mathbf{c}_{\text{center}} = \mathbf{c}_m - \sum_{i}^{n} \frac{\mathbf{c}_i}{n}. \tag{10}$$

During iteration, we treat $\mathbf{L}_m$ as a fixed vector constraint. $\mathbf{c}'_{\text{center}}$ is the updated $\mathbf{c}_{\text{center}}$ by (9). Then the updated position of $\mathbf{c}'_m$ can be computed by (11).

$$\mathbf{c}'_m = \mathbf{L}_m + \mathbf{c}'_{\text{center}}. \tag{11}$$

Local volume preservation The volume preservation is an important constraint in soft tissue deformation. In the deformation of the metaballs model, we propose a straightforward method to preserve the local volume by adjusting the radii of spheres in the area under surgical instruments interaction. Generally, most shape change occurs in the area under surgical instrument interaction for soft tissue deformation. So to simplify the volume preservation of entire organ, we only focus on the local volume preservation of the area under surgical instruments interaction. This straightforward method can be described as Fig. 9.

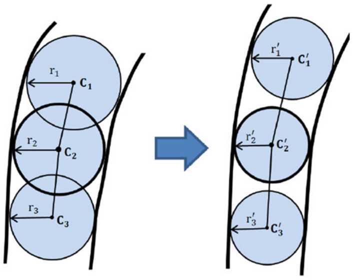


Fig. 9 The illustration of local volume preservation for three connected spheres in the metaballs

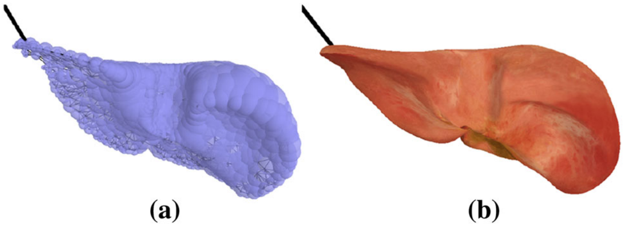


Fig. 10 The deformation of liver model by our hybrid physical modeling method. **a** The metaballs model. **b** The surface mesh with texture

Supposing sphere 2 is the contacted sphere under surgical instruments interaction. Before deformation, the center position of sphere 2 is $\mathbf{c}_2$ and the radius is r_2. The center positions of its adjacent connected spheres are $\mathbf{c}_1$ and $\mathbf{c}_3$. And the radii are r_1 and r_3. $\sum d(\mathbf{c}_2)$ is the sum of center distance among all adjacent spheres and sphere 2. So in Fig. 9, $\sum d(\mathbf{c}_2)$ is $|\mathbf{c}_1 - \mathbf{c}_2| + |\mathbf{c}_3 - \mathbf{c}_2|$. During deformation, supposing the center positions become $\mathbf{c}'_i$ (i=1,2,3). $\sum d(\mathbf{c}'_2)$ is $|\mathbf{c}'_1 - \mathbf{c}'_2| + |\mathbf{c}'_3 - \mathbf{c}'_2|$. Then the radii r'_i of spheres in this local area can be adjusted by (12).

$$r'_i = r_i \left(\frac{\sum d(\mathbf{c}_2)}{\sum d(\mathbf{c}'_2)} \right)^{\frac{1}{2}}, \quad (i = 1, 2, 3). \tag{12}$$

Figure 10a illustrates the deformation of the metaballs model for liver by a grasper in minimally invasive surgery.

5.4 Skinning

After the deformation of metaballs, the final task is transforming this deformation to the exterior surface. So 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 [24]. Here we treat the spheres as the “interior skeleton” and the polygon mesh as the “skin”. And an automatic algorithm based on distance field function is designed to assign the weights for each vertex in surface mesh. It can be described as follows:

For a sphere, its field strength of weighting can be computed by a function about its radius, the distance between the vertex on mesh and its sphere center. Here we use the nonlinear field strength of weighting, which can be described as a Gaussian function:

$$f(d, r) = c \cdot exp^{-\frac{(d-r)^2}{r^2}}, \tag{13}$$

where r stands for the sphere radius, d is the distance between a point and the sphere center and c is a constant coefficient.

For a vertex $\mathbf{v}$ on the surface mesh, we can find its attached spheres $i\,(i = 1, 2, \ldots n)$ among the metaballs model by this condition: The attached spheres satisfy $f(d_{\mathbf{v},\mathbf{c}_i}, r_i) \leqslant T$, where $d_{\mathbf{v},\mathbf{c}_i}$ is the distance between $\mathbf{v}$ and sphere center $\mathbf{c}_i$. T stands for a threshold.

So for $\mathbf{v}$, each attached sphere i has a different influence weight on $\mathbf{v}$. We set a weighted center coordinate of these attached spheres. And it can be computed as (14).

$$\mathbf{c}_{\text{center}} = \sum_{i}^{n} \frac{f(d_{\mathbf{v},\mathbf{c}_i}, r_i)}{\sum_i^n f(d_{\mathbf{v},\mathbf{c}_i}, r_i)} \mathbf{c}_i, \tag{14}$$

where $\mathbf{c}_i$ is the center of sphere i. Before simulation, the initial displacement between $\mathbf{v}$ and $\mathbf{c}_{\text{center}}$ will be computed and saved by (15):

$$\mathbf{disp} = \mathbf{v} - \mathbf{c}_{\text{center}}. \tag{15}$$

During simulation, we can compute the updated $\mathbf{v}'$ using (16) in each frame.

$$\mathbf{v}' = \mathbf{c}'_{\text{center}} + \frac{\sum_i^n r'_i}{\sum_i^n r_i} \mathbf{disp}, \tag{16}$$

where $\mathbf{c}'_{\text{center}}$ is the updated weighted center coordinate of attached spheres. r'_i is the updated radius of attached sphere i. And r'_i can be computed by (12). Figure 10b illustrates the final deformation result of surface mesh for liver model by this skinning method.

6 Experiments and application

We have implemented our metaballs-based physical modeling and deformation technique using OpenGL and OpenHaptics. All the experiments run on a desktop with NVIDIA GeForce GTX 460, Intel(R) Core(TM)2 Quad CPU (2.66-GHz, 4 cores), and 4G RAM. The haptic rendering loop is running on a separate thread, so the update rate is guaranteed around 1 kHz. We have designed three sets of experiments.

The first experiment is constructing the metaballs model for four abdominal organs: liver, gall bladder, intestine, and stomach. These organs are also interacted and deformed by a grasper. Figure 11 illustrates the results except the liver. Table 1 documents the data size and also the computation

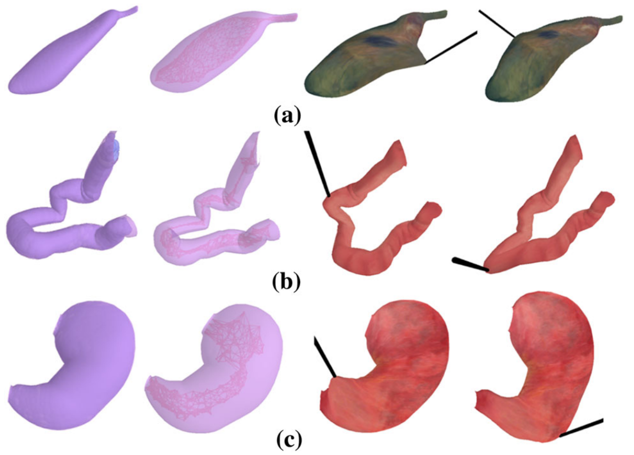


Fig. 11 The metaballs model and deformation result of three abdominal organs (From *left to right* mesh and the metaballs model; the topology connections for the metaballs; deformation result 1; deformation result 2). **a** Gall bladder, **b** intestine, **c** stomach

time in metaballs construction and deformation. Since the metaballs construction is processed off line in initialization, it will not effect the real-time performance in simulation.

The second experiment is to verify the effect of two local shape preservations in our deformation algorithm. The first is the Laplacian coordinates constraint to preserve the local detail. Figure 12 illustrates the comparison result. From Fig. 12b and c, some areas of mesh surface are smoother under the Laplacian coordinates constraint. It preserves the original shape feature of liver. The second is the local volume preservation. Figure 13 illustrates the comparison result. In Fig. 13b, under the local volume preservation, the section of intestine dragged by a grasper becomes thinner. It is more realistic in visual performance.

In the third experiment, we compare our deformation method with three typical approaches (FEM, mass-spring and PBD). Table 2 is the numerical comparison. Hexahedra are used in the FEM tests. PBD tests use stretch constraints and volume conservation constraints. Figures 14 and 15 show the result. Here we choose the Explicit Euler solver for mass-spring method. From this experimental results, the computation time of our method is at the same level with mass-spring and PBD, much lower than the FEM. Due to the local detail and volume preservation, the deformed liver surface by our method is much more realistic than PBD and mass-spring method. So among these approaches, the cost performance of our method is the best.

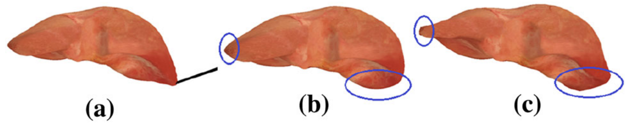


Fig. 12 The comparison of deformation with and without Laplacian coordinates constraint for liver model. **a** Drag the liver tissue by a grasper, **b** deformation recovery with Laplacian coordinates constraint, **c** deformation recovery without Laplacian coordinates constraint

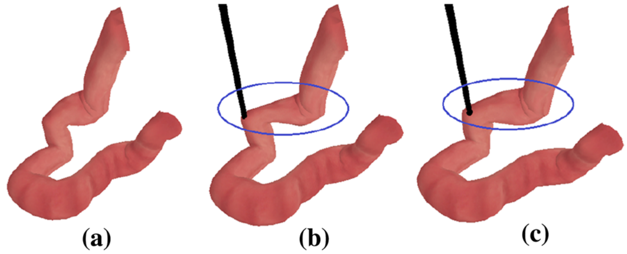


Fig. 13 The comparison of deformation with and without the local volume preservation. **a** Original intestine model, **b** the deformation of intestine with the local volume preservation, **c** the deformation of intestine without the local volume preservation

Our ultimate goal is to apply this novel physical modeling method to the virtual reality-based medical training and treatment. To validate our deformation method, we have incorporated it into a prototyped VR laparoscopic surgery simulator developed. Figure 16 shows the hardware and software interface of this virtual reality simulation system. This prototyped medical simulator has been equipped with the following essential functionalities in laparoscopic surgery training:

- The laparoscope navigation and orientation in the virtual abdominal cavity.
- Laparoscopic real-time graphic rendering and deformation of soft tissue.
- The manipulation of basic MIS instruments, such as grasper, scissor, and hook cautery.
- The force feedback control in haptic rendering.
- The simulation of four basic procedures: small object transfer, dissection, suturing, and ligation.

Table 1 Data size and time performance of our metaballs-based deformation method for four abdominal organs

Model	Number of vertices	Number of spheres	Number of center lines in metaballs	Time for metaballs construction (second)	Time for deformation (ms)
Liver	8128	581	3588	2207	7.5
Gall bladder	1461	475	3293	1179	2.9
Stomach	3349	452	3161	2050	8.1
Intestine	887	383	2625	754	1.2

Table 2 Comparison between our method and three typical deformation approaches

Method	Gall bladder		Liver	
	Elements/constraints	Deformation (ms)	Elements/constraints	Deformation (ms)
FEM	819	39.2	967	48.6
Mass-spring	4991	0.75	6427	2.5
PBD	4991 + 3185	1.15	6427 + 4079	2.2
Our method	3242 + 466	2.7	3588 + 581	7.6

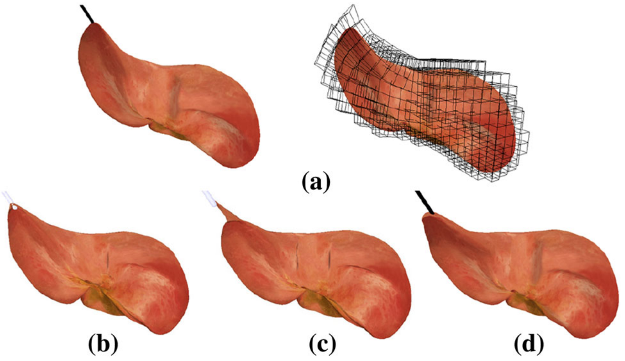


Fig. 14 The comparison between our method and three typical approaches in the liver deformation. **a** FEM, **b** mass-spring method, **c** PBD-based method, **d** our method

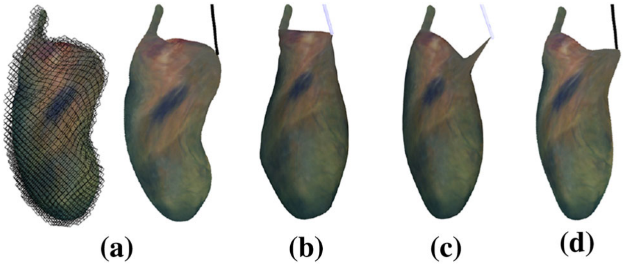


Fig. 15 The comparison between our method and three typical approaches in the gall bladder deformation. **a** FEM, **b** mass-spring method, **c** PBD-based method, **d** our method

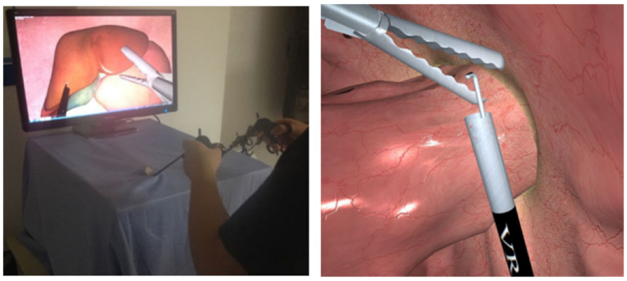


Fig. 16 The interface of a prototyped VR laparoscopic surgery simulator

- Multimedia special effects, such as sound, and smoke generation in electrosurgical dissection.
- Auxiliary functionalities, such as skills evaluation.

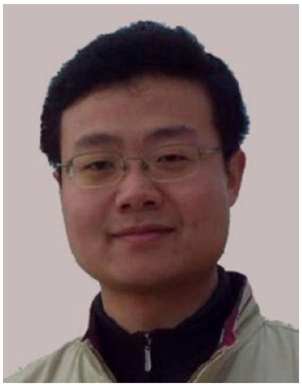
7 Conclusion and future work

In this paper, we have presented a metaballs-based approach for organ modeling and its deformation valuable for virtual surgery. It makes use of a hybrid physical model comprising both surface mesh and the metaballs. The finer surface mesh is to represent the exterior structure of organs, and the interior structure is constructed by a set of overlapping spheres with different radii. The inner metaballs are further equipped with physical properties, enabled by PBD. For the metaballs construction, we proposed an adaptive approach based on Voronoi Diagram for model initialization and suggested an optimization strategy to find the best match between the metaballs model and the shape geometry of organs. For physical defomation, our novel metaballs-based algorithm is devised based on PBD. This algorithm also preserves two local shape properties via additional constraints: Laplacian coordinates and local volume of organs. We also developed a skinning method based on distance field to build the mapping between the metaballs and surface mesh. From comprehensive experimental results, we have observed that this metaballs-based method can offer real-time and realistic deformation of organs, while preserving the local shape feature of original model. Finally, we have successfully migrated this approach into a prototype VR laparoscopic surgery simulator. Nevertheless, our algorithm also has limitations due to its special aim. Currently, it is only suitable to deform blobby-like objects, such as organs. It hardly performs plausible deformation for objects with sharp shape. Besides, we simplify the organs interior via a set of overlapping spheres. The physical accuracy of our deformation method is lower than FEM. In the near future, we plan to extend our metaballs-based method to the dissection simulation. We also plan to further exploit the parallel acceleration and apply CUDA to the metaballs construction and collision detection.

Acknowledgments We thank Junxuan Bai and Chen Yang for their work in the experiments. This research is supported by National Natural Science Foundation of China (No. 61402025, 61190120, 61190121, 61190125), National Science Foundation of USA (No. IIS-0949467, 1047715, 1049448) and the Fundamental Research Funds for the Central Universities

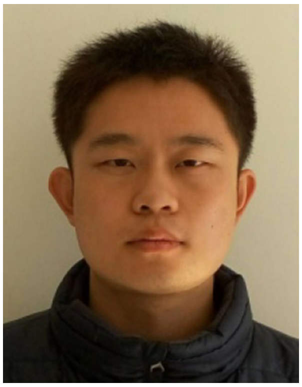
References

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. Eurographics **7**(3), 1–19 (2014)
4. Muraki, S.: Volumetric shape description of range data using blobby model. Comput. Graph. **25**(4), 227–235 (1991)
5. Wei, Y., Cotin, S., Dequidt, J.: A (Near) real-time simulation method of aneurysm coil embolization. Aneurysm **8**(29), 223–248 (2012)
6. 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, Oct. pp. 23–28 (2010)
7. Cueto, E., Chinesta, F.: Real time simulation for computational surgery: a review. Adv. Model. Simul. Eng. Sci. **1**(11), 1–18 (2014)
8. 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)
9. Jeřábková, L., Torsten, K.: Stable cutting of deformable objects in virtual environments using XFEM. IEEE Comput. Graph. Appl. **29**(2), 61–71 (2009)
10. 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)
11. Pan, J., Chang, J., Yang, X., Qureshi, T., Howell, R., Hickish, T., Zhang, J.: Graphic and haptic simulation system for virtual laparoscopic rectum surgery. Int. J. Med. Robot. Comput. Assist. Surg. **7**(3), 304–317 (2011)
12. 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)
13. Steinemann, D., Miguel, A.O., Gross M.: Fast arbitrary splitting of deforming objects. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Sep 10, 63–72, (2006)
14. 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)
15. 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)
16. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. ACM Trans. Graph. **33**(4), 1–10 (2014)
17. France, L., Angelidis, A., Meseure, P., Cani, M.P., Lenoir, J., Faure, F., Chaillou, C.: Implicit Representations of the Human Intestines for Surgery Simulation. In: ESAIM: Proceedings, Nov. 12, pp. 42–47 (2002)
18. Suzuki, S., Suzuki, N., Hattori, A., Uchiyama, A., Kobayashi, S.: Sphere-filled organ model for virtual surgery system. IEEE Trans. Med. Imaging **23**(6), 714–722 (2004)
19. 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)
20. Bradshaw, G., Sullivan, C.O.: Adaptive medial-axis approximation for sphere-tree construction. ACM Trans. Graph. **23**(1), 1–26 (2004)
21. Hubbard, P.M.: Approximating polyhedra with spheres for time-critical collision detection. ACM Trans. Graph. **15**(3), 179–210 (1996)
22. PhysX-Nvidia. http://physxinfo.com/wiki/
23. 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)
24. Pan, J., Yang, X., Xie, X., Willis, P., Zhang, J.: Automatic rigging for animation characters with 3D silhouette. Comput. Anim. Virtual Worlds **20**(2–3), 121–131 (2009)



Junjun Pan is an associate professor in School of Computer Science, Beihang University, China, from 2013. He 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 Ph.D. candidate with full scholarship. In 2010, he received his Ph.D. 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. His research interests include virtual surgery and computer animation.

Chengkai Zhao is a master student in Beihang University. From 2013, he is studying in the State Key Laboratory of Virtual Reality Technology and Systems in China. His research interests include virtual surgery and geometric modeling.



Xin Zhao received his BSc degree in Applied Mathematics, Northwestern Polytechnical University, China. Now he is a master student in Beihang University. From 2013, he is studying in the State Key Laboratory of Virtual Reality Technology and Systems, China. His research interests focus on virtual reality and physical modeling.

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 Ph.D. degrees in Computer Science at Beihang University. His research interests include virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.

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 degrees in Computer Science from Peking University, China. He received his Ph.D. degree in Computer Science from the University of Toronto. 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.