

Real-time deformation and cutting simulation of cornea using point based method

Yanjun Peng¹ · Qiaoling Li¹ · Yingying Yan¹ ·
Qiong Wang²

Received: 16 September 2017 / Revised: 4 March 2018 / Accepted: 29 June 2018 /
Published online: 6 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract This paper proposed an improved point based method to simulate the deformation and cutting of cornea, realized the real-time interaction between the force feedback device and the model in virtual surgery. We construct a hybrid deformable model, the model consists of two parts: the interior of the cornea consists of some particles, and the boundary consists of some meshes which are formed by surfaces nodes, calculate the stress tensor of particle can simulate corneal deformation. In this process, in order to ensure the stability of the deformation, a volume constraint is added to our model. By sampling on the surface of the model to form some meshes, the model could support the changes of topology. Using a node replication method for surface cutting and subdivide the cutting triangles in the cutting process, so that it can generate relatively smooth incision. Experiments show that the model provides a real and efficient deformation simulation, with good stability and scalability.

Keywords Point based method · Real-time interaction · Virtual surgery · Deformation simulation

1 Introduction

Experimental animals are always used in the surgical training and teachings, but there are always some slight differences between the structure of animals and humans, and these human specimens are usually costly and unrepeatable. The advent of virtual surgery is a good solution to these problems. Virtual surgery is an important

✉ Yanjun Peng
pengyanjuncn@163.com

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

application of virtual reality technology in the medical field, by using medical imaging data and virtual reality technology to establish a virtual environment in the computer, and the doctors interact with the virtual environment through interactive devices to carry out the surgical training and exploration. The virtual surgery system can be used to surgical training or teaching exercises, the interns can improve their level of proficiency and the success rate of surgery. At the same time, the system can be reused, so it can greatly reduce the expenditure. Our system can simulate the operation of the cornea in surgery, which will be helpful to the research and development of eye surgery such as cataract.

The simulation objects of virtual surgery are the three-dimensional soft tissue, and the main research purposes is to simulate the deformation and cutting simulation of soft tissue. Traditional simulation method includes mass spring method and Finite Element Method [1] (FEM). But the mass spring method is easy to produce jitter, reducing the stability and authenticity of the system. The calculation of FEM is large, and it is difficult to meet the requirements of real-time. The meshfree method [6] is a new method to simulate the deformation of objects in recent years, it does not need any meshes in the calculation process, but uses the particles with physical attributes as computational unit. The meshfree method is much less computationally than FEM, the real-time and operability are higher than the mass spring method. However, meshfree method has no grids, so, generated some meshes in the process of cutting can form a new incision, and can form a cutting surface for the soft tissue with large volume if have a further research. It is necessary to improve the model in the study of the soft tissue cutting deformation.

M. Müller et al. [14] proposed a point based method to simulate the deformation of elastic-plastic and melt objects. But the model does not apply to crack propagation or simulation cutting because of the interaction between particles. In order to simulate the cutting process, this paper presents an improved point based dynamic model which allows the topology to be changed. In the internal of the deformed object, we distributed some discrete particles with properties, and covered some meshes on the surface of model, then added volume conservation to ensure the stability under the large deformation. BVH bounding box can be used to quickly find the collision intersection point, and improved the speed of interactive between the force feedback device and the cornea model. The effect of the algorithm is demonstrated by the experimental results.

2 Related work

Virtual surgery has developed rapidly in recent years, and the methods used are also different. Divided deformable models into two categories: physics-based and none physics-based, namely geometric models. Physical-based methods are based on continuum mechanics, and could get accurate simulation results by directly solving the partial differential equations (PDEs) using numerical methods. Some of the prevailing methods include FEM, boundary element method (BEM) [18, 25], point-based method [7, 19] and position based dynamics [16, 23], SPH etc. None physics-based models use intuitive methods instead of PDE. For example, the mass spring model uses point

masses connected by a network of springs to represent continuous material, and meshfree shape matching model [15] computes deformations based on geometry shapes.

In this paper we explore the deformation based on physics-based method. The physical model takes into account various characteristics of objects, including stiffness, viscoelasticity, anisotropy etc., it can more accurately describe the interaction between soft tissue and surgical instruments. FEM is a kind of physical modeling, it uses finite element to simulate the properties of internal biomechanical of the tissue, and it is easy to handle but the calculation is large. Martin W et al. [24] proposed a finite element method based on convex polyhedral to simulate the deformation of the objects. In their method the splitting elements do not need to be reconstructed, therefore can reduce the number of nodes and elements in the process of cutting deformation which is more flexible than using the tetrahedron only. Youquan Liu et al. [12] use the GPU to accelerate the finite element method, and its speed is four times faster than it used to be.

Mass spring model is also one of the physical model. One method developed the topology change from 2D to 3D, and using a new method to determine the mesh topology and spring stiffness parameters at once [2]. Min Hong et al. [8] used surface meshes to maintain in the integrity of the material and to simulate the deformation of the soft tissues. Andrew Selle et al. [20] used a mass spring model to simulate the forces between each hair by fabric bending and hair twisting, they realized the bent of the cloth and the torsion of the hair.

SPH method is one of meshless method which is gradually developed in nearly 20 years, its basic principle is below: the deformation object is formed by some particles, these particles have their attributes such as mass, density and volume, regarded these particles as the basic unit of computing. Making some improvement on SPH method, and it could be used to solve the solid deformation issue, SPH method has become more and more widely used in deformation problem [13, 21, 22]. Position based dynamics method is also one of the physic-based methods, it is a meshfree method and used common as well. Mafalda Camara et al. [4] simulated soft tissue deformation with position based dynamics and achieved acceleration on the GPU. Point based method is also one of the meshfree method. In 2004, Müller et al. used point based method to realize the deformation of elastoplastic and meltable objects. In Müller's method, both the volume and the surface representation are point based. But there are some limitations, because the particles are always interacted, therefore, the model would have to be extended.

In addition to the above several physical-based method proposed in the literature, people have already realized many relatively complete virtual training systems according to these methods in the past few decades. For example, kup-sze Choi [5] developed a cataract lens phacoemulsification training system in 2009. They use the multi mass spring model with the force feedback equipment to study the cutting and deformation of cornea. The cataract surgery training system has played an important role in cataract surgery. Yi-Je Lim et al. [11] used PAFF method to simulate the deformation of liver and extending the cutting operation to improve the topology and cutting path with the vertex movement method.

In these virtual interactions, we can know that the real-time, stability and the accuracy of the simulation are the basic requirement of the virtual surgery training system. There have been a lot of studies on thin shell fracture [3, 10], but the corneal

cutting is relatively rare. In terms of corneal transplant virtual surgery, it is a relatively delicate operation, including deformation simulation and cutting simulation. Firstly, the deformation parameters of the model should be determined by real material. Secondly, the model should be stable enough to achieve small deformation and large deformation. Finally, the model should be scalable, in this way, in addition to simulate the deformation of objects, this method also can achieve the cutting simulation.

In this paper, the point based method is used to simulate the deformation of cornea. This method is based on point based dynamics, in our method, we use the Müller's model to calculate the stress and strain. And add volume constraint to point based method to keep the controllability of model, we have realized real-time interactive process of virtual surgery and force feedback device. Compared with the FEM, the meshfree method can ensure the accuracy of calculation and reduce the difficulty of calculation, it can meet the real-time requirement of the operation and achieve an ideal effect in the deformation process.

3 Overview

The system overview of the proposed real-time simulation system is presented at Fig. 1. The system can be roughly divided into three stages: the construct model stage and the simulate deformation stage and the cutting stage.

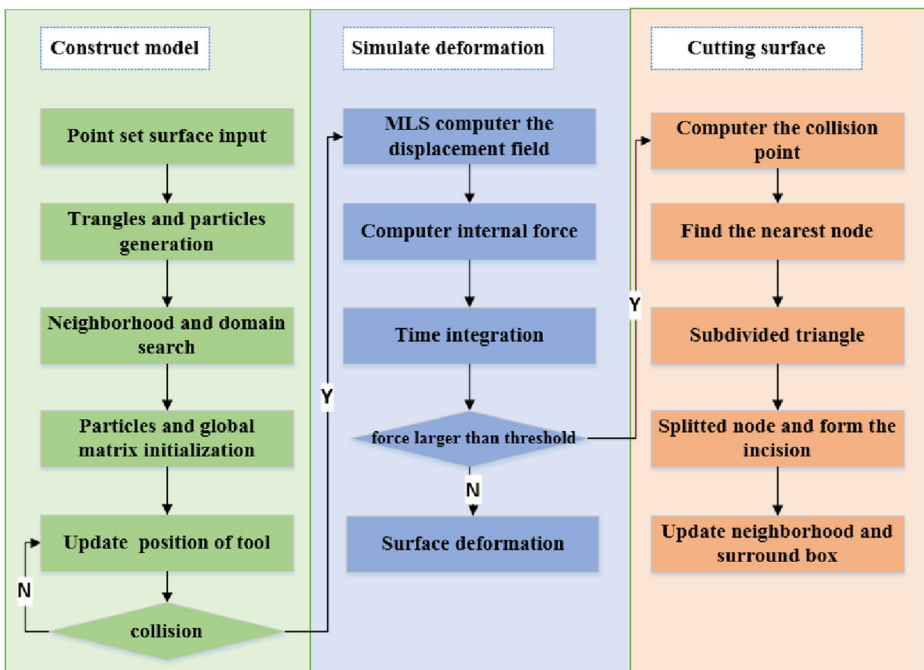


Fig. 1 The overview of our real-time meshless simulation system in virtual surgery

In the construct model stage, N particles are distributed in the boundary area, and construct some triangle with M surface sampling points, initialize the particles' properties. Young's Modulus and Poisson Ratio can be measured according to our experiments, and then set up the global matrix based on linear elastic theory.

In the simulate deformation stage, surgical tool exert an external force on the surface of corneal, the particles have a movement because of the external force, and the displacements of these particles can be calculated according to the Newton's second law. Then, we can calculate the deformation of the surface using dynamic equation based on point.

We designed the cutting stage to demonstrate that our model is scalability and stability. This method uses node replication method to cut the surface and subdivide the cutting triangles according to the topological relations among the surface nodes of the cornea, so that it can generate relatively smooth incision.

4 Real-time cornea deformation based on point method

4.1 Geometry model

Cornea is the outermost layer of the eyeball, it is a fibrous membrane, and has some features, such as: transparency, no blood vessel, its surface is smooth and elastic. Thickness of central corneal is only 0.58-0.64 mm, its horizontal diameter is about 11.6-12 mm, and the vertical diameter is about 10.6-11 mm. According to these physical features of the cornea, the simplified geometry of eyeball as shown in Fig. 2. Simplify the corneal to some meshes which are constructed with M surface sampling points, in the area surrounded by the cornea, we distributed some particles as you can see in Fig. 2c. During the simulation, we only simulate the deformation of the corneal part, and the small deformation of the rest of eyeball which is caused by corneal deformation is not explained in this paper. Therefore, in the following chapters, we emphasize the structure of the cornea just like show in Fig. 2a.

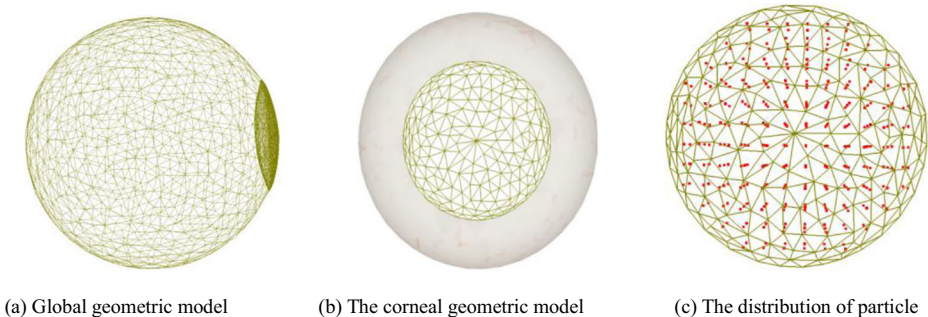
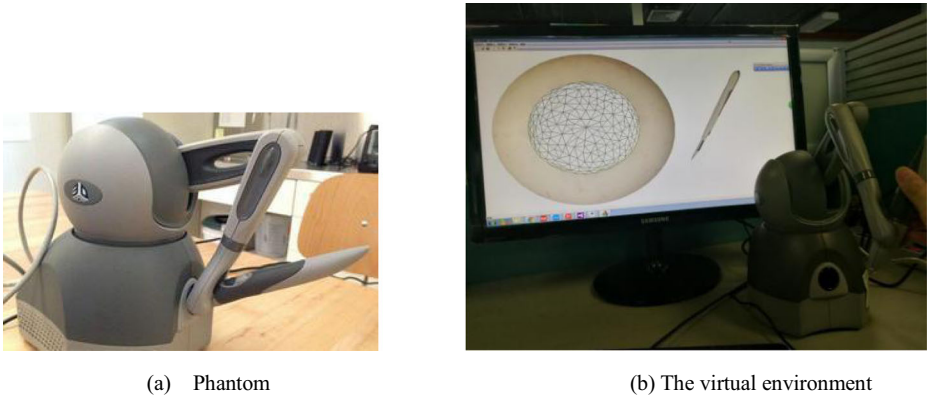


Fig. 2 The geometry model of eyeball

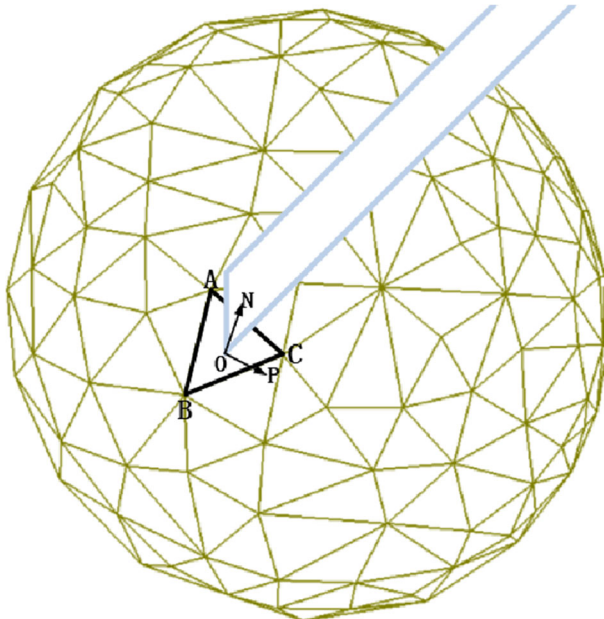


(a) Phantom

(b) The virtual environment

Fig. 3 The force feedback devices

Some discrete particles are distributed in the area surrounded by the cornea, the number of particles can affect the calculation speed of the deformation, and the size of meshes can effect the incision in the cutting stage. All of the numbers can be adjusted in the calculation to achieve the optimal calculation rate. In our model, at initialization time, we subdivide the mesh by software (meshlab) and distribute particles. The corneal model contains 192 simulation nodes and 350 meshes and 734 particles. The number of particles can meets the requirement that each surface nodes has at least k ($k > 10$) neighborhood particles, these particles are associated with the surface node in the topology, and can be used for future calculations during the simulation deformation.

**Fig. 4** The collision detection

4.2 Collision detection

Collision detection is used to determine whether the scalpel and the soft tissue are in contact. In this experiment we use force feedback devices to control the movement of the surgical tool. Force feedback equipment has six degrees of freedom and can provide realistic interactive behavior, as shown in Fig. 3.

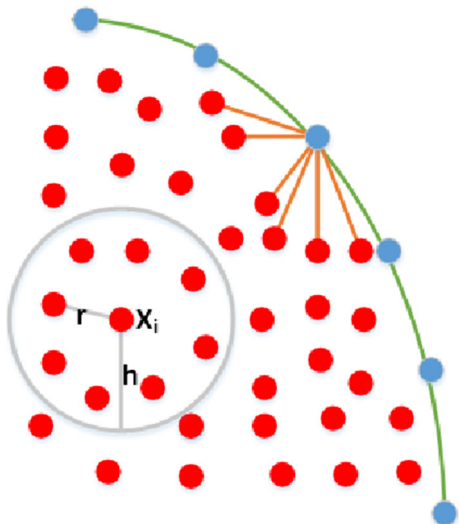
In this paper, the BVH [9] is used to accelerate the collision calculation between corneal model and surgical tool. Constructed a bounding box for the deformed object, and ensured that each underlying bounding box contains one triangle. Simplify the blade model and take ten sampling points evenly on the central axis. These sampling points are used to determine if there is a collision between the tip of the blade and the surrounding box, and then figure out the collision depth. In every time steps, update the position of the blade and the bounding box, traverses the binary tree to judge whether the tip of the blade and the surrounding box collide or not. And then we can analyse if the tip of knife intersects the triangular patches in the bounding box. The collision detection as shown in Fig. 4.

Assuming that O is the tip of the surgical tool, the triangle ABC is the mesh that surrounded by the bounding box, and ON is the normal vector of the triangle, which can be read directly from the obj file. If $\overrightarrow{OP} \cdot \overrightarrow{ON} \leq 0$, it illustrates that the angle between the normal vectors is greater than 90 , that means the tip touches the triangular patches, we can compute the collision point as the tool tip trajectory. Otherwise, the knife does not touch the object, update the position of the blade and continuing collision detection.

4.3 Point based method

We use the improved point-based dynamics method to simulate the deformation process. The main idea of the method is using a set of particles with physical properties to solve the stress tensor of the deformation process and then calculate

Fig. 5 The support domain



the displacement vector of every particle. Assuming that the deformed object can be discretized into N particles, and the properties of each particle can be obtained from the kernel function which is in the support domain. The support domain of particle can be a circular domain or a rectangular domain, Fig. 5 shows how to find the support domain and neighbors of particle and surface node.

Where the green line is the boundary area, and the blue dots represent the surface nodes, the red dots represent the discrete particles, the gray area represents the support domain Ω , therefore, the particles in the grey area are neighbors of X_i . The red line represents the topological relation between the surface nodes and its neighbors.

We can calculate the stress though the strain energy, and calculate the acceleration of the node using Newton’s second law by Eq. (1), and compute the next position of the node finally.

$$\frac{dv_i}{dt} = \frac{f_{external}}{m_i} + \sum_j -\nabla u_j U_i \tag{1}$$

Where v_i is the velocity at t , $f_{external}$ is the external force, including gravity, friction force, etc. u_i is the displacement vector of particles i , $\nabla u_j U_i$ represents the negative gradient of strain energy with respect to the displacement for the neighbor j of particle i . When the object has an elastic deformation under the external force, the strain energy will be stored in it, under the uniaxial stresses, the strain energy density formula is as follows:

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

v_i is the volume of particle i , which has been calculated on initialization. We suppose the investigated material is linear-elastic and isotropic, it is a Hooke material, and then the strain ε_i has a significant linear correlation with the stress σ_i . The linear Cauchy strain has been given:

$$\sigma = C\varepsilon \tag{3}$$

$$\varepsilon = \frac{1}{2} (J^T + J) - I \tag{4}$$

Where C is a four-rank tensor, for an isotropic linear elastic materials, C is determined by only two constants, elastic modulus E and Poisson’s ratio μ , I is the identity matrix. Force exerted by particle j on particle i is calculated as:

$$f_{ij} = -\nabla u_j U_i \tag{5}$$

In order to obtain the displacement gradient ∇u_j , we assume that the displacement file of particle i is u_i , and then estimate the displacement of the next step of the particle i such as:

$$u(x_i + \Delta x) = u_i + \nabla u_i \cdot \Delta x + o(\|\Delta x\|^2) \tag{6}$$

We use the first-order Taylor expansion to approximate, where Δt is the displacement file of the central particle. According to the displacement gradient of the particle i , we can approximate the gradient of the neighborhood j :

$$\tilde{u}_j = u_i + \nabla u_i \cdot X_{ij} \tag{7}$$

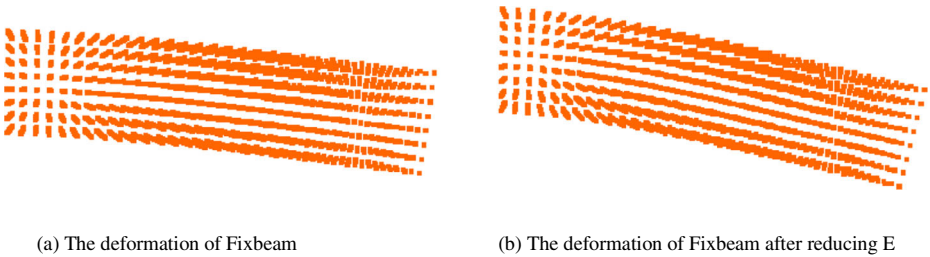


Fig. 6 The deformation of under gravity potential

Where x_{ij} is the distance between the center particle and the neighbor, and the u_j is a known values, according to the MLS:

$$R = \sum_j \left(\tilde{u}_j - u_j \right)^2 w(r, h) \tag{8}$$

To computer the minimum value of R , we can write the above formula as follow:

$$\left(\sum_j X_{ij} X_{ij}^T w_{ij} \right) \nabla u |x_i = \sum_j (u_j - u_i) X_{ij} w_{ij} \tag{9}$$

$$f_i = \sum_j -\nabla u_j U_i \tag{10}$$

Force f_i can be calculated by Eq. (10), and then computer the displacement of particle i . The algorithm required that there are at least four neighborhood points and does not support co-linear or co-planar, so the SVD method is more stable.

4.4 The constraint

The volume constraint could maintain the volume of the deformation object, add the volume constraints to this model can get a better results. We have got the volume of each particle at the time of initialization. Once the volume changed, elastic solids produce a force to resist the change in volume. It can be calculated by the following formula: $f_i = -\nabla u_i U_v$, where U_v is the changed of volume which is caused by the potential energy.

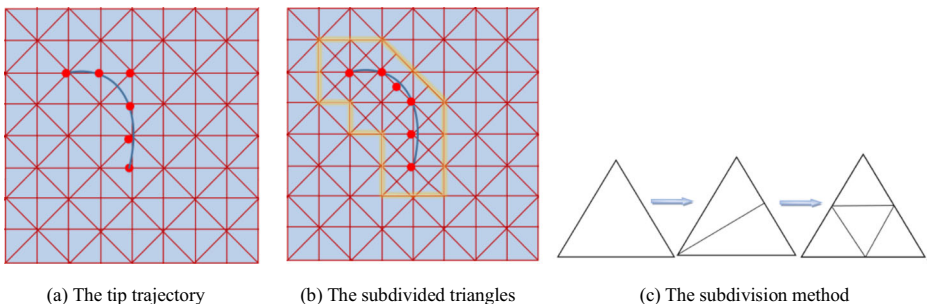
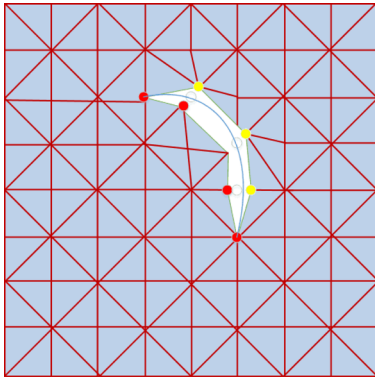
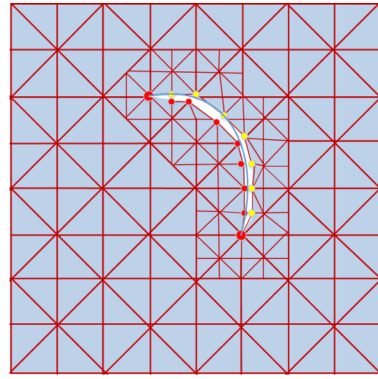


Fig. 7 The trajectory of the knife



(a) Form the incision



(b) Form the incision after subdividing triangles

Fig. 8 Effect contrast situation

The following two experiments are used to verify the correctness and real-time of the algorithm as shown in Fig. 6. Two fixbeam with identical geometry are attached to a wall, all of them consist of 3200 particles and have the same mass. E denote Young's modulus: $E = 10000N/m^2$, the deformation of the object under gravity as shown in Fig. 6a. Keeping the number of particles, and change the parameter to $E = 6000N/m^2$, the result as shown in Fig. 6b. It can be deduced from the two experiments above that elasticity of our model can be altered by adjusting Young's modulus.

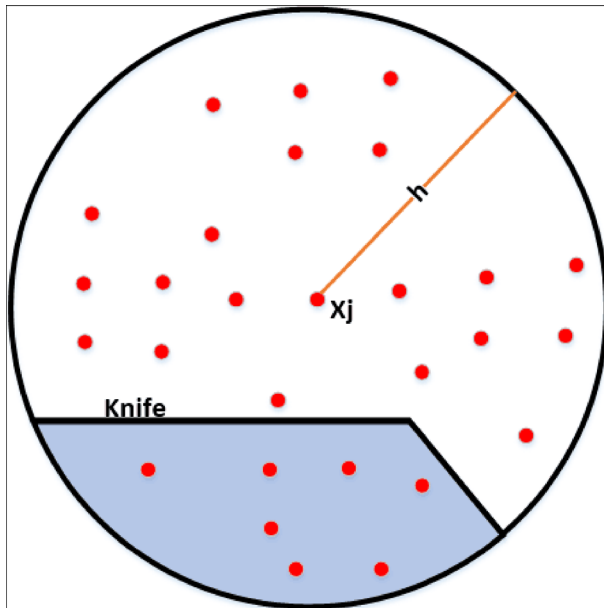


Fig. 9 The visibility criteria

5 Cutting simulation with node replication method

The surface of the cornea model is composed of triangles, the knife is abstracted as a straight line with zero thickness. In each step, there will be a collision detection between triangle and the tip to judge whether there have an intersection point or not. All these intersection points make up the trajectory of the knife, as shown in Fig. 7.

Where the blue line is the cutting path of the tip, the red dot is the vertex of the triangle that nearest the intersection point. As we can see in Fig. 7a, there is a error with the real trajectory. In order to reduce error and increase calculation precision, this paper presents a mesh subdivision method as shown in Fig. 7c. Find all the triangles where the nearest vertex is, and subdivide them. The meshes in the yellow grid region is the mesh who has been subdivided. As you can see in the above Figures, the link line of the vertexes in Fig. 7b is much smoother.

The algorithm of generate as follows:

$$X_R = X_i + \frac{dis}{2} V_{LR} \quad (11)$$

$$X_L = X_i + \frac{dis}{2} (-V_{LR}) \quad (12)$$

Where X_i is the position of nearest vertex to the tip, X_L and X_R are the duplicated nodes, they are in same pane with the trajectory. V_{LR} is the vector that perpendicular to the tip of the knife, dis is the width of incision.

This method only need to deal with the mesh that have been collided when formed the incision, do not need to change the topology of other meshes. In a network with coarse node spacing, the incision is rough. In order to obtain more accurate results, we use an improved node replication method as shown in Fig. 8b to get a more realistic incision. The contrast of results as shown in Fig. 8.

Blade segmentation will cause topology changes, and lead to the neighborhoods change of surface vertices and particles. In this case, it is necessary to recalculate the neighborhoods. Assumed that the blade's impact area is a cylinder, according to the visibility criteria in literature [17], recalculate the neighborhoods as Fig. 9:

Where the blue region is shaded by the blade, and the particle X_j is no longer searched from the region for neighbors. In the remaining area, we look for k neighbors of the particle, and if the quantity is not enough, the algorithm will expand the radius and keep searching.

6 Algorithm of deformation and cutting simulation

The process of the cornea deformation based on point method could be described as: regarded discrete particles as the basic unit, and initialize all of the particles. In order to conform whether the cornea and the knife collide, we employed the collision detection based on BVH in our method. If there was a collision, calculate the particles' stress tensor to driven corneal deformation; and adding volume constraint to the point based method. In this way, the more stability of deformation can be improved. After the stress of the particles reaches a critical value, the cornea is penetrated and start cutting simulation, using node replication method for surface cutting, and subdivide the cutting triangles in the cutting process, so that it can generate relatively smooth incision.

Assuming that a deformation object has n nodes, the quality of each node is m_i , the density is ρ_i , the position is x_i and the velocity is v_i . Each node contains j neighborhood points in support domain, the algorithm is as follows in one time step.

Algorithm of Deformation and Cutting Simulation

```

1: Forall particles  $i$ 
2:   Initialize  $X_i = X_i^0, v_i = v_i^0, W = W(h, r)$ 
3:   Compute neighborhoods, density and volume
4:   Compute the Kernel function  $W(h, r)$ 
5:   Initialize matrix
6: Endfor
7: Loop
8:   Collision detection with BVH
9:   Forall particles  $i$ 
10:    Computer deformation gradient  $\nabla u_i$  with MLS
11:    Computer internal force  $f_{ij}$ 
12:  Endfor
13:  Forall particles  $i$  do
14:     $v_i \leftarrow v_i + (f/m)\Delta t$ 
15:     $p_i \leftarrow X_i + v_i\Delta t$ 
16:    Volume constraint( $C_v$ )
17:    Update surface nodes  $u_{sfl} = \frac{1}{\sum_i w(r_i)} w(r_i)(u_i + \nabla u_i^T (X_{sfl} - X_i))$ 
18:    Update BVH
19:  Endfor
20:  Forall particle  $i$  do
21:    if  $f_i > f_{threshold}$ 
22:      Begin cutting()
23:    Endfor
24: Endloop

```

Compared with the previous algorithm, our algorithm improved the point based model. There is an equilibrium deformation undergoing large deformation. In this paper, we simulated the cutting process in the modified mode, and obtained a better results, proved the scalability of the model.

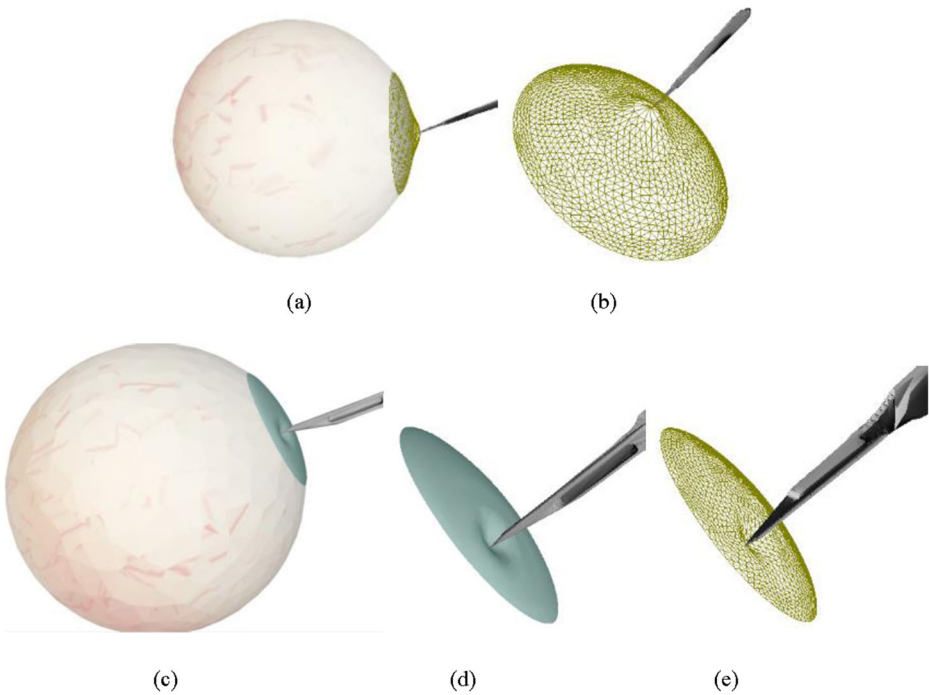


Fig. 10 Add force on the cornea

7 Results

In this section, the method is applied to the simulation of corneal deformation. C++ implementation the program of our system, and OpenGL is used to write graphics, all of the operations are in the CPU 3.6GHz, read memory for 16G computer to complete. By a phantom Omni force feedback device we can control the surgical tool and achieve interactive behavior, the device has 6 degrees of freedom.

Three experiments were performed on the platform: deformation of corneas, livers and stomachs. Firstly, simulate the deformation of cornea under the tensile force caused by surgical tools. The first group of experiments is to add force to the cornea, in this group of tests set the Young’s modulus $E = 6000N/m^2$, Poisson’s ratio $\nu = 0.33$. We used the hybrid model that we have constructed in chapter 4, added a tensile force on the cornea will lead to a deformation of cornea. As can be seen from Fig. 10a and b, the deformation direction of cornea and surgical tools force works in the same direction. In this process, in order to obtain more accurate deformation of the cornea, we subdivided the meshes during initialization.

Table 1 Time and frame rate for deformation, for different models

Model	Nodes/particles/face/E/ ν	Hybrid model		FEM	
		Time	Frame rate	Time	Frame rate
Cornea	192/734/350/6 k/0.33	18 ms	29fps	22 ms	26fps
Liver	502/3024/10 k/10 k/0.33	26 ms	25fps	30 ms	23fps
Stomach	512/3024/991/10 k/0.33	29 ms	25fps	31 ms	22fps

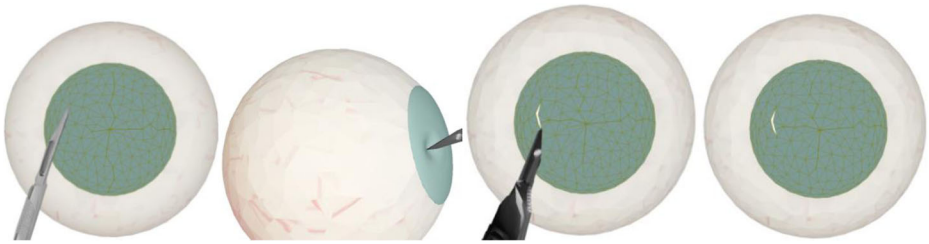


Fig. 11 The cutting of the corneas

Added an opposite direction force on the cornea and remained parameters unchanged, the deformation results with are shown in Fig. 10c. Figure 10d shows the deformation of cornea with texture. Therefore, the experiments suggest that change the direction and the magnitude of force can change the direction of the internal particles, and lead to stable deformation of the cornea.

Used FEM to simulate the model deformation and make comparison between FEM and our method, the result shown in Table 1, the experimental conditions of each experiment are identical. According to the experiments we can know that our proposed method not only has good stability under large deformation, and the real-time performance is also better than FEM (the deformation of liver and stomach will be shown later).

Our method is scalability, when the tip stress reaches the threshold, it will pierce the surface of the film and start cutting. The result of direct cutting is shown in Fig. 11. In this paper, we subdivided meshes when cutting the surface, in this way can form a smooth incision, as shown in Fig. 12. The cutting ran at about 30 fps, and this method can meet the real-time requirement.

Our method is mainly for corneal surgery such as cataract surgery. Because we adopted volume constraint in the model, so, the method can realize stable deformation of objects with large volume, such as liver and stomach. Figure 13 shows the deformation of the soft tissue and the incision formed during the cutting process. But further study is needed to construct the cutting surface. In addition, we employed the force feedback devices to control the surgical tools in all operations, that is to say, our system has realize the real-time interaction between the force feedback device and the model in virtual surgery.

8 Conclusion and future work

In this paper, we present a point based method to simulate the corneal deformation in the virtual surgery. Our model can simulate the deformation of corneal, and also can be used to simulate the liver and stomach, the experiment results have confirmed our algorithm.



Fig. 12 The cutting of the corneas after subdividing

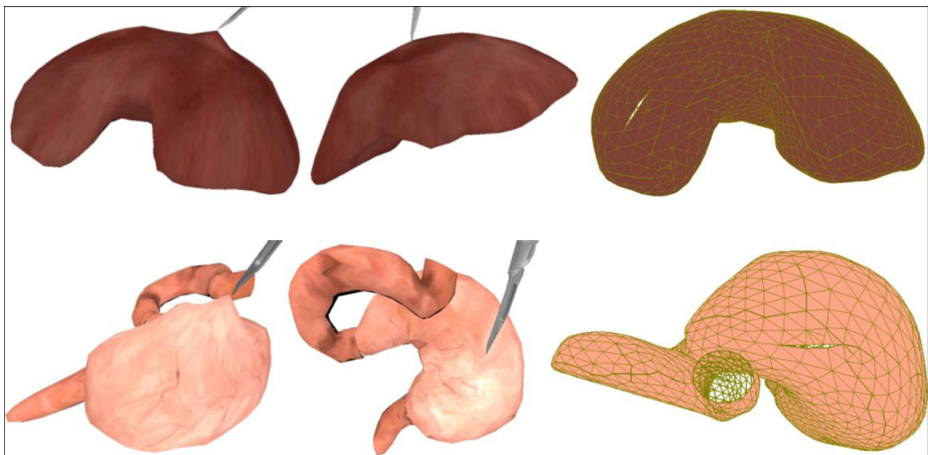


Fig. 13 The deformation of liver and stomachs

Compared with the previous researches, our system have realized the real-time interaction between the force feedback device and the cornea model in virtual surgery. But in order to apply the system to virtual surgery, the force feedback must provide high fidelity, which not only need more accurate calculations, but also need a lot of data processing. Different objects' force feedback determine by the density, angle and other factors. Furthermore, at the end of the article, we only simulated the cutting of the geometric model, there is no further processing after cutting, such as construct the cutting surface and so on. We will continue to study the deformation and cutting of the corneal in future work.

Acknowledgments This work is supported by the National key research and development project of China under Grant No.2016YFC0801406, the Natural Science Foundation of Shandong Province under Grant No. ZR2015FM013, the National Natural Science Foundation of China under Grant No. 61502279, the National key research and development project of the Shandong Province under Grant No. 2016GSF120012, and by Special Project Fund of Taishan Scholars of Shandong Province, Leading Talent Project of Shandong University of Science and Technology.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Bianchi G, Solenthaler B, Székely G, et al (2004) Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations [C]// Medical Image Computing and Computer-Assisted Intervention – Miccai 2004, International Conference Saint-Malo, France, September 26–29, 2004, Proceedings. DBLP, p 293–301
2. Bianchi G, Solenthaler B, Székely G et al (2004) Simultaneous topology and stiffness identification for mass-spring models based on fem reference deformations [J]. *Med Image Comput Comput Assist Interv* 2004:293–301
3. Busaryev O, Dey TK, Wang H (2013) Adaptive fracture simulation of multi-layered thin plates [M]. ACM
4. Camara M, Mayer E, Darzi A et al (2016) Soft tissue deformation for surgical simulation: a position-based dynamics approach [J]. *Int J Comput Assist Radiol Surg* 11(6):919–928
5. Choi KS, Soo S, Chung FL (2009) A virtual training simulator for learning cataract surgery with phacoemulsification [J]. *Comput Biol Med* 39(11):1020–1031
6. Faure F, Gilles B, Bousquet G et al (2011) Sparse meshless models of complex deformable solids [J]. *ACM Trans Graph* 30(4):1–10

7. Guo X, Qin H (2010) Real-time meshless deformation [J]. *Comput Animat Virt W* 16(3-4):189–200
8. Hong M, Jung S, Choi MH et al (2006) Fast volume preservation for a mass-spring system [J]. *IEEE Comput Graph Appl* 26(5):83–91
9. Jung H, Lee DY (2012) Real-time cutting simulation of meshless deformable object using dynamic bounding volume hierarchy [M]. John Wiley and Sons Ltd.
10. Kaufmann P, Martin S, Botsch M et al (2009) Enrichment textures for detailed cutting of shells [J]. *ACM Trans Graph* 28(3):1–10
11. Lim YJ, Hu J, Chang CY, et al (2006) Soft tissue deformation and cutting simulation for the multimodal surgery training [C]// *IEEE Symposium on Computer-Based Medical Systems*. IEEE Computer Society, p 635–640
12. Liu Y, Jiao S, Wu W, et al (2009) GPU accelerated fast FEM deformation simulation [C]// *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*. IEEE, p 606–609
13. Markus B, Markus I, Matthias T (2009) Corotated SPH for deformable solids [C]. *Eurographics Workshop on Natural Phenomena, NPH 2009, Munich, Germany*
14. Müller M, Keiser R, Nealen A, et al (2004) Point based animation of elastic, plastic and melting objects [C]// *ACM Siggraph/eurographics Symposium on Computer Animation*. Eurographics Association, p 141–151
15. Müller M, Heidelberger B, Teschner M, et al (2005) Meshless deformations based on shape matching [C]// *Acm Siggraph*. ACM, p 471–478
16. Müller M, Heidelberger B, Hennix M et al (2007) Position based dynamics ☆[J]. *J Vis Commun Image Represent* 18(2):109–118
17. Pauly M, Keiser R, Adams B et al (2005) Meshless animation of fracturing solids [J]. *ACM Trans Graph* 24(3):957
18. Peng Y, Jia R, Wang Y et al (2011) A virtual endoscopy system for virtual medicine [J]. *Comput Animat Virt W* 22(2-3):277–284
19. Peng Y, Ma Y, Wang Y et al (2017) The application of interactive dynamic virtual surgical simulation visualization method [J]. *Multimed Tools Appl*:1–18
20. Selle A, Lentine M, Fedkiw R (2008) A mass spring model for hair simulation [J]. *ACM Trans Graph* 27(3): 1–11
21. Shrivastava P, Das S (2014) Particle coding for Meshfree cutting of deformable assets [C]// *Indian Conference on Computer Vision Graphics and Image Processing*. ACM, p 6
22. Solenthaler B, Schläfli J, Pajarola R (2007) A unified particle model for fluid–solid interactions: research articles [J]. *Comput Animat Virt W* 18(1):69–82
23. Wang Y, Wu X, Wang G (2014) An angle bending constraint model for position-based dynamics [C]// *International Conference on Virtual Reality and Visualization*. IEEE, p 430–434
24. Wicke M, Botsch M, Gross M (2007) A finite element method on convex polyhedra [C]// *Computer Graphics Forum*. Blackwell Publishing Ltd, p 355–364
25. Zhu B, Gu L (2012) A hybrid deformable model for real-time surgical simulation [J]. *Comput Med Imaging Graph* 36(5):356–365



Yanjun Peng received his Dr. degree in march, 2004. He joined the Department of Computer Science, Shandong University of science and technology, Qingdao, China, in 1996, where he was promoted to professor in 2010. His main research interests include medicine visualization, virtual reality, and image processing.



Qiaoling Li is currently a Master in Science, Shandong University of science and technology, China. Her main research interest is image processing.



Yingying Yan is currently a Master in Science, Shandong University of science and technology, China. Her main research interest is multimedia software.



Qiong Wang received the B.S. and M.S. degrees in information science and technology from Shandong University of Science and Technology, Qingdao Shandong Province, China, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, Hong Kong. She is currently in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Her research interests include human–computer interaction, haptics, virtual reality, and computer-assisted surgery.