IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 7, 2019, accepted March 28, 2019, date of publication April 3, 2019, date of current version April 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909061

Point Primitives Based Virtual Surgery System

YANNI ZOU[1], PETER X. LIU[1,2], (Fellow, IEEE), DEDAO WU[1], XIAOHUI YANG[1], AND SHAOPING XU[1]

[1]School of Information Engineering, Nanchang University, Nanchang 330031, China
[2]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Peter X. Liu (xpliu@sce.carleton.ca)

This work was supported in part by the National Natural Science Foundation of China under Grant 61862044, Grant 51765042, and Grant 61762062, and in part by the Key Project of Science and Technology Department of Jiangxi Province of China under Grant 20171ACB20007.

ABSTRACT In order to achieve a high degree of visual realism in surgery simulation, we present a virtual surgery system framework, which is based on point primitives for the virtual surgery scene rendering and the biomechanical calculation of the soft tissue. To embody the superiority of this framework, two virtual surgery systems based on point primitives we developed are exhibited in this paper. Six critical functional modules were selected as representative of basic and advanced virtual surgery skill. These modules were: 1) point-based texture mapping; 2) deformation simulation; 3) cutting simulation; 4) tearing simulation; 5) dynamic texture mapping; and 6) 3-D display. These modules were elaborated by including working principle, execution process, and the performance of the algorithm. The experimental results have shown that point primitives-based virtual surgery systems obtained higher performance in terms of computational efficiency and rendering effect than traditional meshes-based virtual surgery system.

INDEX TERMS Point primitives, biomechanical calculation, framework, functional module, virtual surgery system.

I. INTRODUCTION

Technical proficiency is a key factor to measure the ability of a surgeon. To improve doctors' surgery techniques, surgical simulation is emerging as a potential method. Virtual surgery simulator can allow autonomous skills training. It can also incorporate the different techniques, anatomies, and pathologies required for a lot of surgical specialties. Using advanced graphics and haptics, simulation is striving toward realistic, dynamic tissue behavior.

In recent years, many researchers have developed lots of virtual surgery systems [1]–[7]. These virtual systems are roughly divided into two types according to the data model of virtual object. One type is based on meshes which records topological information among point data. The other is based on points in which no connectivity information is stored. Virtual simulators based on meshes include many products. For example, Suzuki *et al.* in Jikei University designed a virtual liver surgery system [8]. They first obtain the contour of organs by processing CT and MRI medical image sequence. Then, they filled the organ's contour with rigid spheres of same size, and rendered the surface triangles of soft tissue which connected surface points and the centers of inner rigid spheres. When external forces were applied to the geometric model of soft tissue, the rigid body spheres' position changed and interconnected phenomenon happened under the action of springs which connected these rigid spheres and their neighbors. The pulling, pressing and cutting of soft tissue can be simulated in this virtual surgery system, further the volume of the excised portion can be computed according to the volume of the removed spheres. However, the filling sphere has some disadvantages: the initial rigid spheres are filled randomly, which could lead to different deformation simulation effects when their filling modes are different. Brown B team in Stanford University also designed a virtual surgery simulator which can realize three-dimensional display. The deformation and stitching of blood vessels were simulated and realistic simulation results were realized in the simulator [9]. Spring mass method was adopted to compute deformation, cutting and stitching etc., and the model's performance was evaluated accurately according to the actual material parameters. However, the system fails to achieve the 1000Hz refresh of interactive feedback force, so the user cannot feel authenticity of force.

In addition, Berkley *et al.* at the University of Washington developed a virtual surgical system with augmented reality

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

function [10]. Stereoscopic display technology and augmented reality technology were introduced in stitching simulation, which improved the user's immersion to some extent in the training process. In order to realize real-time deformation calculation, a linear finite element method was employed to build the deformation model for virtual object. This method makes the modeling more precise. However, the drawback of this method is that the original grid data cannot be updated in real time, which makes it no longer applicable in the simulation of large deformation (such as cutting and tearing). INRIA also developed a virtual laparoscopic liver resection surgery system. In this system, the geometric model of liver and its adjacent tissues were reconstructed by using CT image sequence. Then, soft tissue deformation was calculated by using the concentric implicit finite element modeling method [11]. The system can capture the physical properties of soft tissue and provide users correct feedback. Nevertheless, during the process of cutting simulation, the appearance of pathological grids when the grids were updated, could lead to the instability of the numerical value in the calculation process [12]. The team also designed the cataract removal, virtual endoscopy liver resection and brain tumor removal system. Moreover, the national council of Canada (NRC) [13] developed the NeuroTouch virtual surgery simulator. MRI images of patients were rendered into a three-dimensional model of the individual brain. By operating a physical device similar to a scalpel, doctors can view and interact with tumors and other tissues on the screen in real time. Finite-element methods (FEM) was adopted to compute the soft tissue deformation. Gray matter hemorrhage and pulse, hemostasis, and soft tissue suction can also be simulated in the simulator.

These virtual surgery systems mentioned above are based on mesh data. Deformation behavior of soft tissue as a result of interventions with surgical instruments can be achieved by using mass-spring methods (MSM) or FEM. MSM [14]–[18] allows the deformation calculation in real time by using suitable numerical integration with varying step size, while it is difficult to realize a robust and stable simulation. FEM [19]–[23] allows integrating material properties of object in the FEM framework so that it can achieve a high-quality simulation. However, doing this also makes it could suffer high computational costs. As well, the error and distortion from the remeshing process could occur when deformation is large. Furthermore, surface rendering of object based on triangles can lead to surgery scenes with poor visual effects when the topology of the object changes.

To avoid problems caused by the mesh-based approaches, many researchers have studied meshless methods [24]–[27]. The calculation of these methods is based on point data, which makes them particularly suitable for simulating large deformation, tearing and cutting tasks since no grid information needs to be maintained. As far as we know, no one has designed a complete virtual surgery system based on point primitives. In view of this situation, our team designed two virtual surgery systems based on point primitives. The main contributions of our scheme compared with the conventional mesh-based methods can be summarized as follows:

- Both physical calculation (deformation, cutting and tearing) and renderings are all based on point data and no topology information needs to be maintained, the implementation requires much less memory space compared with conventional ones.
- the error and distortion from the remeshing process for mesh models when large deformation happens can be avoided in our system.
- At each time step, topology information needs to be reorganized and updated when the topology is broken in mesh-based system, which could lead to low computational efficiency. While our system avoids this problem and significantly reduces computing costs.

The effect images of our system are shown in Fig.1 and Fig.2. The framework (including main functional modules) of our simulator is shown in Fig.3. All of the functional modules of two simulator include physical computing, collision detection and rendering are based on point data. Only a type of source data is adopted in the system, which makes the calculation of functional modules more concise and convenient.

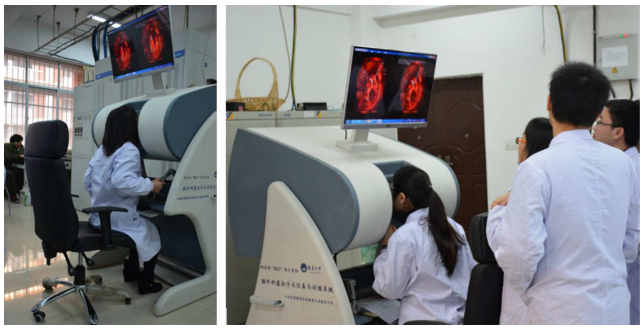


FIGURE 1. Virtual brain surgery system.

FIGURE 2. Virtual breast tumor resection system.

The rest of this paper is organized as follows. Section II describes a point-based texture mapping algorithm.

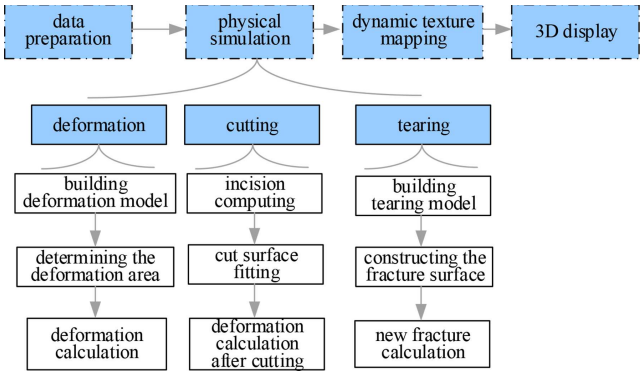


FIGURE 3. System functional module.

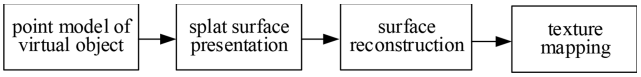


FIGURE 4. The process of point based texture mapping.

In Section III, a deformation model based on point primitive is presented, which has a higher computational efficiency compared with conventional mesh based methods. In Section IV, a soft tissue cutting method is presented. Section V describes a hybrid soft tissue tearing model, which can realize a real-time and realistic tearing simulation. Dynamic texture mapping algorithm and 3D display method are presented in Section VI and Section VII, respectively. Finally, the conclusions and future work are given in Section VIII.

II. POINT-BASED TEXTURE MAPPING

With the development of graphics hardware and 3D scanning techniques, point has become an important modeling and rendering primitive. Point-based surface representation and rendering techniques have recently received increased attention. In fact, points are the basic geometry defining elements of three-dimensional (3-D) objects and surfaces. And surface splatting is a commonly used method, which defines a disk-shape splat on the tangent plane of the model at every point and assigns the color of the point to the closest pixel. In our developed system, we select circular splats as rendering primitive. By controlling the sampling density and automatically adjusting the size of the circular splats, the surface of the simulated object can be seamlessly covered with a much small number of splats than points. In order to render highly realistic image, point based texture mapping method is proposed in our project, the process of which is shown in Fig.4.

A two-step strategy is employed in texture mapping. First, the relationship between the texture image and a spherical surface is established. Second, the relationship between the spherical surface and the splat surface is established to obtain the corresponding texture coordinates. Specifically, to obtain the relationship between a spherical surface and the texture image, the hemispherical geometric constraint mapping scheme is adopted in our project. The ratio of the area of

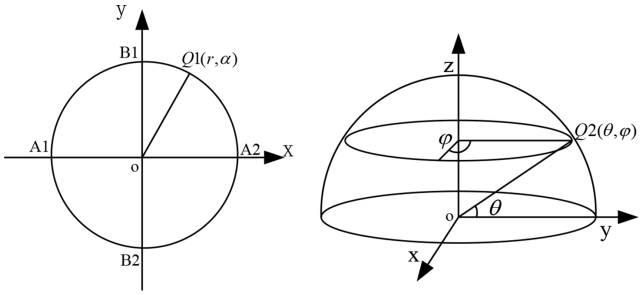


FIGURE 5. Mapping of texture images on hemispherical surfaces.

the texture image and that mapping in the sphere is dynamic change. By adjusting the ratio, the scope of sphere texture mapping can be uniformly controlled. As shown in Fig.5, polar coordinates of an arbitrary point Q_1 on the texture plane are set (r, α), Latitude and longitude coordinates of the corresponding projection point Q_2 on the surface of the hemisphere are set $Q_2(\theta, \varphi)$.

Any one of the circular arcs on texture image is mapped on a longitude of the hemisphere surface. According to the constraint of equal ratio of area, the area s_1/s_2 in texture coordinates should be a constant c which can be computed as

$$\frac{s_1}{s_2} = \frac{2\pi(1 - cos(\varphi))}{\pi r^2} = \frac{2(1 - cos(\varphi))}{r^2} = c \tag{1}$$

The whole circle texture area maps to the spherical cap which needs to satisfy the latitude is less than ϕ. The display effect is better when $0 < \phi \leq \pi/2$, and at the moment the boundary conditions are $r = 1$, $\varphi = \phi$, and the value of c is $2(1 - cos\phi)$. The relationship between texture coordinates and the half spherical coordinate can be obtained

$$u = rcos\alpha = rcos\theta = \frac{x}{\sqrt{1+z}} \frac{1}{\sqrt{1 - cos\phi}} \tag{2}$$

$$v = rsin\alpha = rsin\theta = \frac{y}{\sqrt{1+z}} \frac{1}{\sqrt{1 - cos\phi}} \tag{3}$$

To establish the relationship between a splat surface and a spherical surface, a hemisphere with appropriate radius is first selected to surround the splat surface of the virtual object, the central axis of the hemisphere is designed to pass through the center of the object. Then, the appropriate distance between the center of the sphere and the bottom of the object is determined according to the specific shape of the object. Finally, the texture image is mapped to the spherical dome by using area equal ratio constrained spherical texture mapping algorithm.

In our algorithm, a splat is defined as $Q(c_x, c_y, c_z)$, a fixed splat on the z axis is defined as $P(x_1, y_1, z_1)$, a sphere is expressed as $x^2 + y^2 + z^2 = r^2$, a line between the center of a splat Q and a fixed splat P can be expressed as

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} = k \tag{4}$$

Then, the straight line intersects with the sphere. So far, we can obtain the corresponding relationship between splats

and the sphere according to the value of k. k can be computed as

$$k = \frac{[-2x_1(c_x - x_1) - 2y_1(c_y - y_1) - 2z_1(c_z - z_1)] - \sqrt{mid}}{2[(c_x - x_1)^2 + (c_y - y_1)^2 + (c_z - z_1)^2]} \tag{5}$$

where *mid* can be computed as

$$\begin{aligned} mid = {} & [2x_1(c_x - x_1) + 2y_1(c_y - y_1) + 2z_1(c_z - z_1)]^2 \\ & - 4[(c_x - x_1)^2 + (c_y - y_1)^2 + (c_z - z_1)^2](z_1^2 - r^2) \end{aligned} \tag{6}$$

To simplify the calculation, P is set (0, 0, z), we can derive the relationship between splats and the point of the sphere, that is $x = k(c_x - x_1) + x_1$, $y = k(c_y - y_1) + y_1$, $z = k(c_z - z_1)$. We can obtain $x = kc_x$, $y = kc_y$, $z = kc_z$. At the same time, ϕ is set $\pi/2$. Texture coordinates of a splat isčž

$$\begin{cases} u = \dfrac{x}{\sqrt{1+z}} \\ v = \dfrac{y}{\sqrt{1+z}} \end{cases} \tag{7}$$

So far, texture mapping is well achieved and can meet the invariance principles.

III. DEFORMATION SIMULATION

The geometric models of virtual objects we used were made up of points, meshless model was adopted to calculate the deformation of soft tissue. Before the calculation of deformation, each of the points is assigned material properties (such as mass, size and density) according to a kernel function. The selected polynomial kernel is

$$w(r, h) = \begin{cases} \dfrac{315}{64\pi h^9}(h^2 - r^2)^3 & \text{if r<h} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where, r is the distance between points, and h is the radius of support domain. Then, the density of point can be computed as

$$\rho_i = \sum_j m_j w_{ij} \tag{9}$$

where, $w_{ij} = w(|x_j - x_i|, h_i)$. Points with mass and density are called phyxel. The volume is computed for phyxel i by $v_i = m_i/\rho_i$. For each phyxel i, the location is represented with x_i, the displacement is u_i, the strain is ε_i, the stress is σ_i and the body force is f_i.

For each time step $\triangle t$, the spatial derivatives ∇u_t can be computed according to displacement u_t of every phyxel. Then, the strain and stress are calculated, the elastic force can be obtained according to the strain energy. Finally, the new displacement $u_{t+\triangle t}$ is obtained by integrating over time. In order to calculate the strain, stress and elastic force, the spatial derivatives of the displacement field are approximated using the moving least squares method

$$\nabla u|_{x_i} = A^{-1}(\sum_j (u_j - u_i)x_{ij}w_{ij}) \tag{10}$$

where, u_i is displacement vector of phyxel i and u_j is displacement vector of nearby phyxels, which is computed using spatial hashing [22]. As well as $\nabla u|_{x_i} = (u_{,x}, u_{,y}, u_{,z})^T$, $x_{ij} = x_j - x_i$, and $A = \sum_j x_{ij}x_{ij}^T w_{ij}$. Where, $\nabla u|_{x_i}$ represents the spatial derivative of the displacement field at the phyxel's location x. u_x, u_y and u_z signify partial derivative of u with respect to x, y and z, respectively. A is a 3 by 3 moment matrix. $\nabla u|_{y_i}$ and $\nabla u|_{z_i}$ can be computed similarly. Green strain ε_i, stress σ_i at phyxel i can be computed as

$$\varepsilon_i = \frac{1}{2}(\nabla u + [\nabla u]^T + [\nabla u]^T \nabla u) \tag{11}$$

$$\sigma_i = E\varepsilon_i \tag{12}$$

where, E is a material constant matrix. The strain energy stored around phyxel i is estimated

$$U_i = v_i \frac{1}{2}(\varepsilon_i \cdot \sigma_i) \tag{13}$$

The forces acting at phyxel i and its neighbors j are computed

$$f_j = -\nabla_{(u_j)} U_i = -2v_i(I + \nabla u_i)\sigma_i d_j = Fd_j \tag{14}$$

$$f_i = -2v_i(I + \nabla u_i)\sigma_i d_i = Fd_i \tag{15}$$

where, $d_i = A^{-1}(-\sum x_{ij}w_{ij})$, $d_j = A^{-1}(x_{ij}w_{ij})$.

Finally, the velocity and the displacement of phyxel i at the next time step are computed and it is the similar for all other phyxels.

$$\dot{u}_i = \dot{u}_i + \triangle t f_i/m_i \tag{16}$$

$$u_i = u_i + \triangle t \dot{u}_i \tag{17}$$

In two different virtual surgery systems, the deformation of soft tissue is computed using the proposed meshless algorithm. The deformation effects of breast and tumor are shown in Fig.6 and Fig.7. Unlike other systems, there is a mapping between the deformation model and the rendering model because different data models are used for deformation calculation and rendering. The mapping is not required in our system because only point data is adopted in deformation computing and rendering, which improves computational efficiency and achieves a better visual effect.

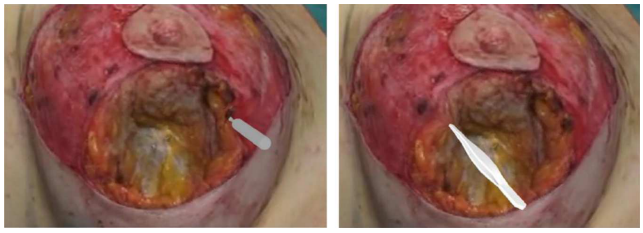


FIGURE 6. Deformation effect of the breast.

IV. CUTTING SIMULATION

In real cutting, the deformation of the soft tissue occurs continuously while the cutting force increases. when the cutting force that is applied to soft tissue exceeds a threshold,

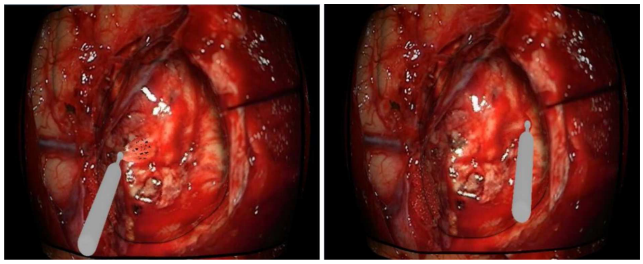


FIGURE 7. Deformation effect of the tumor.

soft tissue will create separation. Thus, two important steps need to be implemented to simulate soft tissue cutting. The first is rendering of cutting surface when the separation of soft tissue occur, the second is cutting calculation of soft tissue.

To render a realistic cutting surface, we use the collision points to fit the cutting plane. The equation of a plane can be expressed as

$$z = a_0 x + a_1 y + a_2 \tag{18}$$

Assuming there are n collision points $P_i(x_i, y_i, z_i)$, the cutting plane for these points should meet requirements

$$MinS = \sum_{i=0} (a_0 x_i + a_1 y_i + a_2 - z_i)^2 \tag{19}$$

$$\begin{cases} a_0 \sum x_i^2 + a_1 \sum x_i y_i + a_2 \sum x_i = \sum x_i z_i \\ a_0 \sum x_i y_i + a_1 \sum y_2 + a_2 \sum y_i = \sum y_i z_i \\ a_0 \sum x_i + a_1 \sum y_i + a_2 n = \sum z_i \end{cases} \tag{20}$$

By solving the equation (20), we can obtain the coefficients a_0, a_1, a_2 and then obtain the cutting plane.

After the cutting plane is ascertained, the displacement of the points that is caused by cutting force need to be computed. The process of cutting includes two stages: separation and deformation. When the force applied on the soft tissue exceeds the threshold of cutting force, the soft tissue will be separated. Then, the deformation of the soft tissue depends on the force F_p. The cutting force is decomposed into two parts: the part within the cutting plane (F_l) and the other part perpendicular (F_p). To improve the accuracy of simulation, we introduce the velocity v and contact angle θ to model this behavior. The F_l and F_p can be expressed as

$$F_l = F_x sin\theta v \tag{21}$$

$$F_P = F_x cos\theta v \tag{22}$$

where F_x is the external force applied to the soft tissue. After being cut open, the soft tissue deforms under the forces imposed by the vertical force F_p. Since all the points are independent, we use the model introduced in Section III to compute the deformation caused by cutting. According to equation (14), the loading force F_p can be represented

$$F_P(t + \triangle t) = -v_i \sigma_i (t + \triangle t) \nabla u_i \varepsilon_i (t + \triangle t) \tag{23}$$

As mentioned above, we use a leap frog scheme to calculate the displacement of the deformation caused by cutting. The render effect of breast cutting is shown in Fig.8.

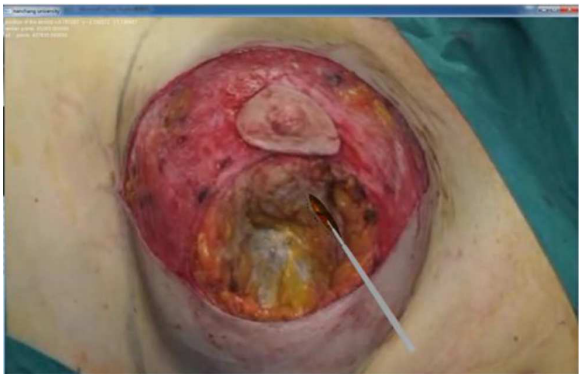


FIGURE 8. The rendering effect of breast cutting.

In addition, the cutting forces feedback f_c can be computed as

$$f_c = f_l + f_p \tag{24}$$

where f_l and f_p are the frictions caused by F_l and F_p, respectively. f_l and f_p can be calculated as

$$f_l = \mu F_l d \tag{25}$$

$$f_p = \mu F_p \tag{26}$$

where μ is the coefficient of the friction force, and d is the penetration depth of the scalpel. In general mesh based methods, the cut units need to be reorganized at each step. Our method does not need to do so, which greatly improves computational efficiency. In our system, the cutting force update rate can reach about 450 Hz. while haptic rendering requires a high update rate of 1000 Hz for virtual objects to achieve real-time simulation. Similar to [28], the missing force data are generated using extrapolation.

V. TEARING OF SOFT TISSUE

The suction of soft tissue in neurosurgery is a common operation. In fact, the whole process is the fracture process of soft tissue. At present, object breaking process simulation methods are roughly divided into two categories: physically based methods and geometrically based methods. Physically based methods achieve the object fracture simulation by establishing the mechanical model with the continuous medium theory, such as the spring mass model [29], the finite element model [30], the meshless local Petrov-Galerkin [31], and the boundary element method [32] etc. We can get accurate simulation results and vivid animation by using physically based methods. However, the efficiency of these methods is low. Geometrically based methods achieve fracture simulation by setting a fixed pattern to simulate broken phenomenon under specific conditions. As shown in Fig.9, fragments are designed in advance. They can rupture according to the scheduled plan when collision occurs. Although geometrically based methods have high computational efficiency, their simulation results are mechanical and lack of authenticity.

To achieve a realistic and real-time tearing simulation effect, a hybrid method based on physical and geometric

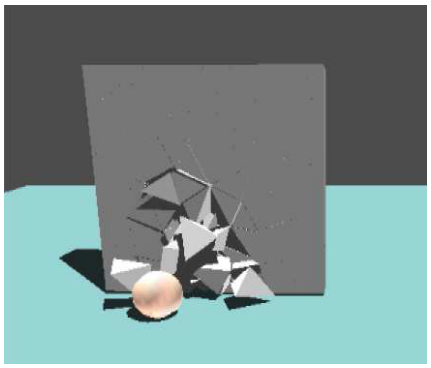


FIGURE 9. The simulation effect of breaking based on geometrical method.

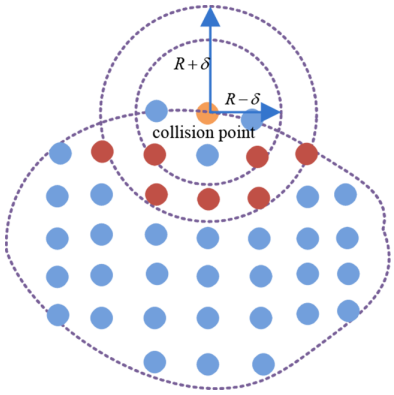


FIGURE 10. Determination of fracture surface points.

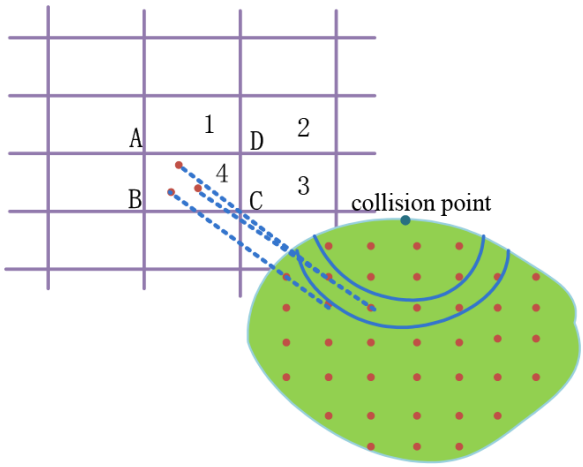


FIGURE 11. The projection of points on the fracture surface.

method is proposed to simulate the fracture process of soft tissue. First, the meshless method is adopted to establish the physical model for soft tissue, so as to determine the breaking point; then, the fracture surface is determined according to the distance between the collision point and the breaking point; finally, the fracture surface is rendered and carried out noise interference. Details are as follows:

A. THE FORMING OF FRACTURE SURFACE

In the process of real operation, a fracture shape of soft tissue after suction is similar to spherical shape. In order to determine spherical shape, the center and radius need to be computed. In our project, we set the collision point as center. To determine the radius, the initial rupture points need to be found. Specific steps are as follows:

Step1: When the soft tissue is stressed, meshless model is built for soft tissue, the stress value of the each individual element points σ is calculated according to the equation (12). Then, the eigenvalue of stress tensor is computed and the maximum characteristic value is compared with a preset threshold, voxel points which satisfy the maximum eigenvalue is greater than the threshold stress are deposited to the collection F.

Step2: Surface points are selected from the collection F and the distance between surface points and collision point is computed. Then, the surface point which has the maximum distance is selected as the breaking point, and the maximum distance is defined as fracture radius R. Finally, a sphere is obtained with collision point as its center and R as its radius.

Step3: The volume data which intersects the sphere will be exposed as the fracture surface. To avoid sparse exposed point could not form a surface, points are included to fracture surface which satisfy the distance to collision point is within $R \pm \delta$, where δ is a error range. As shown in Fig.8, red dots are exposed and formed new fracture surface.

B. THE RENDERING OF FRACTURE SURFACE

Interior points are exposed to form the fracture surface. In order to render the new rupture surface, the normal vectors of these points need to be computed. Many algorithms are proposed to solve the normal vectors of discrete points, such as algorithms proposed in the literature [33], [34]. However, these methods are very time-consuming. As well as sparse distribution of interior points could lead to holes. To avoid this situation, the Height Field is employed in our project, details are as follows.

First, a 2D grid is defined as projective planes. Specifically, all points of soft tissue model are traversed and the maximum and minimum values in the X and Y direction are found. Furthermore, the size of each grid is determined according to the number of grids you need. Then, a two-dimensional array is used to store the height of each grid of two-dimensional grid. The initial value of every element in the array is assigned to the minimum value in the Z direction of each grid.

To render fracture surface, surface points are first projected in the corresponding grid and the maximum Z of all the projection points is assigned to the height value of corresponding grid and the height of the point at the lower left corner of the grid. Finally, the two diagonal triangles in each grid are rendered. As shown in Fig. 11, three points on fracture surface are projected to the grid 4, the biggest Z value is assigned to the height values of the point B. Similarly, the height value of the point A is maximum Z value of all the projection points

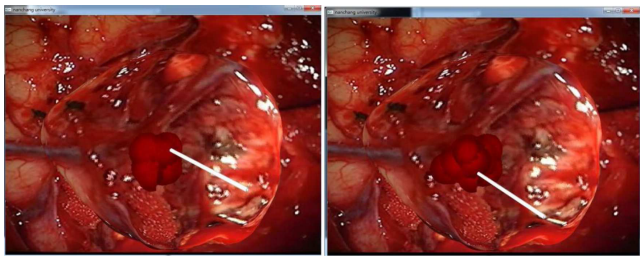


FIGURE 12. The rendering effect of a fracture surface.

in the grid 1; the height values of the point D and point C are maximum Z value of all the projection points in the grid 2 and grid 3. In grid 4, the normal vectors of triangle ABD and triangle BCD can be calculated according to coordinate values of four points A, B, C and D. Each grid is computed in the same way. Two triangles in each grid are rendered to ensure that no holes are on the fracture surface. The rendering effect of a fracture surface is shown in Fig. 12.

C. THE DISTURBING OF NOISE

As can be seen from the Fig. 12, the rendering effect of the fracture surface is smooth and is not realistic, perlin noise is used to interfere fracture surface in our project. Specifically, discrete points of fracture are first disturbed. Then, the disturbed points are projected to a two-dimensional plane and rendered with height field method. To disturb a plane with perlin noise, original frequency and original amplitude of noise are first set. Then, sample points of rows and columns of the model are selected based on the frequency and a random number is generated and assigned to the corresponding point with the pseudo-random number generator.

In our project, a square is used as noise plane. We first define the center point of the plane as $(ro/2, co/2)$, where ro is the number of rows, co is the number of columns. Then, the function values of these points are set 0 if they are greater than $0.9R$ away from the center point. Finally, to avoid producing discontinuous surface, the circle function value reduction method is adopted, which makes function value of points in a circle decrease with distance to center point increase.

$$f(x, y) = f(x, y)(1 - (d - R_i)/(R_o - R_i)) \tag{27}$$

where d is the distance between point (x, y) and center point, R_i and R_o are the radius of inter ring and outer ring, respectively. The radius of circle range is set $[0.5R, 0.9R]$, the rendering effect of plane disturbed by noise is shown in Fig. 13.

After interference plane is established, new fracture surface can be disturbed. Specifically, the values of x and y are projected on the interference plane, respectively and the corresponding point (x, y) on the interference plane is found. Then, z value of the point in the original model is overlaid to the corresponding function value of the point (x, y) on the

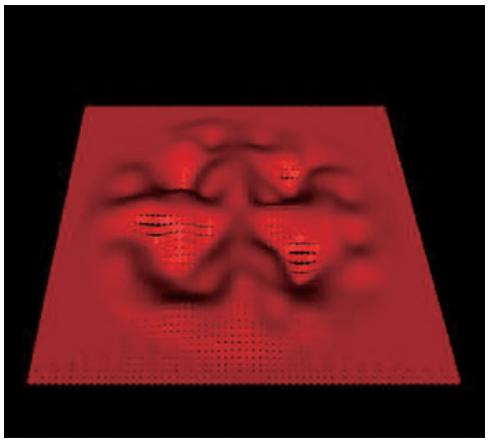


FIGURE 13. Rendering effect of plane disturbed by noise.

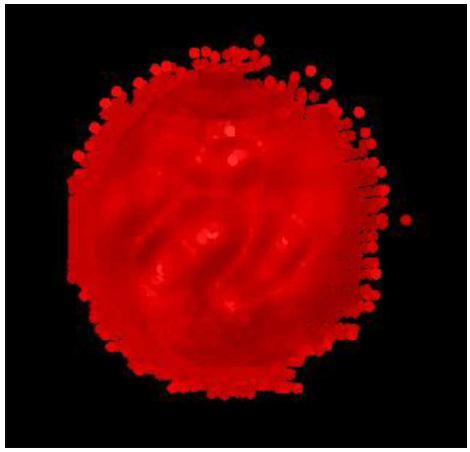


FIGURE 14. A interfered fracture surface.

interference plane. Mapping function are:

$$X(x) = min_x^p + (x - min_x)\frac{max_x^p - min_x^p}{max_x - min_x} \tag{28}$$

$$Y(y) = min_y^p + (y - min_y)\frac{max_y^p - min_y^p}{max_y - min_y} \tag{29}$$

where, min_x^p and min_y^p are the lower bounds of x axis and y axis on the interference plane, respectively. max_x^p and max_y^p are the upper bounds of x axis and y axis on the interference plane, respectively. min_x and min_y are the lower bounds of x axis and y axis on the fracture surface, respectively. max_x and max_y are the upper bounds of x axis and y axis on the fracture surface, respectively. Overlay function is defined as

$$Z_{DP} = z_{DP} + f(X, Y) \tag{30}$$

where, z_{DP} is z of every point on the fracture surface. $f(X, Y)$ is the function value of point (X, Y) on the interfere plane. The local rendering effect of a interfered fracture surface is shown in Fig.15.

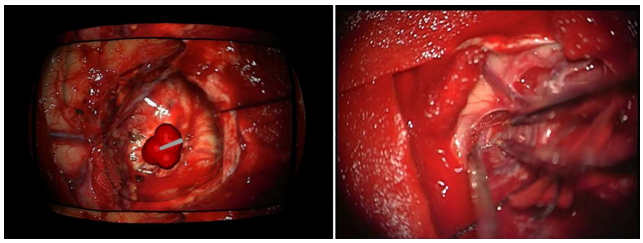


FIGURE 15. The rendering effect of our system (left) and real scenarios (right).

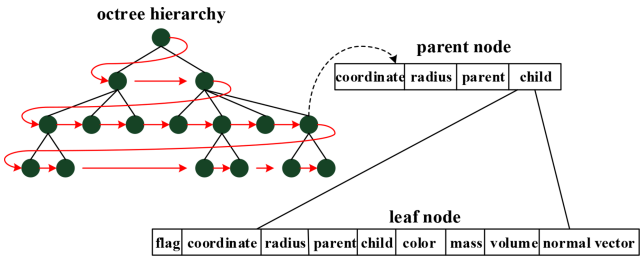


FIGURE 16. Hierarchical octree.

The fracture model is applied to the neurosurgery tumor excision surgery simulation system. The rendering effect of our system and real scenarios is shown in Fig.15.

D. THE UPDATE OF DATA

In virtual surgery simulation, the update of the data model is very important. When the topological structure of the soft tissue changes, real-time updating data can ensure the correctness of the next collision detection, and effective data management is the assurance of real-time updating data.

To achieve effective data management in our system, the surface model of soft tissue is first subdivided by using octree space subdivision algorithm to obtain interior points. Then, similar to Qsplat [35], every surface point and every interior point are stored as a sphere. Finally, every voxel is defined as a struct, which contains a variety of properties (flag, coordinate, radius, child pointer, parent pointer, color, mass, volume, normal vector). We can distinguish surface points from interior points by the value of the flag. As shown in Fig. 16, to get a quick access to each individual element, we construct a hierarchical octree for spheres. Each individual element of the soft tissue is a leaf node of this octree.

When fracture occurs, the topology structure of soft tissue needs to be updated, updating process of an octree is shown in Fig. 17. First, the stress and the characteristic values of stress of each point are calculated by meshless deformation model, stress maximum eigenvalue of each points is contrasted with the preset threshold so that fracture points are determined. Then, we can compute fracture radius according to the distance between the breaking point and collision point. Finally, according to the collision point's parent node to determine fracture radius size range. Rupture range can be also judged according to the parent node of the collision point.

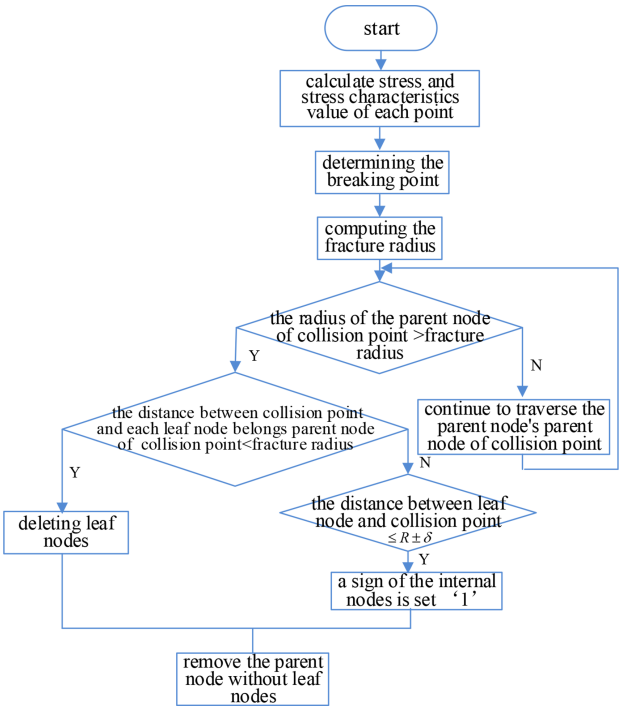


FIGURE 17. The update process of octree.

For all the leaf nodes of the parent node of the collision point, they will be deleted if the radius of the parent node of the collision point is greater than the fracture radius, and their parent node will also removed if leaf nodes have been deleted. If the distance between the leaf node and the collision point is less $R \pm \delta$, a sign of the interior nodes attribute value is set '1' and surface node remain unchanged. If the radius of the parent node of the collision point is less than the fracture radius, the parent node of the parent node of the collision point will be continued traversing until the radius of the parent node is greater than the fracture radius, and then continue with the above steps. At each time step, only surface points of updated octree are rendered and collision detection is carried on the whole tree.

VI. DYNAMIC TEXTURE MAPPING

At present, many universities and scientific research institutions have successively developed virtual surgical simulation systems. Nevertheless, most systems only display virtual lesion organs without background, or use simple static background images. Although virtual tissues and organs are precisely simulated in these virtual reality systems, they are not fused with the real environment, which leads to a low sense of reality and immersion.

To enhance the realism and immersion of the simulation system, the brain surgery video is used as a dynamic texture and environment mapping technology is adopted to set up highly realistic virtual surgery scenes. The method is divided into three steps, the details are as follows:

Step 1: preprocessing the video

Video frame image U is segmented to get spot area image U_s and the region image of brain tissue U_b, where $U = U_s + U_b$. We defined the variance between two area

$$\delta^2(k) = \frac{(P_s(k)g_s - g(k))^2}{P_s(k)P_b(k)} \tag{31}$$

where k is segmentation threshold, $P_s(k)$ is the probability of a pixel belongs to area U_s, $P_b(k)$ is the probability of a pixel belongs to area U_b, g_s is gray average of U_s and $g(k)$ is the cumulative gray value. We can segment U_s and U_b according to k.

We magnify Us by 1.1 times to get Us' images with bilinear interpolation. Specifically, the origin image is first set $M * N$ and the target image is $1.1M * 1.1N$. The gray value of point $p(i, j)$ in Us' is computed with neighbor mean summation. Then, spot area registration is achieved by computing and matching center of mass of Us and Us'. Finally, we can obtain complete brain tissue image Ub' by repairing the area which is removed the spot. Further, each frame image of video is performed weighted sum to get the pixel value of each point in the repaired area in frame k

$$f_k(i, j) = \sum_{s=1}^{frames} W_s f_s(i, j) \tag{32}$$

where, W_s is weight of each frame image.

Step 2: dynamic texture mapping

video stream is first extracted from video and dispersed into finite frame sequences according to video duration. Then, the frame image is processed into a texture format that can be used for texture mapping according to frame information. Finally, the video is mapped to brain tissue areas as a dynamic texture. The image difference caused by the end point image hopping is dispersed to the whole cycle so as to eliminate the visual jumpy. Meanwhile, the frame image is divided into grids and grids with complete trajectories are extracted. To ensure that the initial grids complete the movement, the interrupt defect grids at video stop point are replaced by grids with a complete trajectory.

Step 3: Mirror environment texture mapping for the brain tumor

The surrounding tissue environment image is mapped to the brain tumor surface as a texture by adopting cube mirror environment mapping, which can ensure the texture of the brain tumor is homologous with that of brain tissue. Specially, a cube is first built. Then, a square texture in the middle of the texture plane is covered on the top surface of the cube. Finally, the annular region outside of the square is covered by the remaining sides of the cube. The mapping formula for point $P(u, v)$ on the texture plane is mapped to point $Q(x, y, z)$ on the the cube is

$$u = \begin{cases} \dfrac{x}{|x|}\sqrt{\dfrac{KD^2}{4} + \dfrac{D^2}{2K} - \dfrac{D}{K}z} & |x| = \dfrac{D}{2} \\ x\sqrt{K + \dfrac{2}{K} - \dfrac{4}{KD}z} & |y| = \dfrac{D}{2} \end{cases} \tag{33}$$

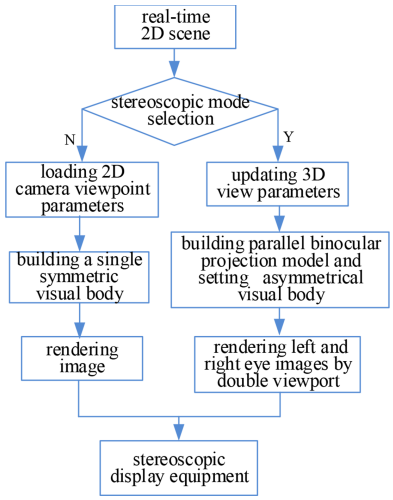


FIGURE 18. The implementation process of 3d display.

where the ratio of the area of before texture mapping and after texture mapping is K. The side length of cube is D.

$$v = \begin{cases} \dfrac{y}{|y|}\sqrt{\dfrac{KD^2}{4} + \dfrac{D^2}{2K} - \dfrac{D}{K}z} & |y| = \dfrac{D}{2} \\ y\sqrt{K + \dfrac{2}{K} - \dfrac{4}{KD}z} & |x| = \dfrac{D}{2} \end{cases} \tag{34}$$

We adopted dynamic texture and mirror environment mapping to achieve high fidelity rendering effect, which greatly improved authenticity and immersion of the surgery simulation. Compared with other existing virtual surgery simulation systems, trainee doctors can experience a more realistic scenarios in our system, which can help them get more fast and efficient promotion of the operation level.

VII. 3D DISPLAY

Stereo display is an important technology in the field of virtual reality. Broad perspective of binocular stereo vision solutions is proposed to achieve stereo display in our project. Unlike the traditional display technology, the proposed method can completely cover the eye's field of view and avoid producing dark background. The technology can reproduce a realistic operation scene so as to let users obtain better training effect.

The implementation process of 3D display is shown in Fig.18. A monocular two-dimensional scene is real-time rendered and two display modes are provided for users to choose. If a two-dimensional mode is selected, we first set the interface parameters of a single camera and establish a single symmetrical view body according to a mathematical modeling matrix. Then, a single viewport is set to render the image. Finally, the image is rendered and displayed on the screen by exchanging render buffer. If a stereo display mode is selected, we first set up 3D scene parameters of parallel binocular projection model and build this model with OpenGL. Then, eyes position and visual parameters

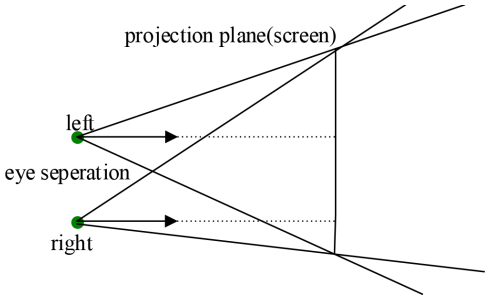


FIGURE 19. Parallel binocular projection model.

are set by using asymmetric visual body. Finally, image is rendered and displayed on the stereo equipment by a viewport transformation.

Parallel binocular projection model is shown in Fig. 19, two camera view are in the same direction and the line of sight is in the same plane which avoids vertical parallax. Left and right eye image exists only horizontal parallax which makes pair image fusion more easy. Parallel binocular projection model can provide more standard parallax three-dimensional images to show higher quality images, ultimately provide better visual effect and could not make the observer's eye discomfort.

In our system, a passive 3D display screen is located on the near cutting plane, that is, the projected plane is the near cutting plane in the viewing cone. The realization of parallel binocular vision projection model is shown in Fig.20, *IPD* is eyes distance, *aperture* is the angle of viewing field of the camera. They can be assigned empirical values. *Top*, *bottom*, *left* and *right* (they express the distance from the top, bottom, left and right section to the eye, respectively) are all computed according to parameter values of the camera. Asymmetric view frustum for left and right cameras can be set by using function interface glFrusturm( ) to achieve parallel binocular projection model. The origin image of our virtual surgery system and left and right eye image are shown in Fig.21. The final image rendering can show the characteristics of the surgical scene and achieve the expected effect.

VIII. CONCLUSION

A virtual surgery framework based on point primitives is presented in this paper. The data models of virtual objects in this framework are made of points without topological connectivity. Whether virtual surgery scene rendering or mechanical calculation of soft tissue is based on point primitives, which avoids unlifelike surgery scene and improves computational efficiency duo to omitting to update the information of triangulation at each time step. To embody the superiority of this framework, two virtual surgery systems based on point primitives developed by our team were exhibited in this paper. The important functional modules of two systems were elaborated by including working principle, execution process, and the performance of the algorithm. Experiments have shown a high degree of visual realism in surgery simulation is

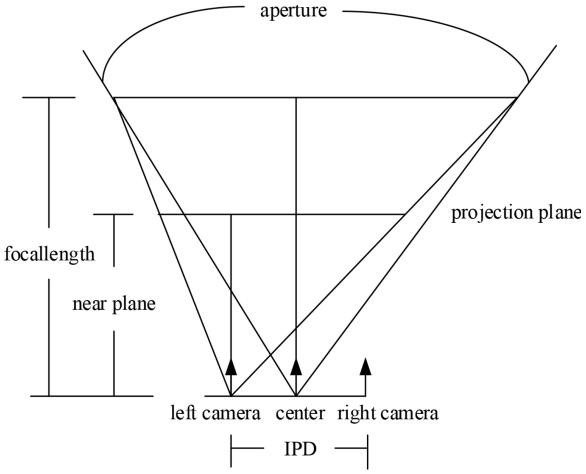


FIGURE 20. Realization of parallel binocular vision projection model.

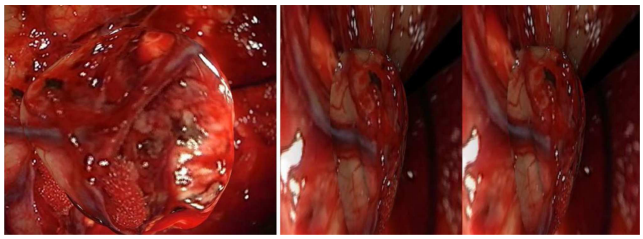


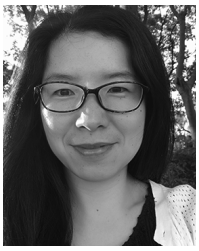
FIGURE 21. Origin image (left) and left and right eye image (right).

achieved. In future, the evaluation module will be considered to integrate into the system framework.

REFERENCES

[1] M. A. Burgos, E. Sanmiguel-Rojas, N. Singh, and F. Esteban-Ortega, "DigBody®:A new 3D modeling tool for Masal virtual surgery," *Comput. Biol. Med.*, vol. 98, no. 1, pp. 118–125, Jul. 2018.
[2] P. Ashkan, Z. Naghmeh, and G. David, "Mechanical model of orthopaedic drilling for augmented-haptics-based training," *Comput. Biol. Med.*, vo. 89, pp. 256–263, Oct. 2017. doi: 10.1016/j.compbiomed.2017.06.021.
[3] Y. Zou and P. X. Liu, "A new deformation simulation algorithm for elastic-plastic objects based on splat primitives," *Comput. Biol. Med.*, vol. 83, pp. 84–93, Apr. 2017.
[4] L. Glondu, M. Marchal, and G. Dumont, "Real-time simulation of brittle fracture using modal analysis," *IEEE Trans. Visualizat. Comput. Graph.*, vol. 19, no. 2, pp. 201–209, Feb. 2013.
[5] Y. Zou, P. X. Liu, Q. Cheng, P. Lai, and C. Li, "A new deformation model of biological tissue for surgery simulation," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3494–3503, Nov. 2017.
[6] B. Bickel, M. Otaduy, W. Matusik, H. Pfister, and M. Gross, "Capture and modeling of non-linear heterogeneous soft tissue," *ACM Trans. Graph.*, vol. 28, no. 3, p. 89, Jul. 2009.
[7] Y. Zou and P. X. Liu, "A high-resolution model for soft tissue deformation based on point primitives," *Comput. Methods Programs Biomed.*, vol. 148, pp. 113–121, Sep. 2017.
[8] S. Suzuki, N. Suzuki, A. Hattori, A. Uchiyama, and S. Kobayashi, "Sphere-filled organ model for virtual surgery system," *IEEE Trans. Med. Imag.*, vol. 23, no. 6, pp. 714–722, Jun. 2004.
[9] J. Brown, S. Sorkin, C. Bruyns, J. C. Latombe, K. Montgomery, and M. Stephanides, "Real-time simulation of deformable objects: Tools and application," in *Proc. Comput. Animation 14th Conf. Comput. Animation*, 2001, pp. 228–258.
[10] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, and S. Weghorst, "Real-time finite element modeling for surgery simulation: An application to virtual suturing," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 3, pp. 314–325, May/Jun. 2004.

[11] M. Marchal, J. Allard, C. Duriez, and S. Cotin, "Towards a Framework for Assessing Deformable Models in Medical Simulation," in *Proc. Int. Symp. Biomed. Simul.*, 2008, pp. 176–184.
[12] H. Courtecuisse, H. Jung, J. Allard, C. Duriez, D. Lee, and S. Cotin, "GPU-based real-time soft tissue deformation with cutting and haptic feedback," *Progr. Biophys. Mol. Biol.*, vol. 103, nos. 2–3, pp. 159–168, Dec. 2010.
[13] S. Delorme, D. Laroche, R. Diraddo, and R. D. Maestro, "NeuroTouch: A physics-based virtual simulator for cranial microneurosurgery training," *Neurosurgery*, vol. 71, pp. 32–42, Jan. 2012.
[14] K. S. Choi, "Interactive cutting of deformable objects using force propagation approach and digital design analogy," *Comput. Graph.*, vol. 30, no. 2, pp. 233–243, Apr. 2006.
[15] G. San-Vicente, I. Aguinaga, and J. T. Celigueta, "Cubical mass-spring model design based on a tensile deformation test and nonlinear material model," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 2, pp. 228–241, Feb. 2012.
[16] C. E. Etheredge, *A Parallel Mass-Spring Model for Soft Tissue Simulation With Haptic Rendering in CUDA*. Enschede, The Netherlands: Univ. Twente, 2011.
[17] P. E. Hammer, M. S. Sacks, P. J. D. Nido, and R. D. Howe, "Mass-spring model for simulation of heart valve tissue mechanical behavior," *Ann. Biomed. Eng.*, vol. 39, no. 6, pp. 1668–1679, Jun. 2011.
[18] J. Qin, W. M. Pang, Y. P. Chui, T. T. Wong, and P. A. Heng, "A novel modeling framework for multilayered soft tissue deformation in virtual orthopedic surgery," *J. Med. Syst.*, vol. 34, no. 3, pp. 261–271, Jun. 2010.
[19] Z. A. Taylor, M. Cheng, and S. Ourselin, "High-speed nonlinear finite element analysis for surgical simulation using graphics processing units," *IEEE Trans. Med. Imag.*, vol. 27, no. 5, pp. 650–663, May 2008.
[20] S. A. Cover, N. F. Ezquerra, J. F. O'Brien, R. Rowe, T. Gadacz, and E. Palm, "Interactively deformable models for surgery simulation," *IEEE Comput. Graph. Appl.*, vol. 13, no. 6, pp. 68–75, Nov. 1993.
[21] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Trans. Visualizat. Comput. Graph.*, vol. 5, no. 1, pp. 62–73, Jan./Mar. 1999.
[22] G. Picinbono, H. Delingette, and N. Ayache, "Non-linear anisotropic elasticity for real-time surgery simulation," *Graph. Models*, vol. 65, no. 5, pp. 305–321, Sep. 2003.
[23] F. Faure *et al.*, "Sofa:A multi-model framework for interactive physical simulation," *Stud. Mechanobiol. Tissue Eng. Biomater.*, vol. 11, pp. 283–321, Feb. 2012.
[24] A. Horton, A. Wittek, G. R. Joldes, and K. Miller, "A meshless total lagrangian explicit dynamics algorithm for surgical simulation," *Int. J. Numer. Methods Biomed. Eng.*, vol. 26, no. 8, pp. 977–998, Aug. 2010.
[25] G. R. Joldes, A. Wittek, and K. Miller, "Stable time step estimates for mesh-free particle methods," *Int. J. Numer. Methods Eng.*, vol. 91, no. 4, pp. 450–456, Jul. 2012.
[26] X. Jin, G. R. Joldes, K. Miller, K. H. Yang, and A. Wittek, "Meshless algorithm for soft tissue cutting in surgical simulation," *Comput. Methods Biomech. Biomed. Eng.*, vol. 17, no. 7, pp. 800–811, May 2014.
[27] J. Belinha, R. M. N. Jorge, and L. M. J. S. Dinis, "Bone tissue remodelling analysis considering a radial point interpolator meshless method," *Eng. Anal. Boundary Elements*, vol. 36, no. 11, pp. 1660–1670, Nov. 2012.
[28] G. Picinbono, J. C. Lombardo, H. Delingette, and N. Ayache, "Improving realism of a surgery simulator Linear anisotropic elasticity, complex interactions and force extrapolation," *Comput. Animation Virtual Worlds*, vol. 13, no. 3, pp. 147–167, 2010.
[29] A. Norton, "Animation of fracture by physical modeling," *Vis. Comput. Int. J. Comput. Graph.*, vol. 7, no. 4, pp. 210–219, Jun. 1991.
[30] J. F. O'brien and J. K. Hodgins, "Graphical modeling and animation of brittle fracture," IN *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, Jun. 1999, pp. 137–146.
[31] N. Liu, X. He, S. Li, and G. Wang, "Meshless simulation of brittle fracture," *Comput. Animation Virtual Worlds*, vol. 22, nos. 2–3, pp. 115–124, Apr. 2011.
[32] D. Hahn and C. Wojtan, "High-resolution brittle fracture simulation with boundary elements," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 151:1–151:12, 2015.
[33] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," *Int. J. Comput. Geometry Appl.*, vol. 14, no. 4, pp. 261–276, 2008.
[34] N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proc. 25th Annu. Conf. Comput. Graph. Interact. Techn.*, Sep. 1998, pp. 415–421.
[35] S. Rusinkiewicz and M. Levoy, "QSplat: A multiresolution point rendering system for large meshes," in *Proc 27th Annu. Conf. Comput. Graph. Interact. Techn.*, Jul. 2000, pp. 343–352.



YANNI ZOU received the Ph.D. degree from Nanchang University, Nanchang, China, in 2016. She has been with the School of Information Engineering, Nanchang University, since 2017.

PETER X. LIU received the B.Sc. and M.Sc. degrees from Northern Jiaotong University, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Alberta, Canada, in 2002.

He has been with the Department of Systems and Computer Engineering, Carleton University, Canada, since 2002. He is currently a Professor and the Canada Research Chair. He is also with the School of Information Engineering, Nanchang University, Nanchang, China, as an Adjunct Professor. He has published more than 280 research articles. His interests include interactive networked systems and teleoperation, haptics, micro-manipulation, robotics, intelligent systems, context-aware intelligent networks, and their applications to biomedical engineering.

Dr. Liu is a Licensed Member of the Professional Engineers of Ontario (P.Eng.) and a Fellow of the Engineering Institute of Canada (FEIC). He was a recipient of the 2007 Carleton Research Achievement Award, the 2006 Province of Ontario Early Researcher Award, the 2006 Carty Research Fellowship, and the 2003 Province of Ontario Distinguished Researcher Award. He received the Best Conference Paper Award from the 2006 IEEE International Conference on Mechatronics and Automation. He serves as an Associate Editor for several journals including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE/ASME TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE ACCESS.

DEDAO WU received the M.S. degree in computer science from Nanchang University, Nanchang, China, in 2006, where he is currently pursuing the Ph.D. degree with the School of Information Engineering. His research interests include image processing, computer vision, and pattern recognition.

XIAOHUI YANG received the Ph.D. degree from Nanchang University, Nanchang, China, in 2015. He has been with the Department of Automatic Control, School of Information Engineering, Nanchang University, since 2006, where he is currently an Associate Professor. His current interests include haptics, robotics, and computing intelligence.

SHAOPING XU received the M.S. degree in computer application from the China University of Geosciences, Wuhan, China, in 2004, and the Ph.D. degree in mechatronics engineering from the University of Nanchang, Nanchang, China, in 2010. He is currently a Professor with the Department of Computer Science and Technology, School of Information Engineering, Nanchang University, Nanchang. His current research interests include digital image processing and analysis, computer graphics, and virtual reality.