COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2015; **26**:321–335
Published online 22 April 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1655

SPECIAL ISSUE PAPER

Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics

Junjun Pan[1]*, Junxuan Bai[1], Xin Zhao[1], Aimin Hao[1] and Hong Qin[2]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, 100191, China
[2] Department of Computer Science, SUNY Stony Brook, NY, USA

ABSTRACT

This paper systematically describes an interactive dissection approach for hybrid soft tissue models governed by extended position-based dynamics. Our framework makes use of a hybrid geometric model comprising both surface and volumetric meshes. The fine surface triangular mesh with high-precision geometric structure and texture at the detailed level is employed to represent the exterior structure of soft tissue models. Meanwhile, the interior structure of soft tissues is constructed by coarser tetrahedral mesh, which is also employed as physical model participating in dynamic simulation. The less details of interior structure can effectively reduce the computational cost during simulation. For physical deformation, we design and implement an extended position-based dynamics approach that supports topology modification and material heterogeneities of soft tissue. Besides stretching and volume conservation constraints, it enforces the energy preserving constraints, which take the different spring stiffness of material into account and improve the visual performance of soft tissue deformation. Furthermore, we develop mechanical modeling of dissection behavior and analyze the system stability. The experimental results have shown that our approach affords real-time and robust cutting without sacrificing realistic visual performance. Our novel dissection technique has already been integrated into a virtual reality-based laparoscopic surgery simulator. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS
position-based dynamics; hybrid geometric models; deformation; interactive cutting; tetrahedra

***Correspondence**
Junjun Pan, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, 100191, China.
E-mail: pan_junjun@hotmail.com

1. INTRODUCTION AND MOTIVATION

In recent years, there have been remarkable progresses in the application of virtual reality-based (VR-based) surgery simulation and training [1]. Several commercial VR surgical simulators have been developed with great success [2,3]. An essential component of surgical simulators is the technique that supports realistic dissection of soft, deformable tissues in real time.

In principle, soft tissues can be represented by surface or volumetric models. Tetrahedral meshes, as one primary type of volumetric geometries, are frequently applied to the cutting simulation because the interior structure of objects must be presented during virtual surgery for the purpose of physical fidelity. However, the computational cost for topology modification in volumetric models tends to be extremely high due to its geometric and topological complexity [4]. In contrast, surface mesh models are relatively easy to handle in deformation and dissection simulation [5,6]. Nonetheless, it cannot exhibit the incision because the surface model is a simple skin that wraps around the model without volumetric substance. Although complex surface models with multiple layers offer an alternative solution to showcase cutting depth in training environment [7], it is far from being perfect to afford a realistic visual performance to simulate the incision of solid organs, such as liver, kidney, and spleen.

In many practical abdominal surgery procedures, both the exposure time and area of organs' interior structure are usually very limited during cutting [8]. So the surgeon tends to concentrate on the surface of organs as the major visual input during most of his or her operation time. According to this clinical common sense, we introduce a hybrid geometric model comprising both surface and volumetric meshes for interactive dissection simulation.

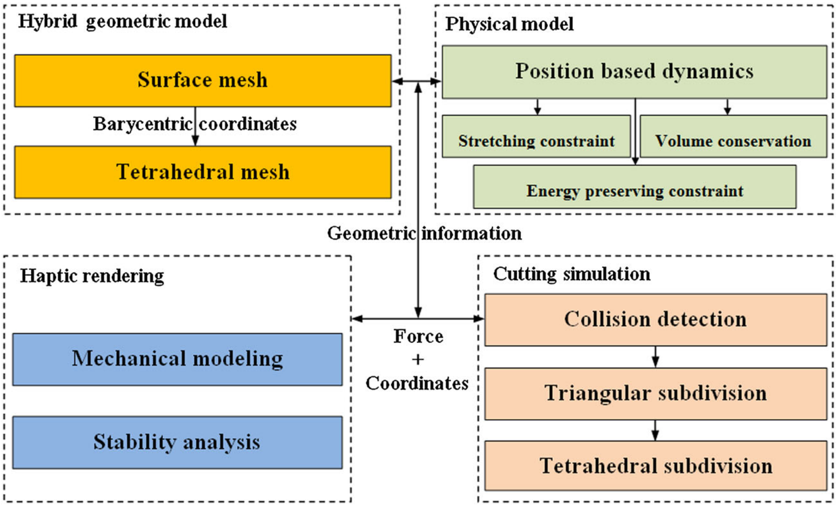


Figure 1. The framework of our dissection simulation method.

A fine triangular mesh is employed to represent the exterior structure of the soft tissue model. With an accurate geometric structure and texture at the detailed level, this surface model can offer a realistic graphical performance. Meanwhile, we represent the interior structure of soft tissue model with much coarser and larger tetrahedra. It significantly reduce the computational cost during deformation and element refinement/subdivision in cutting.

Figure 1 intuitively illustrates the pipeline and framework of our method. Specifically, the innovative contributions can be summarized as follows:

- We develop an interactive dissection approach for the hybrid soft tissue models, which comprise both surface and volumetric meshes with different detail levels. The connection between surface and volumetric meshes is generated by the mapping of barycentric coordinates. It facilitates a realistic graphical performance with low computational cost in deformation and dissection simulation.
- For physical deformation, we improve the position-based dynamics (PBD) and devise a new method to support the topology modification and material heterogeneities of soft tissue. Besides stretching constraint with strain limit and volume conservation constraint, it enforces the energy preserving constraint, which takes the different stiffness of material into account and improves the visual performance of soft tissue deformation.
- After subdivision of triangular and tetrahedral mesh, we present a straightforward but effective method to handle the cutting surface generation and re-mapping between new added surface vertices and tetrahedra in hybrid model.
- We integrate the mechanical model of cutting behavior and stability analysis with haptic rendering. Finally, this novel cutting method is adapted to a VR-based laparoscopic surgery simulator.

2. RELATED WORK

Dissection is an essential procedure in surgery. This is because cutting simulation requires real-time geometry and topology modification, which forms the most complex task in virtual surgery. Despite earlier research endeavors, interactive realistic dissection simulation remains to be a challenging topic and continues to attract a great deal of research efforts in the recent past [4,9]. In principle, cutting simulation approaches can be classified into three different categories.

Mass-Spring-Based Methods

This type of methods has been employed most frequently in dissection simulation. Zhang *et al.* [10] used a surface mass-spring model to simulate virtual dissection by progressive subdivision and re-meshing. A geometry-based surface subdivision algorithm is introduced to generate the interior structures that show the cutting gap between instruments and the soft tissue model. Choi *et al.* [11] presented a dissection algorithm for 3D triangular meshes. It can be used in the simulation of cataract surgery with phacoemulsification. Pan *et al.* [6] presented a surface mesh-based cutting algorithm with little topology modification. In order to improve the computing efficiency, a hierarchical database is applied to handle the mesh subdivision.

Finite Element Model-Based Methods

Finite element subdivision-based approaches are usually used to solve topological update, but it can create ill-conditioned subdivided elements [4]. Courtecuisse *et al.* [9] developed a finite element model (FEM)-based incision algorithm for heterogeneous soft tissues. Dick *et al.* [12] proposed a geometric multi-grid solver for cutting simulation based on a hexahedral discretization. Jeřábková *et al.* [13] proposed an interactive FEM-based cutting algorithm by removing the finest level voxels from multi-resolution volumetric images. Wu *et al.* [14] presented an efficient collision detection for composite finite element cutting simulation for deformable objects.

Mesh-Free Methods

Mesh-free methods can be also called mesh-less methods. Steinemann *et al.* [15] proposed a fast arbitrary splitting method using a mesh-less discretization of the deformation field. A novel visibility graph for fast shape updating functions are applied in mesh-less discretizations. Pietroni *et al.* [16] presented the splitting cubes, which is based on a mesh-free technique, to handle the interactive virtual cutting on deformable objects. Takayama *et al.* [17] used the lapped solid anisotropic textures to fill a volumetric model. It can be applied to simulate the cross-sectional cutting of solid objects.

Physically based deformation is another important research topic in virtual surgery. In this paper, we focus on the PBD method [18] because of its robustness and position-based manipulation feature, which makes it quite popular in the game industry and VR surgical simulator development [19,20]. Besides the aforementioned advantages, PBD method is easy to implement, and the constraints are easy to extend. Most recently, Müller *et al.* [21] also applied the PBD framework to simulate fluid. However, to the best of our current knowledge, there is little research work in the utility of PBD for deforming 3D soft objects that support topological change and material heterogeneities, which are the essential tasks in dissection simulation of 3D soft objects.

3. FRAMEWORK OVERVIEW

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

Hybrid Geometric Model

We use a hybrid geometric model comprising both surface and volumetric meshes to represent soft tissue models. The connection between surface and volumetric meshes is hinging upon the mapping of barycentric coordinates.

Physical Model

When computing deformation, we implement the extended PBD approach that accommodates the topology modification and material heterogeneities of underlying tissue. Three geometric constraints, including stretching, volume conservation, and energy preserving, are incorporated into the dynamic system.

Cutting Simulation

We design a straightforward collision detection method for dissection simulation. The cutting of hybrid soft tissue model is divided into two stages: triangular subdivision and tetrahedral subdivision. The constraints and linkages information are also updated in concert with cutting.

Haptic Rendering

We integrate both mechanical modeling of cutting behavior and system stability analysis for haptic rendering, to ensure the output force is smooth and the proxy date being transferred is robust.

4. PHYSICAL DEFORMATION USING POSITION-BASED DYNAMICS

4.1. Hybrid Geometric Models

We use both the surface and volumetric models to describe the soft tissue. A fine triangular mesh is applied to represent the exterior structure of the soft tissue model. With an accurate geometry structure and texture information at the detailed level, this surface model can afford a realistic graphics performance. Figure 2(a) and (b) illustrates the triangular mesh and the texture of the surface model for a liver. Meanwhile, we represent the interior structure of soft tissue model with large tetrahedra. This tetrahedral mesh is also used as physical model in deformation. The less detail of interior structure can reduce the computational cost in cutting and deformation. Here, we use the tool package PhysxViewer in NVIDIA PhysX SDK [19] to generate the tetrahedral volumetric mesh from the triangular surface mesh model. Figure 2(c) illustrates the corresponding tetrahedral mesh model of the liver. This volumetric model is slightly larger than the surface model to make sure that the tetrahedral mesh contains the triangular mesh completely. To generate the connection between tetrahedral model and surface model, we attach each vertex on the surface mesh to the closest tetrahedron in the volumetric mesh and store its barycentric coordinates. The coordinate mapping method in [22] is used to compute the geometry of surface mesh derived from tetrahedra. During deformation, the position of each vertex on the surface mesh can be interpolated by the positions of the attached tetrahedron using the stored barycentric coordinates. Figure 2(d) illustrates the liver model with both texture surface mesh and tetrahedral mesh. Figure 3 illustrates a scheme of this coordinate mapping.

4.2. Deformation Computing Using Position-Based Dynamics

For physical deformation, we use the PBD method primarily because of its two major advantages. The first advantage is that positions of vertices and parts of an object can be directly manipulated during simulation. The second advantage is that the position-based simulation offers the direct control over Verlet integration and removes the typical instability problem. These features make PBD much more attractive and convenient in the rapid development of VR surgery simulator [20]. Moreover, PBD method is easy to implement, and it is feasible to introduce more types of constraint to ensure the deformation obey the physical

DOI: 10.1002/cav

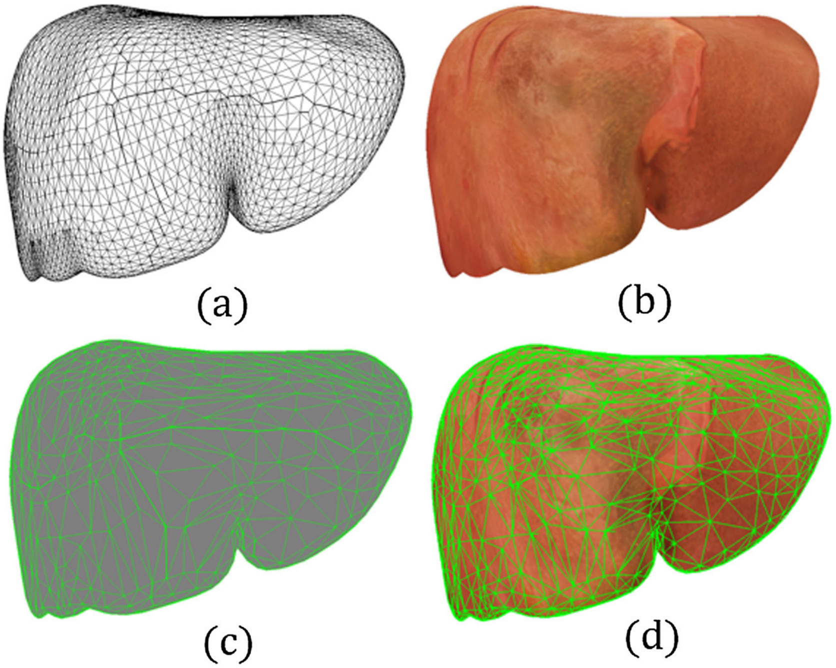


Figure 2. The hybrid geometry model of liver: (a) The triangular mesh, (b) Surface model with texture, (c) Tetrahedral volumetric model, and (d) Hybrid model with both texture surface mesh and tetrahedral mesh.

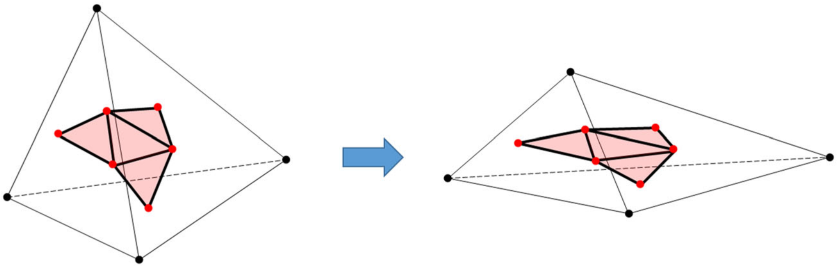


Figure 3. The coordinates of vertices on the surface mesh (colored in red) can be interpolated using the positions of the attached tetrahedron via barycentric coordinates.

rules. Our extended PBD deformation method can be described briefly in the following steps:

Step 1. Initialize the position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$ for each vertex in the tetrahedral mesh.

Step 2. At each time step Δt, update the velocity of each vertex through external force $\mathbf{f}$ and internal damping of the system. Assume that m_i represents the mass of vertex i, and $w_i = 1/m_i$, and we have

$$\mathbf{v}_i^{new} = \mathbf{v}_i + \mathbf{f}\Delta t w_i + Damp(\mathbf{v}_i) \tag{1}$$

Then, update $\mathbf{p}_i$, we obtain

$$\mathbf{p}_i^{new} = \mathbf{p}_i + \mathbf{v}_i^{new}\Delta t \tag{2}$$

Step 3. Traverse all the constraints for the estimated position $\mathbf{p}_i^{new}$. Apply these constraints to the dynamic system to obtain the updated position $\mathbf{p}_i^{sol}$ by solving the Gauss Seidel type iteration.

Step 4. After Step 3, we can obtain the final position and velocity through the following formula

$$\mathbf{p}_i^{fin} = \mathbf{p}_i^{sol} \tag{3}$$

and

$$\mathbf{v}_i^{fin} = \left(\mathbf{p}_i^{sol} - \mathbf{p}_i\right)/\Delta t \tag{4}$$

Step 5. According to the current position $\mathbf{p}_i$ of vertices in the tetrahedral mesh, update the positions of vertices in the surface triangular mesh.

Step 6. Go back to Step 2, compute the position and velocity of each vertex in the tetrahedral mesh for the next time step.

During the deformation of 3D soft object, we consider three types of constraints, which are directly applied to the positions of tetrahedral vertices. The first is the stretching constraint with strain limit, and the second is the volume-preserving constraint. The detail of these two constraints can found in [18]. To solve the material heterogeneities of soft tissues, we present the third constraint, energy preserving constraint, which takes the different stiffness of material into account. As a major innovative contribution in this paper, the technical detail is described in the following section.

4.3. Energy-Preserving Constraint

Currently, none of the constraints used in PBD are concerned with material property of object. Here, we design a new constraint, energy-preserving, which uses the spring potential of each deformed tetrahedron with spring stiffness (elastic coefficient). By setting different spring stiffness for tetrahedra in the area with variety of bio-mechanical material properties, it can give the plausible deformation for the heterogeneous soft tissue model.

Figure 4 shows an example of energy-preserving constraint for $\mathbf{p}_1$. $\mathbf{p}_0$ is the barycenter of $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$, there is

$$\mathbf{p}_0 = \sum_{i=1}^{4} m_i \mathbf{p}_i / \sum_{i=1}^{4} m_i \tag{5}$$

According to Hookes law, we use the following equation to express the constraint of energy conservation:

$$C_{energy}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \frac{1}{2}\sum_{i=1}^{4} k_i(\|\mathbf{p}_i - \mathbf{p}_0\| - d_i)^2 \tag{6}$$

where k_i is the elastic coefficient of virtual spring $\mathbf{p}_i\mathbf{p}_0$. d_i is the rest length of $\mathbf{p}_i\mathbf{p}_0$. So (6) describes the spring potential of deformable tetrahedron $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4$. And we can have

$$\nabla_{\mathbf{p}_i} C(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \frac{\mathbf{p}_i - \mathbf{p}_0}{|\mathbf{p}_i - \mathbf{p}_0|} \tag{7}$$

In [18], following equation is provided to calculate the positional increment $\Delta\mathbf{p}_i$.

$$\Delta\mathbf{p}_i = \frac{w_i C(\mathbf{p}_1, ..., \mathbf{p}_n)}{\sum_j w_j |\nabla_{\mathbf{p}_j} C(\mathbf{p}_1, ..., \mathbf{p}_n)|^2} \nabla_{\mathbf{p}_i} C(\mathbf{p}_1, ..., \mathbf{p}_n) \tag{8}$$

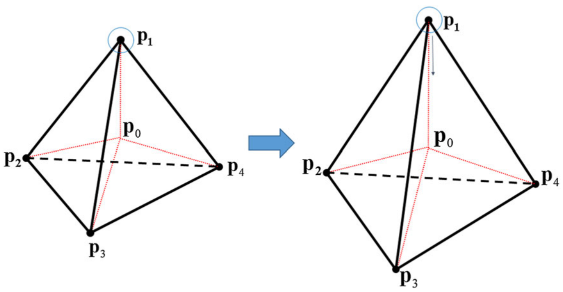


Figure 4. The illustration of energy-preserving constraint.

where $C(\mathbf{p}_1, ..., \mathbf{p}_n)$ is the constraint of an object for all of its points: $\mathbf{p}_1, ..., \mathbf{p}_n$, and $\nabla_{\mathbf{p}_i} C(\mathbf{p}_1, ..., \mathbf{p}_n)$ is the constraint's gradient at point $\mathbf{p}_i$. Its direction is dictated by the maximal change.

Finally, from (8), we can obtain

$$\Delta\mathbf{p}_i = \frac{w_i C_{energy}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)}{\sum_{j=1}^{4} w_j} \frac{\mathbf{p}_i - \mathbf{p}_0}{|\mathbf{p}_i - \mathbf{p}_0|} \tag{9}$$

According to (6), (9) contains k_i, which describes the spring stiffness of material for tetrahedral element $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4$. k_i can be set by the experimental measurement of physical parameters for soft tissues in practice [23]. We now incorporate the previously documented constraints into the PBD system for deformation.

Figure 5 shows an example of deforming the liver tissue by a grasper in a minimally invasive surgery (MIS).

5. INTERACTIVE DISSECTION OF HYBRID SOFT TISSUE MODELS

5.1. Triangular Subdivision

Collision detection is the first task that must be handled properly before the dissection process starts. We use the generalized cylinder-based collision detection method [24] to avoid the local concavity trap of surgical instruments during the interaction with soft tissues.

After collision detection, if the scalpel moves across any edge of triangles, the cutting algorithm divides the current triangle into several sub-triangles. The neighboring triangle with a common edge is added to the cutting list for the next triangle to be processed. During triangular subdivision, a hierarchical database [25] is constructed to record the relationship between every original triangle prior to its subdivision and its subdivided triangles. The subdivided triangles will be displayed when the scalpel leaves the current triangle. The vertices linked to the tetrahedron and barycentric coordinates will also be updated. Finally, once the simulation loop detects that the scalpel has left the mesh surface, the position of the last contact point is recorded, and the triangle with this point will be added to the cutting list as the last triangle for subdivision. Figure 6 illustrates the triangular subdivision on the liver surface mesh. The red curve is the cutting trajectory.

5.2. Cutting in Tetrahedra

Once the cutting starts on the surface mesh model, the cutting in tetrahedral volumetric model will be processed simultaneously. We treat the motion of scalpel as a finite length cutting edge passing through a soft object. Suppose the thickness of the blade can be completely ignored, then such problem is simplified to simulating a line segment moving in and through a tetrahedral mesh. In our algorithm, the position of line segment in both current and last

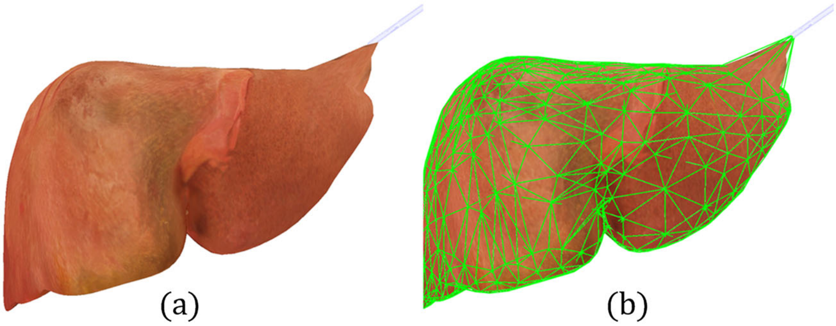


Figure 5. The deformation result of liver using the position-based dynamics method: (a) Surface model with texture, and (b) hybrid tetrahedral and surface model.

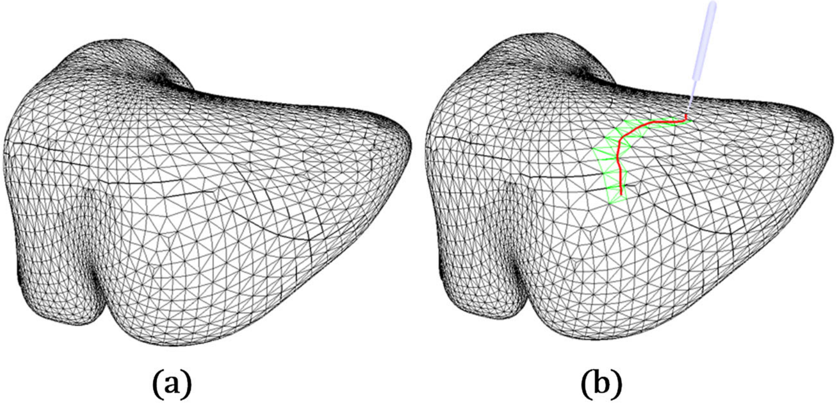


Figure 6. The triangular subdivision of cutting on the liver surface mesh: (a) Before cutting, and (b) after cutting.

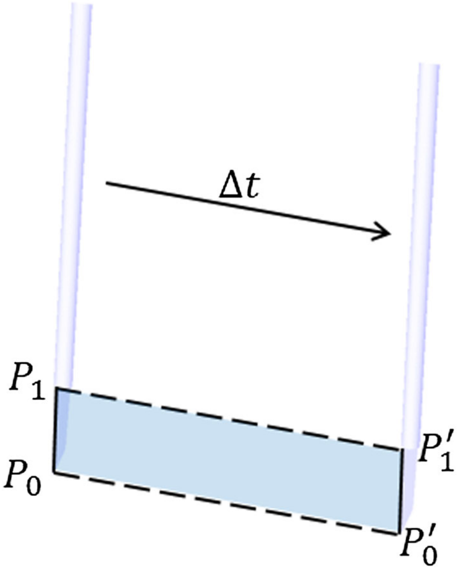


Figure 7. The construction of sweep surface.

time step is stored to form the sweep surface, which is built by connecting the corresponding endpoints of the two line segments (Figure 7).

Next, to determine the sweep surface and intersections created by the cutting edge, the following two tests are required. One is to detect the intersection between the scalpel tip path and the tetrahedral faces. The other is to detect the intersection of the sweep surface and the tetrahedral edges. We use the search method in [26] to handle these two detection tests. If an intersection occurs, the intersection information of the current tetrahedron will be updated. The neighboring tetrahedra that share the edge or face are updated accordingly. If the cutting edge no longer passes through an intersected edge or face, the original tetrahedron will be subdivided. To update the PBD physical model, the proceeded stretch, volume-preserving, and energy-preserving constraints shall be removed. And new constraints for the subdivided tetrahedra must be added as well. The spring stiffness of new edges can be computed by the interpolation of its adjacent connected edges. Finally, once the scalpel has left the mesh surface, the cutting in tetrahedra ends at the same time.

Consider both symmetry and rotation transformation, we enumerate five common subdivision cases based on the number of intersected faces and edges of tetrahedron (Figure 8). After subdivision computation for all the tetrahedra, the interior mesh will be separated along the sweep surface. Because of the large size of tetrahedra as interior geometric elements for soft tissue model, the merging

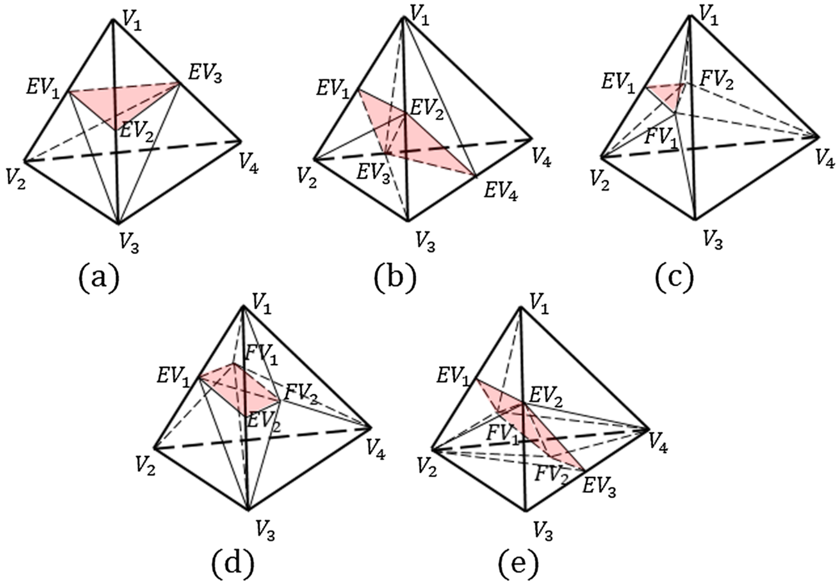


Figure 8. Five common cases for tetrahedral subdivision.

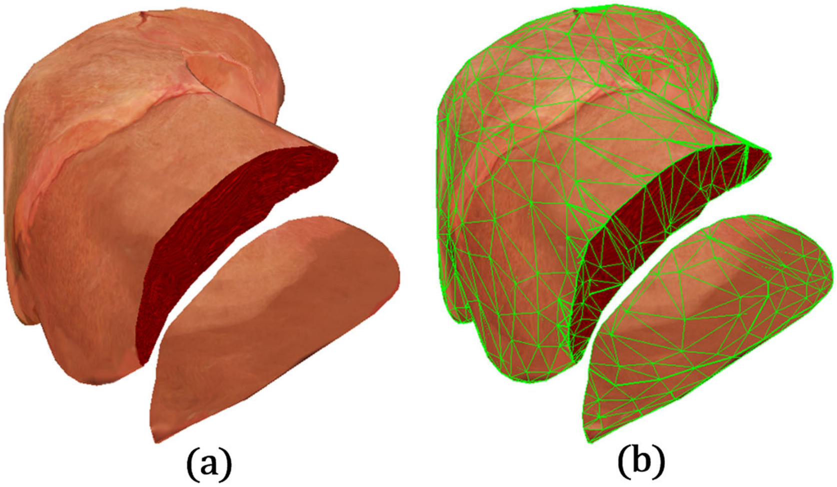


Figure 9. The cutting result of the hybrid liver model: (a) Surface model with texture, and (b) tetrahedral volumetric model.

of small or ill-shaped tetrahedra after subdivision can be avoided. It can reduce the computational cost during cutting. Figure 9 illustrates the result of cutting off a piece of tissue from the liver. Here, we use the technique provided by [27] to generate the texture of cross-section after incision.

5.3. Generating Incision Surfaces

After cutting and subdivision of surface mesh and tetrahedra, new triangular surface needs to be generated for cutting area. Figure 10 illustrates this process. There are two major operations. One is the cutting surface generation; the other is re-mapping the existing and newly added surface vertices to the correct subdivided tetrahedra. Before a cut, there are a tetrahedron and two surface triangles in Figure 10(a). The vector n indicates the normal of surface mesh. The cutting surface generation can be described as follows. Firstly, we compute the two types of intersections as Figure 10(b) illustrated. One is between sweep surface and tetrahedron; the other is between sweep surface and surface triangles. Then, we re-triangulate the external surface and the internal surface (the internal surface is generated by connecting the tetrahedron's intersection with triangles' intersections, the red triangles) as Figure 10(c-1). Meanwhile, we subdivide the original tetrahedron as Figure 10(c-2). After subdivision, we need to rebuild the mapping between surface vertices and tetrahedra. Finally, the two part of new surface are separated from each other as Figure 10(d).

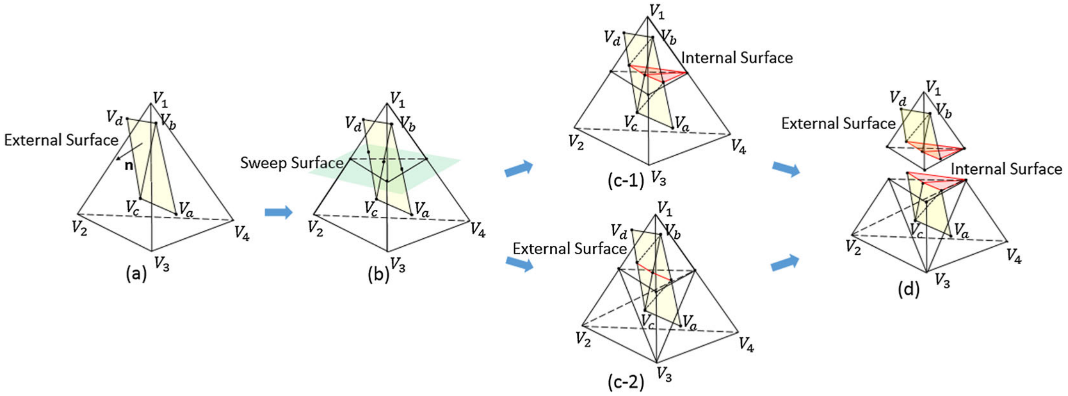


Figure 10. The generation of cutting surface.

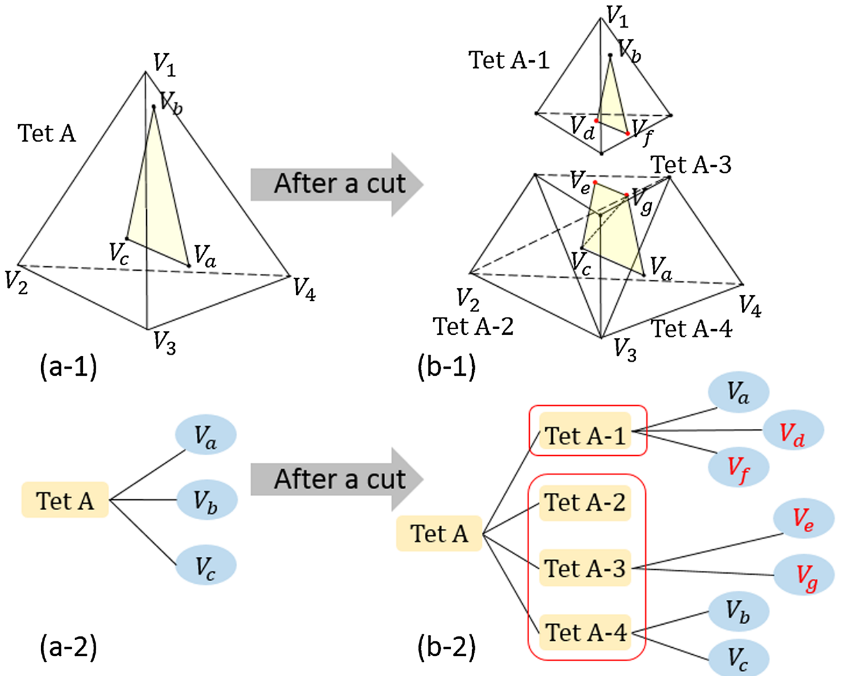


Figure 11. Re-mapping between surface vertices and the tetrahedra.

5.4. Remapping Between Surface Vertices and Tetrahedra

Correct re-mapping between tetrahedra and surface vertices after cutting is another important issue in geometry updating. Figure 11 illustrates this process. Before a cut, the original tetrahedron (Tet A) contains a triangle whose vertices are V_a, V_b, and V_c as shown in Figure 11(a-1). Without loss of generality, we assume that each of the three vertices locates inside Tet A. Figure 11(a-2) shows the relation between Tet A and its contained three surface vertices. There is a link between each surface vertex and Tet A. After a cut, the tetrahedron subdivides into four sub-tetrahedra, which are Tet A-1, Tet A-2, Tet A-3, and Tet A-4. Four intersection vertices (V_d, V_e, V_f, and V_g) are added. V_d and V_e have same position when cutting happens. Under the effect of stretching constraints in Section 4.3 , V_d and V_e begin to separate. And same happens for V_f and V_g. For existing vertex (V_a, V_b, and V_c), we just compute whether it locates in the subdivided tetrahedron and build a link between the vertex and the attached tetrahedron as in Figure 11(b-2). For intersection vertices, the pair of vertices cannot be divided into two different tetrahedra easily as they have same positions. Here, we set a small displacement to the pair of vertices at opposite directions. The directions are perpendicular to the sweep surface. After adding a displacement, we look for the bounding tetrahedron for each intersection vertex and build the link. Finally, we update its barycentric coordinates.

6. INTEGRATION WITH HAPTIC RENDERING

To apply our technique to virtual surgery training, we also have to integrate the haptic rendering into the cutting simulation. Geomagic Touch (Sensable, Phantom Omni) is chosen as the haptic device, which can simulate the manipulation of scalpel, grasper, hook cautery, or any other surgical instrument in practical surgery.

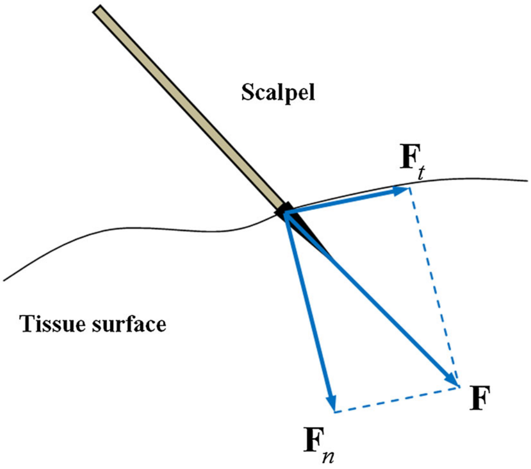


Figure 12. Decomposition of force during cutting.

6.1. Mechanical Analysis of Cutting Behavior

Technically speaking, there are two stages of the interaction between the scalpel and the object surface when cutting the soft tissue. First of all, the scalpel presses the surface, and the soft tissue deforms afterwards. When the pressure exceeds a threshold, F_p, the scalpel penetrates into the tissue, and the surface will be opened. Then, the scalpel will move when the tangential force exceeds the static friction. Here, we use the following mechanical model (Figure 12) to describe the cutting behavior.

$$\mathbf{F} = \mathbf{F}_t + \mathbf{F}_n \tag{10}$$

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

$$|\mathbf{F}_n| > F_p, |\mathbf{F}_t| > F_s \tag{11}$$

In (11), the instrument will open the tissue only if the magnitude of $\mathbf{F}_n$ is larger than a given tissue-dependent

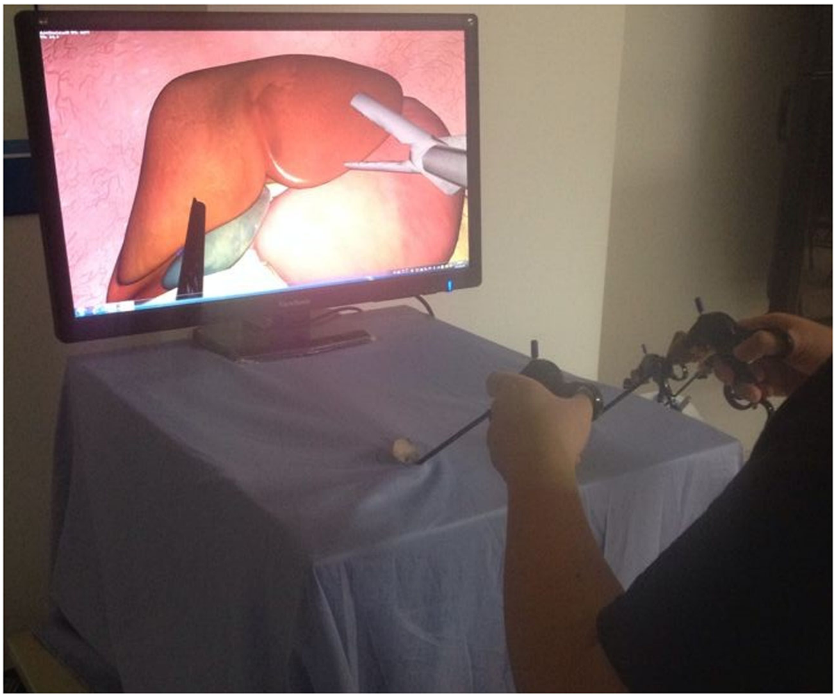


Figure 13. The interface of a prototyped VR laparoscopic surgery simulator.

threshold F_p. Then, if the magnitude of $\mathbf{F}_t$ exceeds the value of static friction of tissue, F_s, the instrument will start dissection. During cutting, the system will monitor and analyze the value of $\mathbf{F}$ in each haptic frame. If $\mathbf{F}_t$ is smaller than the value of dynamic friction of tissue, F_d, the scalpel stops. If the scalpel leaves the tissue surface, the cutting ends. For this process, all parameters, F_p, F_s, and F_d can be set by the experimental measurement.

This mechanical model can be utilized to output the intuitive force feedback during dissection simulation. The direction of output force as cutting friction is opposite to the direction of scalpel movement. The force magnitude

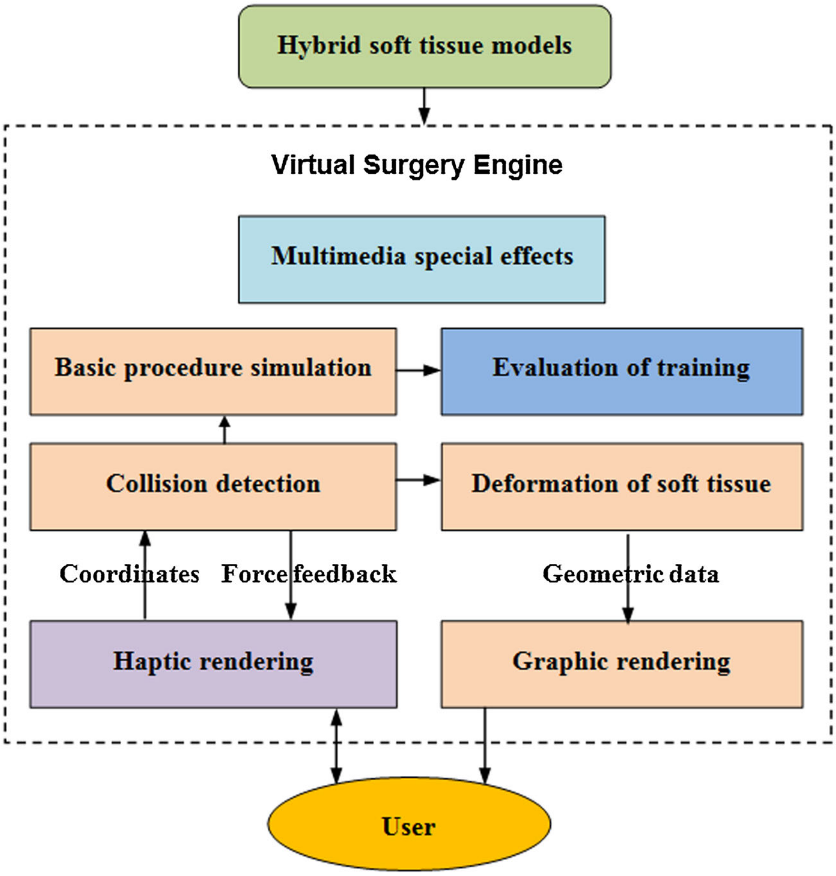


Figure 14. The architecture of our surgical simulator.

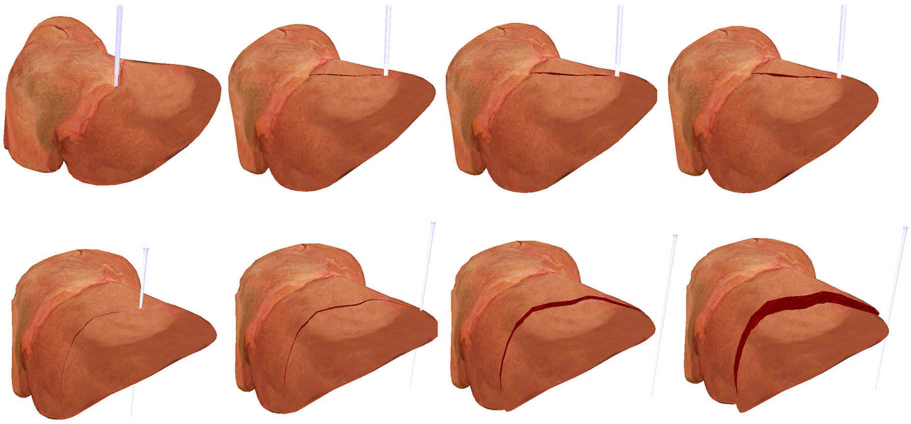


Figure 15. The cutting simulation of the liver tissue.

is proportional to the depth of scalpel edge stabbed into the tissue.

6.2. Stability Analysis

In haptic rendering, the stability analysis of virtual environments is still an open problem. Because all the hardware parameters are different and unknown to system developers, it is difficult to set up a general stability criterion for a haptic device-based system. Moreover, along with the deformation of soft tissues and the updating of 3D motion information for virtual surgery instruments in each haptic frame, the virtual force mapping is usually a nonlinear problem [28]. Based on the virtual wall model and passive characteristics of proxy in contact, we use the method in [6] to control three parameters: stiffness and damping of the virtual tissue and removing coefficient of material element from the soft tissue, to ensure the system stability.

7. APPLICATION IN VIRTUAL REALITY SURGICAL SIMULATOR

Our ultimate goal is to apply this dissection simulation method to the virtual reality-based medical training and treatment. To validate our dissection method, we have incorporated it into a prototyped VR laparoscopic surgery simulator developed. Figure 13 shows the interface of this VR simulation system, where two MIS instruments, grasper and hook cautery, are controlled by the Geomagic

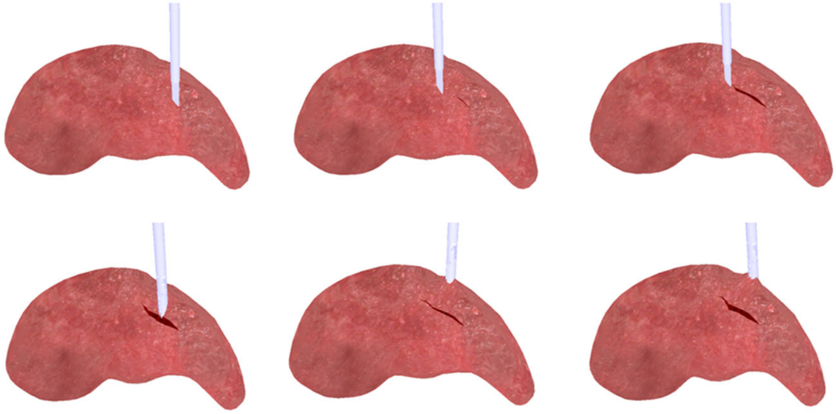


Figure 16. The cutting simulation of the spleen model.

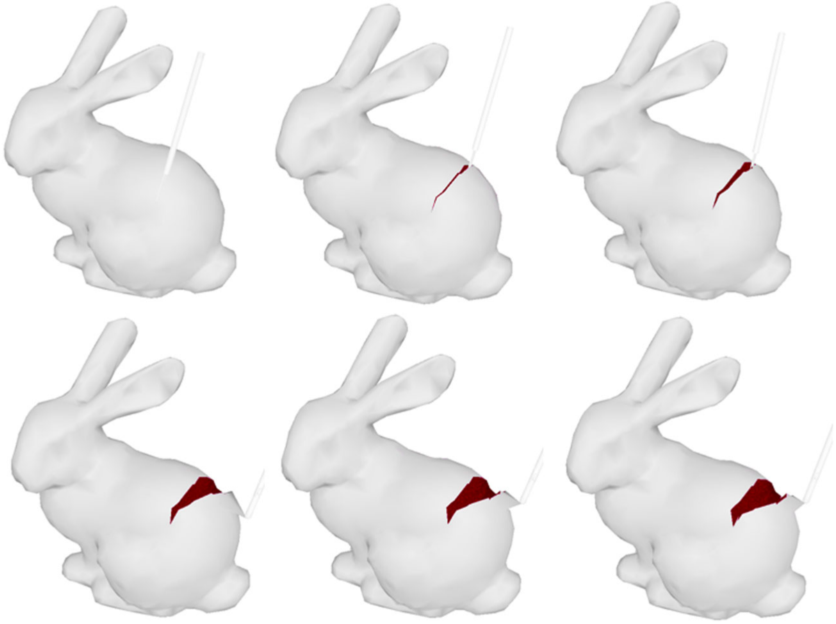


Figure 17. The cutting simulation of the Stanford bunny.

Touch haptic devices in the box. This prototyped medical simulator has been equipped with the following essential functionalities in laparoscopic surgery training:

- The laparoscope navigation and orientation in the virtual abdominal cavity.
- Laparoscopic real-time graphic rendering and deformation of soft tissue.
- The manipulation of basic MIS instruments, such as grasper, scissor, and hook cautery.
- The force feedback control in haptic rendering.
- The simulation of four basic operation procedures: small object transfer, dissection, suturing, and ligation.
- Multimedia special effects, such as sound, and smoke generation in electrosurgical dissection.
- Auxiliary functionalities, such as evaluation of training.

Figure 14 illustrates the system architecture. The core modules make up the Virtual Surgery Engine of this simulator.

8. EXPERIMENTAL RESULTS AND DISCUSSION

We have implemented our interactive dissection simulation technique using OpenGL, CUDA, and OpenHaptics. All the experiments run on a desktop with NVIDIA GeForce GTX 580, Intel(R) Xeon(R) CPU (2.53 GHz, 8 cores), and 12G RAM. The haptic rendering loop is running on a separate thread, so the update rate is guaranteed around 1 kHz. We have designed three sets of experiments. The first experiment is cutting a liver model. The first row in Figure 15 illustrates the simulation results of a small incision on the surface of liver tissue. Because of the influence of gravity and neighboring strain in biological soft tissue [23], we can find the cut grows bigger afterwards. The second row in Figure 15 illustrates the procedure that is cutting off a piece of tissue from the liver. We can observe that our method easily handles the large topology and volumetric structure modification. The second experiment is cutting a spleen model (Figure 16). The first row is having a small incision on the spleen. The second row is stretching the tissue by a grasper to open this incision. The third experiment is cutting and deforming a soft Stanford bunny by the same way (Figure 17). From Figures 15–17, we can observe that our method gives the dissection simulation with both accurate topology modification and realistic deformation performance.

Table I. Time performance (in millisecond).

Model	Triangles	Elements	Deformation	Cutting
Liver	16 252	4079	4.1	15.2
Spleen	2826	2385	2.9	5.2
Bunny	5110	2129	2.6	5.1

Furthermore, to fully analyze the time performance, we test our cutting simulation of liver model, spleen model, and the Stanford bunny. Table I lists the number of tetrahedra, the time costs for major steps, which include collision detection, physical deformation, topological updating, and dynamic rendering. The time record in Table I shows that our method is competitive for real-time virtual surgery simulators.

We also compare our dissection method with latest FEM method [29]. Their result is showed in Figure 18(a). We cut the steak model with the same direction and position and record four frames in the animation as the last four pictures in Figure 18(b)–(e). The iteration time for PBD is 3. Table II illustrates that our method can use far less

Table II. Comparison with mass-spring methods.

Method	Constraints	Iterations	Deformation (ms)
PBD	6427 + 4079	1	1.42
		3	4.64
		5	6.83
Explicit Euler	6427		1.16
Runge-Kutta	6427		2.28
Implicit Euler	6427		5.36

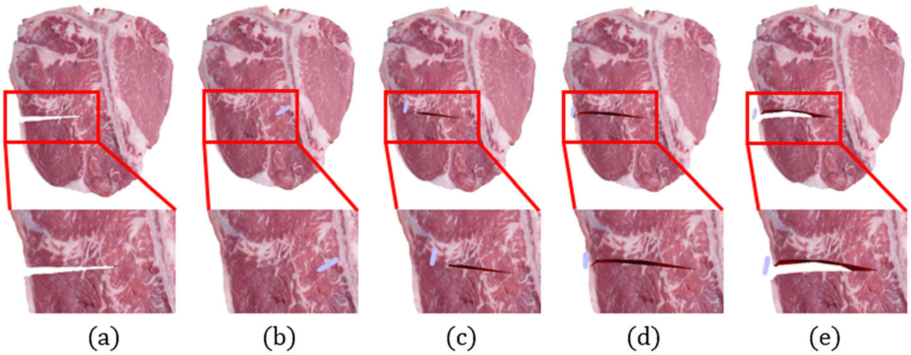


Figure 18. The comparison of cutting a steak model with the latest FEM method: (a) The result in [29], and (b)–(e) the result in our method.

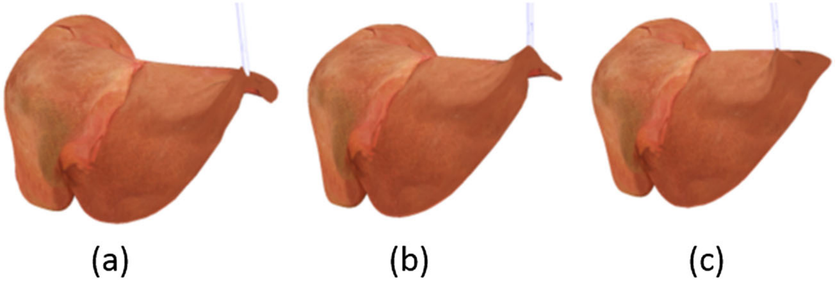


Figure 19. The comparison between our method and mass-spring model methods in soft tissue deformation: (a) Explicit Euler method, (b) implicit Euler method, and (c) our method.

Table III. Comparison with FEM (in millisecond).

Method	Elements	Deformation	Cutting
[29]	6272	25.7	8.6
Ours	1477	1.7	6.5

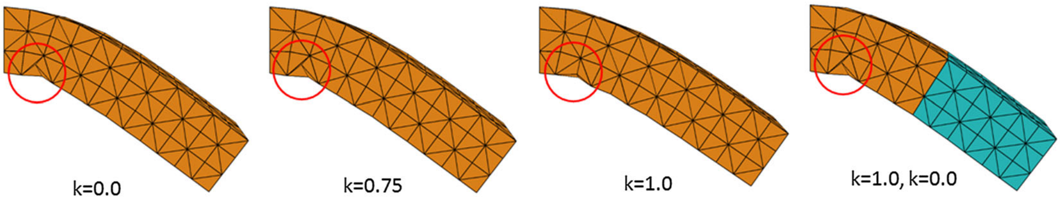


Figure 20. The deformation of a bar by extended PBD with different spring stiffness.

elements and computation time to obtain the almost same visual performance of deformation.

Moreover, we compare our method with Explicit Euler Method (EEM) and Implicit Euler Method (IEM) in mass-spring model (Figure 19). Table III is the numerical comparison. Three methods have the same stretching constraints, but PBD owns additional 4079 volume conservation constraints. These additional constraints need more processing times. The iteration time in EEM is 1; the iteration time in PBD is 3. In Figure 19, volume conservation is missing in EEM and IEM, but it is maintained in PBD.

Finally, we demonstrate our extended PBD method to deform a bar with different spring stiffness. This bar contains 3 × 3 ×12 grids and the left end is fixed. The bar will deform under the gravity. The extended PBD becomes plain PBD when k is 0. From Figure 20, the bar becomes "stiffer" when k increases. And the deformation result is different for the bar with homogeneous material and with heterogenous material.

9. CONCLUSION AND FUTURE WORK

We have presented an interactive dissection simulation approach based on the extended PBD for hybrid soft tissue models. Our method employs a hybrid geometric model comprising both surface and volumetric meshes in different detail level. It can greatly reduce the cost of computation without loss of realism. Our physical deformation is governed by the extended PBD approach that accommodates the topology modification of underlying tissue models and enforces their volume-preserving constraint. The energy-preserving constraint is applied to takes the different spring stiffness of material into account and improves the visual performance of material heterogeneities for soft tissue deformation. In addition, we integrate both mechanical modeling of cutting behavior and system stability analysis with haptic rendering. From comprehensive experimental results, we have observed that this new dissection method can offer real-time and robust simulation without sacrificing the realistic visual performance. As an application case, we have migrated this approach into a prototype VR laparoscopic surgery simulator.

In the near future, with the assistance and qualitative/quantitative evaluation from surgeons for pilot study, we plan to validate our VR surgical simulator in several practical surgery training tasks, such as hepatectomy and cholecystctomy. At present, we only have CUDA implementation in collision detection in our prototype system. We plan to further exploit the parallel acceleration and apply CUDA to both deformation and geometric refinement during cutting simulation. We also plan to add several special visual effects, such as bleeding, in the procedure to make the simulation much more realistic and visually appealing.

ACKNOWLEDGEMENTS

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

REFERENCES

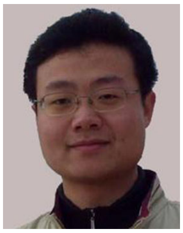
1. Grantcharov T-P. Virtual reality simulation in training and assessment of laparoscopic skills. *European Clinics in Obstetrics and Gynaecology* 2006; **2**(4): 197–200.
2. Simbionix. 2014. Available from: http://simbionix.com/simulators/lap-mentor/ [Accessed on 30 March 2015].
3. Mentice. Available from: http://www.mentice.com/ [Accessed on 30 March 2015].
4. Wu J, Dick C, Westermann R. Physically-based simulation of cuts in deformable bodies: A survey. In *Eurographics 2014 State-of-the-Art Report*. Eurographics Association: Aire-la-Ville, Geneva, Switzerland, 2014; 1–19.
5. Pan J, Yang X, Xie X, Willis P, Zhang J. Automatic rigging for animation characters with 3d silhouette. *Computer Animation and Virtual Worlds* 2009; **20**(2–3): 121–131.
6. Pan J, Chang J, Yang X, Qureshi T, Howell R, Hickish T, Zhang J. Graphic and haptic simulation system for virtual laparoscopic rectum surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery* 2011; **7**(3): 304–317.
7. Halic T, Sankaranarayanan G, De S. Gpu-based efficient realistic techniques for bleeding and smoke generation in surgical simulators. *International Journal of Medical Robotics and Computer Assisted Surgery* 2010; **6**(4): 431–443.
8. Aggarwal R, Moorthy K, Darzi A. Laparoscopic skills training and assessment. *British Journal of Surgery* 2004; **91**(12): 1549–1558.
9. Courtecuisse H, Allard J, Kerfriden P, Bordas S, Cotin S, Duriez C. Real-time simulation of contact and cutting of heterogeneous soft-tissues. *Medical Image Analysis* 2014; **18**(2): 394–410.
10. Zhang H, Payandeh S, Dill J. On cutting and dissection of virtual deformable objects. In *Proceedings of 2004 IEEE International Conference on Robotics and Automation.* New Orleans, LA, USA, 2004; 3908–3913.
11. Choi K, Soo S, Chung F. A virtual training simulator for learning cataract surgery with phacoemulsification. *Computers in Biology and Medicine* 2009; **39**(11): 1020–1031.
12. Dick C, Georgii J, Westermann R. A hexahedral multigrid approach for simulating cuts in deformable objects. *IEEE Transactions on Visualization and Computer Graphics* 2011; **17**(11): 1663–1675.
13. Jeřábková L, Bousquet G, Barbier S, Faure F, Allard J. Volumetric modeling and interactive cutting of deformable bodies. *Progress of Biophysics and Molecular Biology* 2010; **103**(2–3): 217–224.
14. Wu J, Dick C, Westermann R. Efficient collision detection for composite finite element simulation of cuts in deformable bodies. *The Visual Computer* 2013; **29**(6–8): 739–749.
15. Steinemann D, Otaduy MA, Gross M. Fast arbitrary splitting of deforming objects. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association: Aire-la-Ville, Geneva, Switzerland, 2006; 63–72.
16. Pietroni N, Ganovelli F, Cignoni P, Scopigno R. Splitting cubes: a fast and robust technique for virtual cutting. *The Visual Computer* 2009; **25**(3): 227–289.
17. Takayama K, Okabe M, Ijiri T. Lapped solid textures: Filling a model with anisotropic textures. *ACM Transactions on Graphics (TOG)* 2008; **27**(3): 53–62.
18. Bender J, Muller M, Teschner M, Macklin M. A survey on position based simulation methods in computer graphics. *Computer Graphics Forum* 2014; **33**(6): 228–251.
19. PhysX-Nvidia. 2015. Available from: http://physxinfo.com/wiki/ [Accessed on 30 March 2015].
20. Maciel A, Halic T, Lu Z, Nedel LP, De S. Using the physx engine for physics-based virtual surgery with force feedback. *International Journal of Medical Robotics and Computer Assisted Surgery* 2009; **5**(3): 341–353.
21. Macklin M, Müller M. Position based fluids. *ACM Transactions on Graphics* 2013; **32**(4): 104–116.
22. Müller M, Gross M. Interactive virtual materials. In *Proceedings of Graphics Interface*. CIPS: Mississauga, Ontario, Canada, 2004; 239–246.
23. Alastrué V, Garía A, Peña E, Rodríguez J-F, Martínez M-A, Doblaré M. Numerical framework for patient-specific computational modelling of vascular tissue. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**(1): 35–51.
24. Pan J, Chang J, Yang X, Liang H, Qureshi T, Howell R, Hickish T, Zhang J. Virtual reality training and assessment in laparoscopic rectum surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery* 2015; **7**(3): 204–217.
25. Choi C, Kim J, Han J, Ahn B, Kim J. Graphic and haptic modeling of the oesophagus for vr-based medical simulation. *The International Journal of Medical*

DOI: 10.1002/cav

Robotics and Computer Assisted Surgery 2009; **5**(3): 257–266.

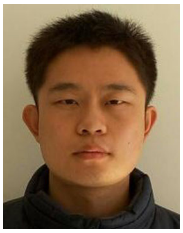
26. Bielser D, Gross M-H. Interactive simulation of surgical cuts. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications,* PG '00. IEEE Computer Society: Washington, DC, USA, 2000; 116–442.
27. Li X, Guo X, Wang H, He Y, Gu X, Qin H. Meshless harmonic volumetric mapping using fundamental solution methods. *IEEE Transactions on Automation Science and Engineering* 2009; **6**(3): 409–422.
28. Lin M-C, Otaduy M. *Haptic Rendering: Foundations, Algorithms and Applications*. A. K. Peters, Ltd: Natick, MA, USA, 2008.
29. Yang C, Li S, Wang L, Hao A, Qin H. Real-time physical deformation and cutting of heterogeneous objects via hybrid coupling of meshless approach and finite element method. *Journal of Visualization and Computer Animation* 2014; **25**(3–4): 423–435.

AUTHORS' BIOGRAPHIES



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

Junxuan Bai is a master's student in Beihang University. From 2012, he studied in the State Key Laboratory of Virtual Reality Technology and Systems in China. His research interests include virtual surgery and 3D visualization.



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

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

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