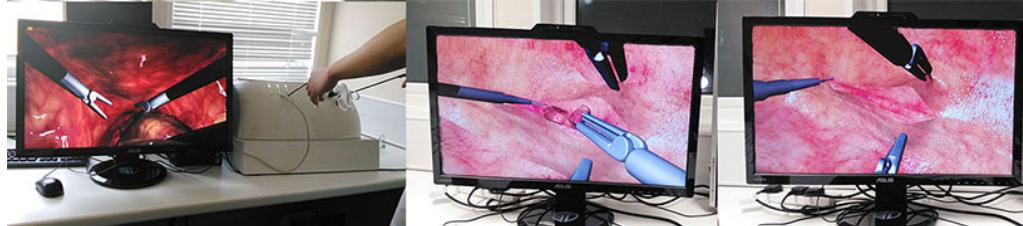


# Virtual Reality Based Laparoscopic Surgery Simulation

Kun Qian \*<sup>1</sup>, Junxuan Bai <sup>2</sup>, Xiaosong Yang <sup>1</sup>, Junjun Pan<sup>2</sup>, and Jianjun Zhang<sup>1</sup>

<sup>1</sup>National Center for Computer Animation, Bournemouth University

<sup>2</sup>State Key Laboratory of Virtual Reality Technology and Systems, Beihang University



**Figure 1:** Our laparoscopic surgery simulator, which includes different training modules, surgery tools, haptic device and surgery box

## Abstract

With the development of computer graphic and haptic devices, training surgeons with virtual reality technology has proven to be very effective in surgery simulation. Many successful simulators have been deployed for training medical students. However, due to the various unsolved technical issues, the laparoscopic surgery simulation has not been widely used. Such issues include modeling of complex anatomy structure, large soft tissue deformation, frequent surgical tools interactions, and the rendering of complex material under the illumination of headlight. A successful laparoscopic surgery simulator should integrate all these required components in a balanced and efficient manner to achieve both visual/haptic quality and a satisfactory refreshing rate. In this paper, we propose an efficient framework integrating a set of specially tailored and designed techniques, ranging from deformation simulation, collision detection, soft tissue dissection and rendering. We optimize all the components based on the actual requirement of laparoscopic surgery in order to achieve an improved overall performance of fidelity and responding speed.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

**Keywords:** Laparoscopic surgery, deformation, collision detection, dissection, rendering

## 1 Introduction

Laparoscopy surgery is a popular minimally invasive operation. It allows the surgeon to access the inside of the human body without having to make large incisions on the skin. However, due to the limitation of manipulation space and viewing angle, there is a

higher risk of damaging the internal organs, nerves and major arteries. For new surgeons, performing operations under the supervision of experienced surgeons becomes a practical solution in many scenarios. However, it will inevitably involve a long learning curve for the trainee to gain adequate skills to become a qualified surgeon. With the development of virtual reality technology, training surgeons with a VR based simulator has proven to be effective and can greatly reduce the both the risk to patients and the training costs.

Most of the current research on surgical simulation focuses on very specific topics (deformation [Székely et al. 2000], haptic rendering [Coles et al. 2011], dissection [Mor and Kanade 2000] etc.) rather than the practical framework, especially for laparoscopic surgical simulation. The best laparoscopic surgery simulators and frameworks in the market such as LapSim and SOFA [Allard et al. 2007] aim only for training basic skills following strict and relatively simplistic routines. They are not capable of simulating the whole surgery procedure because of the aforementioned unsolved issues. Getting familiar with the whole surgery process is essential not only for developing the surgeons perception in understanding the overall surgical procedure, but also allows them to plan and master the operation tasks in complex operations. There is a high demand from the surgery training market to develop a practical and efficient framework for education oriented surgical simulation.

In this paper, we are aiming to tackle some of the main challenging issues of laparoscopic surgery simulation, including modeling of complex anatomy structures, large deformation of soft tissues, surgical tool interactions, and the rendering of complex materials. We propose an efficient and stable framework integrating a set of specially tailored and designed techniques, ranging from rendering, deformation simulation, collision detection and soft tissue dissection. From a scientific perspective, the main contribution of this paper is the proposal of efficient, stable and fast converging deformation and interaction techniques. They are versatile in simulating complex anatomical structures composed of soft tissues with different properties. In particular, we are:

- Proposing a set of efficient, fast converging nonlinear large deformation techniques for soft tissue simulation. The biomechanical behaviours can be controlled by the designated parameters.
- Proposing an adaptive spherical collision detection and resolution method, which can improve the convergence rate and alleviate the collision tunnelling artefacts.

\*e-mail:KQian@bournemouth.ac.uk

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

VRST '15, November 13 – 15, 2015, Beijing, China.  
© 2015 ACM. ISBN 978-1-4503-3990-2/15/11...\$15.00  
DOI: <http://dx.doi.org/10.1145/2821592.2821599>

- Proposing an interactive haptic dissection approach using implicit shapes.
- Proposing a physically based rendering pipeline to visualize different materials, ranging from soft tissues with mucous layer to metal medical tools.

## 2 Related Works

### 2.1 Deformation Simulation

The simulation of deformable objects has been an active research topic in computer graphics area for decades. A few good surveys [Nealen et al. 2006][Bender et al. 2014] give a comprehensive overview of this area. For the deformable object simulation, force based dynamics and position based dynamics [Müller et al. 2007] are two widely used models. Mass spring and finite element method are typical force based methods which will cause overshoot under large time steps. Finite element is based on the continuum mechanics which provides an accurate computation model for elastic objects of different material property, but the main disadvantages of finite element method is its high computation cost. Position based dynamics has gained great popularity recently due to its simplicity, high efficiency and unconditional stability. The main feature of PBD is its direct control over vertex positions which eliminates the overshoot problem in force based method and simplifies the implementation process.

Simulating the whole surgery procedure using force based method is not practical because of the daunting simulation cost. Although PBD is not as accurate as physically based method, its stability and efficiency far outperform force based method in simulating the whole laparoscopic surgery procedure. Meanwhile, PBD has many control mechanisms to make it more flexible for the improvement on accuracy under the guidance and feedback from practitioner. It has the potential to achieve the same effect as real surgery. Balancing the pros and cons of stability, efficiency and accuracy, PBD is a good choice for laparoscopic surgery simulation.

### 2.2 Collision Detection

Collision detection is essential for an interactive real-time application but it could easily become the bottleneck for complex simulation. At coarse level, high efficient spatial data structure, such as bounding volume hierarchies (BVH), spatial hash, distance fields and image based method are proposed to accelerate the broad phase and narrow phase collision detection process. A comprehensive survey has been made in [Teschner et al. 2005][Weller 2013]. Our method only focuses on the finest level collision detection and resolution so it is compatible with all the above hierarchical spatial data structures and their optimization strategies.

One bottleneck of collision detection is the test of polygon intersection. The idea of using computationally efficient simplified representation for collision primitives is used to solve this problem. Sphere is the most used simplified representation due to its efficiency in overlapping test. For rigid body, generating sphere packing structure based on medial axis [Bradshaw and O’Sullivan 2002][Bradshaw and O’Sullivan 2004] and inner sphere tree [Weller and Zachmann 2009] are very popular. However, as the computation of these methods are expensive and mostly occurs in the initial configuration, they are not suitable for dynamic updating of deformable objects. Mendoza [Mendoza and OSullivan 2006] proposed a time-critical collision detection algorithm for deformable objects based on a sphere tree constructed using an adaptive medial-axis. Mendoza used detailed mesh for deformation and coarse mesh for sphere tree which is not good for han-

dling detailed collision. Also, this sphere representation and similar works [Weller and Zachmann 2009] does not approximate the original surface well. BD-tree (Bounded Deformation Tree) [James and Pai 2004] proposed an output sensitive collision detection for reduced deformable models, but it will suffer from over conservative bound, computational and memory overhead when use high resolution models. Pan [Pan et al. 2015b] proposed a metaball based collision detection method, but it can not provide adaptive approximation when mesh deformed. In this paper, we proposed a dynamically generated circumsphere structure which is not only computationally efficient for overlapping test but also can approximate the original mesh surface better than existing methods.

### 2.3 Soft Tissue Dissection

There are two types of dissection procedures for laparoscopic surgery training. One is called electrosurgical dissection. This procedure usually involves a membranous fat tissue which can be simplified as a 2D mesh. The other type is cutting a substantial soft tissue like a polyp, this type is called scalpel dissection. For electrosurgical dissection, tearing and cutting are commonly employed. Müller [Müller et al. 2007] first introduced tearing of cloth in PBD and Maciel [Maciel et al. 2009] had used this method in the laparoscopic surgery simulation. Pan [Pan et al. 2011] used triangle subdivision to simulate surface cutting procedure, but it is only suitable for coarse mesh. In this paper, we propose a surface cutting method suitable for mesh at any resolution using an unified implicit shape.

Volumetric mesh is commonly used in scalpel dissection. To obtain a smooth incision surface, element subdivision is a conventional way. Mor [Mor and Kanade 2000] subdivided the colliding tetrahedron into sub-elements at minimal quantity, and it has been employed by recent work [Courtecuisse et al. 2014] and [Pan et al. 2015a] for its practicability. Other dissection method for volumetric mesh can be found in survey [Wu et al. 2014]. In this paper, both electrosurgical dissection and scalpel dissection are provided.

### 2.4 Realistic Rendering

The scene inside the patient’s abdomen is complex because of the presence of various types of soft tissues, blood vessels and the dynamic glistening of the membrane under the illumination of the head light. Most previous works on the rendering of surgical simulation use the traditional shading models [ElHelw et al. 2004] or video image based method [Lim et al. 2007]. The traditional shading model needs physically plausible albedo, normal, specular and shininess (gloss) maps. The video image based method combined the image mosaicing and view-dependent texture-mapping techniques used the images obtained from video. However, the flexibility of those techniques are limited by the quality of the image. Due to the development of game industry, physically based rendering gradually replaced the traditional shading model and widely used in photo-realistic rendering [Hoffman 2010][Sébastien Lagarde 2014]. Physically based rendering (PBR) refers to the concept of using realistic shading models along with measured surface values to accurately represent real-world materials. Yet, little literature focuses on PBR in surgical simulation. In this paper, we will introduce a PBR based rendering pipeline for surgical simulation.

## 3 Deformation Simulation

The soft tissues involved in a laparoscopic surgery can be categorized into three types: volumetric tissue (such as the human organs), tubiform and glandular soft tissue (blood vessels, nerves and glands), fascia soft tissue (such as mesentery). In the following

sections, we will introduce how each type of soft tissues is handled with our simulation framework.

### 3.1 Volumetric Soft Tissue Simulation

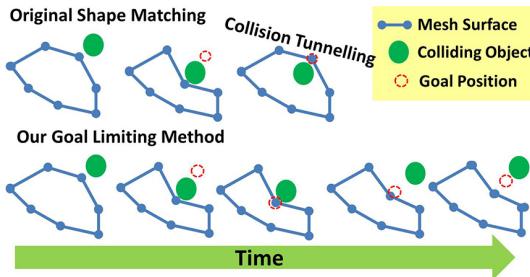
Most of the organs inside the abdomen will keep the shape during the surgery procedure. When a surgical tool contacts them, the deformation will only happen in the local contact area. The deformation algorithm needs to keep the global shape as well as the local deformation. If using volumetric mesh filled with tetrahedral elements, it will cost more computation and memory resources. If using a surface mesh, shape matching can preserve the shape but it cannot handle local deformation well because it provides a global transformation approximation. The core idea of shape matching is to find the optimal least square best-fit transformation from the rest state ( $\mathbf{x}_i^0$ ) to the current configuration ( $\mathbf{x}_i$ ) which equals to minimize  $\sum_{i=1}^n w_i (\mathbf{A}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) - (\mathbf{x}_i - \mathbf{x}_{cm}))^2$ , where  $n$  is the number of vertices of the mesh,  $\mathbf{x}_{cm}^0$  and  $\mathbf{x}_{cm}$  are the centre of mass of rest state and deformed state respectively,  $w_i$  are weights of individual vertices. The global transformation matrix  $\mathbf{A}$  can be calculated as:

$$\mathbf{A} = \sum_i^n m_i (\mathbf{x}_i - \mathbf{x}_{cm}) (\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) \quad (1)$$

The rotation matrix  $\mathbf{R}$  can be extracted from  $\mathbf{A}$  using polar decomposition  $\mathbf{A} = \mathbf{R}\mathbf{S}$  [Müller et al. 2005]. The goal position  $\mathbf{g}_i$  can be calculated using  $\mathbf{R}$  for each vertex  $i$ .

$$\mathbf{g}_i = \mathbf{R}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm} \quad (2)$$

Then each vertex is dragged towards its goal position to recover to the original shape. To achieve the local and non-linear deformation, cluster could be used to discretize the shape matching area. However, too many clusters will greatly influence the overall performance. On the other hand, less clusters cannot handle detailed local deformation well (such as collision with small object) because the optimal transformation matrix  $\mathbf{A}$  is a global rigid transformation matrix. When collision happened, obvious oscillation or collision tunneling will happen. The oscillation is caused by the repetitive violation of the collision constraint (unilateral constraint, which is solved only when it is violated) by shape matching. The collision tunneling is caused by the procedure of recovering to goal position. The goal position may be far from current position so that the recovery step may make the vertex run through thin object, resulting in collision tunneling. In this paper, we solve this problem by



**Figure 2:** Comparison between our method and shape matching. The goal position of our method is limited to the position of collided vertex and then gradually recover to the position of original shape

dynamically limiting the tendency of going back to the goal position. For a vertex  $\mathbf{x}_i$  and its predicted position  $\mathbf{p}_i$ , we interpolate its goal position  $\mathbf{g}_i$  and  $\mathbf{p}_i$  by a dynamic recovering factor  $\eta_i \in [0, 1]$ ,  $\mathbf{g}_i = \mathbf{g}_i + \eta_i(\mathbf{p}_i - \mathbf{g}_i)$ . When  $\eta_i = 0$ ,  $\mathbf{x}_i$  will directly move to the goal position. When collision happened we set  $\eta_i = 1$ , then

$\mathbf{g}_i = \mathbf{p}_i$  which means  $\mathbf{p}_i$  will not move immediately. After that it will be dragged to original shape gradually by recovering  $\eta_i$  at the user defined rate  $R_r$ ,  $\eta_i = \eta_i - R_r$  (See Figure 2). During the process, both  $\eta_i$  and  $(\mathbf{p}_i - \mathbf{g}_i)$  will decrease with the evolving of time which means the speed of recovering to original shape is decreasing with time (see Algorithm 1). This can well reflect the property of highly damped soft tissue and its shape recovery procedure. The value of  $\eta_i$  can be determined by the further evaluation process performed by surgeons.

---

#### Algorithm 1 Goal Limiting Shape Matching

---

```

1: Parameters: time step ( $h$ ), goal position limiting factor for  $\mathbf{x}_i^0$  ( $\eta_i$ ), vertex velocity ( $\mathbf{v}_i$ ), position in rest configuration ( $\mathbf{x}_i^0$ ), recovering rate ( $R_r$ ), stiffness factor ( $\alpha$ )
2: procedure GOAL_POSITION_LIMITING
3:   for all Vertex  $\mathbf{x}_i$  do
4:      $\mathbf{v}_i = \mathbf{v}_i + \alpha \frac{\mathbf{g}_i - \mathbf{x}_i}{h} + h\mathbf{f}_{ext}/m_i$ 
5:      $\mathbf{p}_i = \mathbf{x}_i + \mathbf{v}_i h$ 
6:   endfor
7:   for all cluster  $c_i \in C$  do
8:     Compute the mass center  $\mathbf{p}_{cm}^{c_i}$  for all  $\mathbf{p}_i \in c_i$ 
9:     Compute the mass center  $\mathbf{x}_{cm}^{0c_i}$  for all vertices  $\mathbf{x}_i^0 \in c_i$ 
10:    Compute  $\mathbf{A}_{c_i} = \sum_i^n m_i (\mathbf{p}_i - \mathbf{p}_{cm}^{c_i})(\mathbf{x}_i^0 - \mathbf{x}_{cm}^{0c_i})$ 
11:    Calculate  $\mathbf{R}_{c_i} = polarDecomposition(\mathbf{A}_{c_i})$ 
12:    for all  $\mathbf{p}_i \in c_i$  do
13:      if  $\mathbf{p}_i$  has been collided then
14:         $\eta_i = 1$ 
15:      else
16:         $\eta_i = max(\eta_i - R_r, 0)$ 
17:      endif
18:       $\mathbf{g}_i^{c_i} = \mathbf{R}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^{0c_i}) + \mathbf{p}_{cm}^{c_i}$ 
19:       $\mathbf{g}_i = \mathbf{g}_i^{c_i} + \eta_i(\mathbf{p}_i - \mathbf{g}_i^{c_i})$ 
20:    endfor
21:   for all cluster  $c_i \in C$  do
22:     Get the number of clusters ( $n_c$ ) sharing  $\mathbf{g}_i$ .
23:      $\mathbf{g}_i / n_c$ 
endfor
for all vertex  $\mathbf{x}_i$  do
24:    $\mathbf{v}_i = (\mathbf{p}_i - \mathbf{x}_i)/h$ 
25:    $\mathbf{x}_i = \mathbf{p}_i$ 
endfor

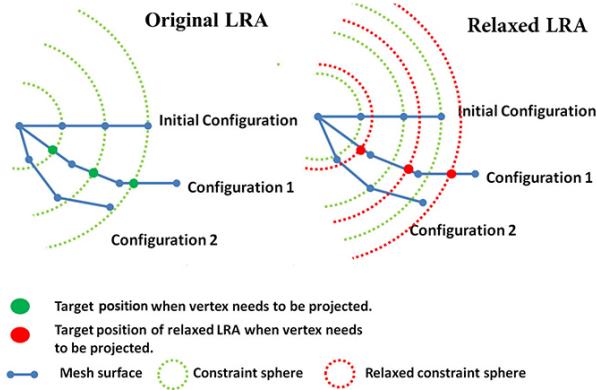
```

---

### 3.2 Fascia Simulation

The property of most fascia is not totally inextensible. The extensibility of fascia varied according to the way it connected to surrounding tissues. The portion of fascia which is far from attached tissue is stretchier than the part connected by other tissue. Low convergence rate of PBD will result in the stretchy looking of the fascia, making it like a piece of cloth. Long range attachment (LRA) [Kim et al. 2012] constraint is used to deal with the inextensibility of cloth in a fast converged manner. The core idea of LRA constraint is confining each unconstrained vertex within the initial distance of the attached vertices. The LRA method can be divided into two steps: Firstly, calculate an initial distance  $d_i^{LRA'}$  between unconstrained vertex ( $\mathbf{x}_i$ ) and attached vertex ( $\mathbf{x}_i^a$ ). Secondly, if the unconstrained vertex is out of range of the initial distance, project it back to the sphere centered at the attached vertex with radius of  $d_i^{LRA'}$ .

For each unconstrained vertex, it may be constrained by different attached vertices. The influence of each attached vertex to the unconstrained vertex can be averaged using Jacobian style. However, the fascia does not like inextensible cloth. If directly using LRA in fascia simulation, obvious artefact and oscillation will appear when

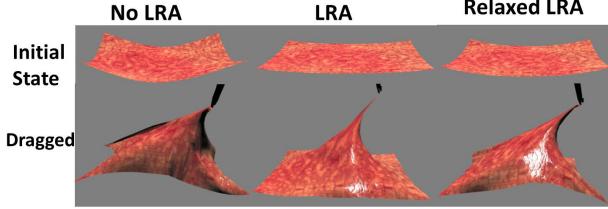


**Figure 3:** Comparison between original LRA and relaxed LRA

surgical tool drag the fascia. This is because the LRA constraint is too hard which conflicts with the drag constraint. In fact, the portion of fascia which is far from attached vertex is more stretching than the part near attached vertex. In order to take the advantage of the fast convergence rate of LRA, we make relaxation to the LRA constraint, making each vertex's tendency of stretching evolves with the distance to attach point ( $\mathbf{x}_i^a$ ) (See Figure 3):

$$d_i^{LRA} = d_i^{LRA'} + \beta \max((\|\mathbf{x}_i^a - \mathbf{x}_i\| - d_i^{LRA'}), 0) \quad (3)$$

where  $\beta$  is a user-defined parameter controlling the level of relaxation, which influences the stiffness of the fascia.  $d_i^{LRA}$  is the new LRA constraint distance between  $\mathbf{x}_i$  and  $\mathbf{x}_i^a$ . Such relaxation turns the hard LRA constraint into a soft constraint which can benefit the stability and convergence rate.

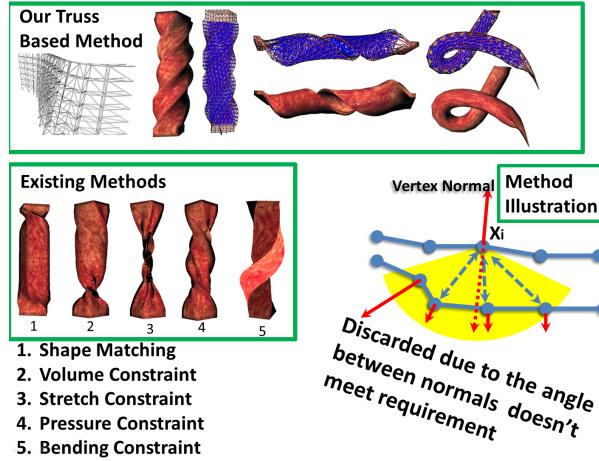


**Figure 4:** The improvement by using relaxed LRA

### 3.3 Tubiform and Glandular Soft Tissue Simulation

Compared to the major organs and fascias, the nonlinearity characteristic of tubiform and glandular soft tissue are more obvious. Due to the fact that neither the volume nor the shape of glandular remain constant, the volume or shape preservation constraint will not work efficiently for the simulation of such tissue. For the same reason, surface constraints such as bending, stretching, pressure are hard to achieve this effect (see Figure 5). The combination of the above constraints possibly works but appropriate combination strategy is hard to find. In this paper, we inspired by the truss structure[Smith et al. 2002] from structural engineering (see Figure 5), which is a triangle based support structure. We fill the tubiform and glandular soft tissue with elastic truss structure which means each strut is treated as a distance constraint with different stiffness.

Such designing cannot only support the shape of the soft tissue but also provide high movement flexibility by strut with different stiffness. In the pre-processing stage, we generate the truss structure as described in Algorithm 2: Firstly, localize potential strut pair.



**Figure 5:** Truss structure based deformation and comparison of the deformation results using existing constraints

For vertex  $\mathbf{x}_i$  with normal  $\mathbf{n}_i$ , we find the nearest face intersected by  $-\mathbf{n}_i$ . Using the vector determined by the face center and  $\mathbf{x}_i$  as axis to generate a cone with the user-defined angle  $\theta$ . Marking the vertices inside the cone as the potential strut pair (Line 2-9, also see Figure 5). Secondly, from the strut candidates, if cosine of the angle between the potential strut vertex's normal and  $\mathbf{n}_i$  satisfies the user-defined angle threshold ( $\theta_{threshold}$ ), then connect them as a strut. The value of  $\theta_{threshold}$  determines the bending feature of the object. During this process, the maximum and minimum strut length was recorded for stiffness coefficient calculation (Line 11-18). Thirdly, define the shortest strut as the most stiff one. Based on a user-defined minimum stiffness factor  $k_{min}$ , we make an interpolation between the shortest and longest struts to determine the stiffness factor for each strut (Line 19-22). For the detail of the algorithm, please refer to Algorithm 2.

## 4 Collision Detection and Resolution

In this part, we present an adaptive implicit circumsphere collision detection and resolution method based on the local geometry features and material properties, which is a much faster alternative for polygon collision tests. Under the PBD framework, we demonstrate that our method can effectively accelerate the convergence rate without causing noticeable visual artefacts. What's more, although our method can not eliminate collision tunnelling completely, our method can actually provide better prevention to collision tunnelling without applying any continuous collision detection because the structure of circumsphere actually provide a well approximated safety thickness for the thin surface, which is from the safety thickness provided by traditional methods.

### 4.1 Overview

In the broad phase, we use the spatial hash structure to localize the potentially colliding pairs due to its convenience to handle self-collision. Our method concentrates on the subsequent narrow phase where we separate it into two sub-phases, bounding sphere phase and circumsphere phase. For the bounding sphere phase, potentially colliding pairs are checked by a simple bounding sphere test of the triangle primitives. When the likelihood of collision is established, it moves on to the circumsphere phase, in which we test the overlap of the dynamically generated circumsphere for the underlying primitives and use the circumsphere for the succeeding collision resolution.

---

**Algorithm 2** Truss Structure Pre-processing

---

```

1: Input: vertex position ( $\mathbf{x}_i$ ), vertex normal ( $\mathbf{n}_i$ ), cone angle ( $\theta$ ),
   minimum strut stiffness ( $k_{min}$ )
2: procedure TRUSS_STRUCTURE_GENERATION
3:   for all vertex  $\mathbf{x}_i$  do
4:     for all Face  $\mathbf{f}_k$  do
5:       Calculate face normal  $\mathbf{nf}_k$ 
6:       if  $\text{dotproduct}(\mathbf{n}_i, \mathbf{nf}_k) < 0$  then
7:         Find the nearest face ( $\mathbf{f}_n$ ) intersected with  $-\mathbf{n}_i$ 
8:       endif
9:     endfor
10:     $\mathbf{d} = \mathbf{f}_n^{center} - \mathbf{x}_i$ , where  $\mathbf{f}_n^{center}$  is the center of  $\mathbf{f}_n$ 
11:    Using  $\mathbf{d}$  and user-defined  $\theta$  to generate a cone  $\kappa_i$ 
12:  endfor
13:  for all vertex  $\mathbf{x}_i$  do
14:    Initialize min and max strut length  $D_{min}$  and  $D_{max}$ 
15:    for all vertex  $j$  inside the cone  $\kappa_i$  do
16:      if  $\text{dotproduct}(\mathbf{n}_i, \mathbf{n}_j) < \theta_{threshold}$  then
17:        Connect  $\mathbf{x}_i$  with  $\mathbf{x}_j$  as strut;
18:        if  $\text{distance}(\mathbf{x}_i, \mathbf{x}_j) < D_{min}$  then
19:           $D_{min} = \text{distance}(\mathbf{x}_i, \mathbf{x}_j)$ 
20:        endif
21:        if  $\text{distance}(\mathbf{x}_i, \mathbf{x}_j) > D_{max}$  then
22:           $D_{max} = \text{distance}(\mathbf{x}_i, \mathbf{x}_j)$ 
23:        endif
24:      endif
25:    endfor
26:    for all vertex  $j$  inside the cone  $\kappa_i$  do
27:      Calculate the stiffness factor
28:       $k_{ij} = (\text{distance}(\mathbf{x}_i, \mathbf{x}_j) - D_{max}) / (D_{min} -$ 
29:            $D_{max}) + k_{min}$ 
30:    endfor
31:  endfor

```

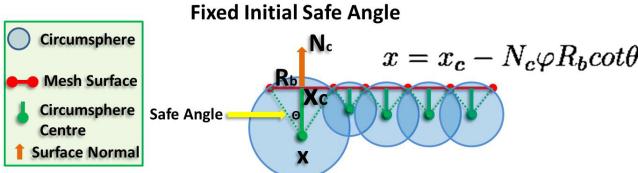
---

## 4.2 Adaptive Circumsphere Generation

The circumsphere is computed based on a shift of the circumsphere center of each triangle along the negative direction of the surface normal. Firstly, we generate a coarse circumsphere for potentially collided primitive. Then we adjust the size of the circumsphere according to the geometry and material property of the surface. We generate the coarse circumsphere according to the primitive size:

$$\mathbf{x} = \mathbf{x}_c - N_c \varphi R_b \cot \theta \quad (4)$$

where  $\mathbf{x}_c$  is the circumcenter of the corresponding triangle primitive,  $N_c$  is the normalized surface normal at  $\mathbf{x}_c$  and  $\varphi$  is a global scale factor for translating  $\mathbf{x}$  along  $-N_c$ ,  $R_b$  is the bounding sphere's radius,  $\theta$  is a fixed angle called safe angle as can be seen in Figure 6.



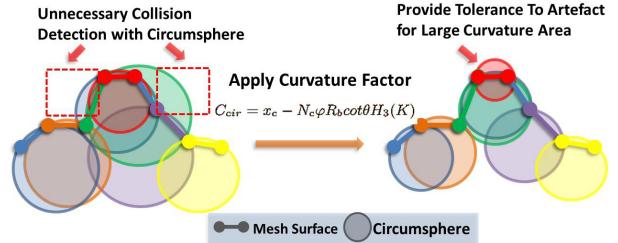
**Figure 6:** The generation of the initial circumsphere

### 4.2.1 Local Feature Based Circumsphere

However, such circumsphere approximation will become poor when the mesh is under deformation. To improve the approximation accuracy, we take the surface curvature as a factor to influ-

ence the position of the circumsphere. To cater for the need of real-time response, we adopt a computationally efficient dual mesh [Wu and Tai 2010] based structure to approximate the curvature of the original triangle mesh surface. We connect the geometric centre of the neighbour faces to construct a dual mesh structure and use the curvature of the dual mesh's vertices to approximate the curvature of its corresponding original surface. The Gauss-Bonnet scheme is used to approximate the local curvature of the dual mesh vertex  $\mathbf{x}_i^{dual}$  [Surazhsky et al. 2003]. To integrate the influence of curvature into the position of circumsphere, we use cubic Hermite  $H_3(K)$  to interpolate the curvature factor in the range of [0,1] (see Equation 5 and Figure 7). When  $K = 0$ , we use the initial circumsphere to approximate the mesh surface ( $H_3(K)=1$ ). When  $K \rightarrow \infty$ , we let the circumsphere shrink to the bounding sphere of the primitive ( $H_3(K) = 0$ ) which actually provides good tolerance to artefact and reduce unnecessary collision detection. The cubic Hermite can be written as:  $H_3(x) = y_0 h_0(x) + y_1 h_1(x) + y'_0 H_0(x) + y'_1 H_1(x)$ , where  $(x_0, y_0) = (0, 1)$ ,  $(x_1, y_1) = (\infty, 0)$ ,  $y'_0(x_0) = 0$ ,  $y'_1(x_1) = 0$ ,  $h_0(x)$ ,  $h_1(x)$ ,  $H_0(x)$ ,  $H_1(x)$  are the Hermite basis functions. This interpolation mapping function can be calculated at the initial configuration and used directly afterwards.

$$C_{cir} = x_c - N_c \varphi R_b \cot \theta H_3(K) \quad (5)$$



**Figure 7:** Adjust the size of circumsphere according to curvature

### 4.2.2 Material Property Based Circumsphere

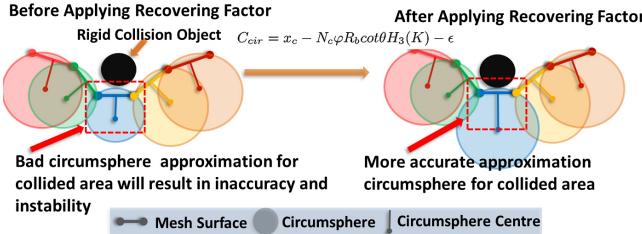
For a deformable object, the surface geometry feature changes all the time especially when collision occurs. When the collided area is subject to large deformation, the circumsphere's size should be larger for this area to approximate the deformed surface better and provide more accurate collision response. Elasticity reflects the tendency of recovering to initial state so that we use a scalar factor  $\epsilon_i$  ( $i = 1, 2, 3$ ) for each vertex of triangle primitive to dynamically adjust the size of circumsphere according to material property. Under PBD framework, for each vertex  $\mathbf{x}_i$ , we define  $d_i$  as the distance between its current position and goal position (refer to shape matching constraint in section 3.1).  $\|d_i\|$  can well reflect elasticity. We use  $\|d_i\|$  as  $\epsilon_i$  to add the impact of material property to the size of the circumsphere. As can be seen in Figure 8, the circumsphere centre can be modified as:

$$C_{cir} = x_c - N_c \varphi R_b \cot \theta H_3(K) - \epsilon_i \quad (6)$$

We calculate the triangle surface recovering tendency  $\epsilon$  by averaging its vertices tendency  $\epsilon = (\epsilon_i + \epsilon_j + \epsilon_k)/3$ , where  $i, j, k$  are the indices of vertices of the triangle primitive.

### 4.2.3 Updating of Circumsphere

Instead of updating each frame, we update the structure only when the shape change of its underlying primitive exceeds a threshold



**Figure 8:** Adjust the size of circumspHERE according to material property

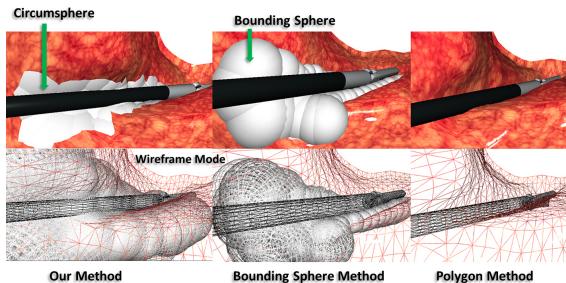
$\delta$  compared to last frame. We use the change of primitive area to measure the rate of shape change  $h_{shape} = S_{current}/S_{last}$ , where  $S_{current}$  and  $S_{last}$  are the area of primitive in current frame and last frame respectively. If  $|h_{shape} - 1| < \delta$ , we just update the centre of corresponding circumspHERE using the same curvature factor and recovering factor in last frame. When  $|h_{shape} - 1| \geq \delta$ , we update the curvature factor and recovering factor. There is no denying that such updating method will affect accuracy but has little influence on the visual plausibility. The tradeoff is the improvement of real-time performance.

### 4.3 Collision Resolution

When collision happened, we bounced the collided circumspHERE pair and its underlying triangle primitive away. Firstly, we find the state which collision just happened. Secondly, update the velocity for each sphere using the principle of linear impulse and momentum. Finally, along the direction of new velocity, move each sphere to final position. Then update the position of sphere's underlying triangle primitive using the translation of sphere.

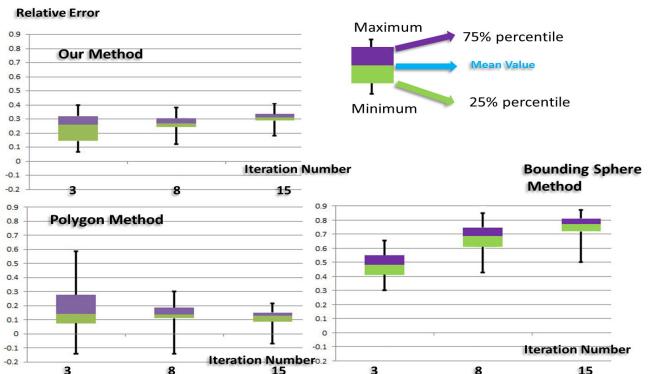
### 4.4 Method Analysis

We compare our method with polygon based method [Ericson 2004] and bounding sphere based method [Mendoza and OSullivan 2006] from accuracy, stability and efficiency perspectives. In order to compare with ground truth, we devised a special experiment of collision detection between a rigid cylinder and a deformable soft tissue as can be seen in Figure 9. Thus the exact distance for collided part and the main axis of rigid cylinder should be the cylinder's radius. This exact distance ( $d_{exa}$ ) will be compared in the following experiments.



**Figure 9:** Overview of the comparison between our method and other methods

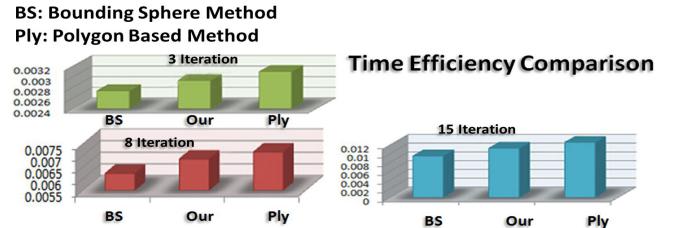
For accuracy comparison, we use the average distance ( $d_{ave}$ ) from collided vertices to the main axis of the cylinder to measure accuracy. In Figure 10, we use the box plot graph to demonstrate the



**Figure 10:** Method analysis from stability, accuracy perspective

relative error of average distance  $(d_{ave} - d_{exa})/d_{exa}$ . When iteration is low, although the polygon based method is the most accurate, the negative relative errors indicate collided primitive will run into collision object. Although such artefacts can be avoided by using collision margin, safe bound or sweep sphere, they are fixed rather than adaptive, which provides bad approximation compared to our method. Bounding sphere method is the most inaccurate method because its poor approximation of the mesh surface.

For the stability, the range of the maximum and minimum value and the proportion distribution of the box plot graph (Figure 10) can give a good reflection of position variation, which can be perceived as stability. The polygon and bounding sphere based methods are unstable under small iteration, which is caused by the high variability of polygon shape and poor approximation of mesh surface respectively. Our method is the most stable because the circumspHERE structure cannot only provides good approximation of the mesh surface but also adds fewer unstable factors to the constraint solving process.



**Figure 11:** Efficiency comparison between our method and others

Time consumed by collision detection and resolution in each frame is a good measurement of algorithm efficiency. As can be seen in Figure 11, the bounding sphere method is the most efficient because it only needs to update the centre, radius and perform overlap test when collision happens. Our method needs re-calculation of circumspHERE when the shape change of underlying primitive exceeds a given threshold. The polygon intersection based method is the most time consuming.

## 5 Soft Tissue Dissection with Implicit Shape

There are two types of dissection procedures in laparoscopic surgery: electrosurgical dissection and scalpel dissection. Electrosurgical dissection mainly used to split thin soft tissue like membrane (surface mesh). Scalpel dissection is used to cut solid soft tissue (volumetric mesh). In the field of computer graphics, tearing

and cutting are two major methods for dissection simulation. Tearing splits the originally connected vertices apart. Cutting changes the topology of the original mesh by a cutting geometry. In this section, we proposed a generalized dissection method based on implicit shape.

### 5.1 Implicit Shape for Surgical Instrument

Rather than a simple cutting plane, the real surgical instruments used for dissection have various shapes, such as column, sphere, polyhedron etc. Considering the applicability of implicit shape, we employ a combination of implicit shapes to represent the dissection part of the surgical instrument. Then dissection simulation with different surgical instruments can be unified into a general framework. During dissection, we move the surgical instrument and generate a sweep volume by connecting the implicit shapes at different times. The surface mesh or the volumetric mesh is dissected by the sweep volume. The details of the dissection method will be introduced in the following sections.

### 5.2 Surface Mesh Dissection

For the surface mesh dissection, we integrate both cutting and tearing. We employ implicit shape to generalize the cutting procedure. The surface mesh is composed of triangle faces. The algorithm is summarized in Algorithm 3. First, we generate the sweep volume  $V_{sweep}$ . Then we compute intersections and remove all the faces inside  $V_{sweep}$  (line 3-5). For each intersected face  $f_j$ , we compute the area  $A_j$  and subdivide it into a set of small triangles  $\{f_{jk}\}$ . From line 10-15, we remove  $f_{jk}$  if it is inside  $V_{sweep}$  or it does not satisfy a user defined threshold  $v_{threshold}$  (see Algorithm 3). Figure 12 illustrates the process of surface cutting. In Figure 12, the purple triangles are the primitives intersected with sweep volume. After cutting, the green triangles which are contained in the sweep volume are deleted, also the yellow triangles which are too small are deleted.

---

#### Algorithm 3 Surface Cutting

---

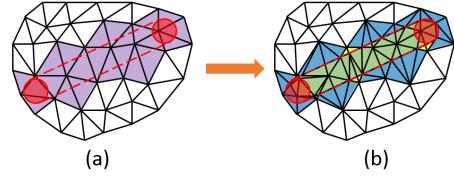
```

1: procedure SURFACE CUTTING
2:   Generate the sweep volume  $V_{sweep}$ 
3:   Compute intersections
4:   for all face  $f_i$  inside sweep volume  $V_{sweep}$  do
5:     Delete  $f_i$  from the mesh
end for
6:   for all intersected face  $f_j$  do
7:     Compute the area  $A_j$  for face  $f_j$ 
8:     Subdivide face  $f_j$  into small triangles  $\{f_{jk}\}$ 
9:     for all newly generated face  $f_{jk}$  do
10:      if face  $f_{jk}$  is inside  $V_{sweep}$  then
11:        Delete face  $f_{jk}$ 
12:      else
13:        Compute the area  $A_{jk}$  for  $f_{jk}$ 
14:        if  $A_{jk}/A_j < v_{threshold}$  then
15:          Delete face  $f_{jk}$ 
end if
end for
end for

```

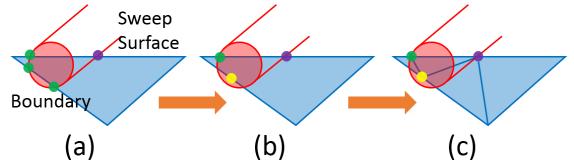
---

In real surgery environment, the most commonly used shape for surgical instruments in surface dissection is column. A sweep surface is generated by connecting the shapes at different times. The intersections on the sweep surface are intuitive. There may be more than one intersections on the shape's boundary. If there are more than one boundary intersections and no sweep surface intersection, we combine these intersections into a median point to represent the



**Figure 12:** The process of surface cutting. (a)The intersected triangles are in purple color. (b)After cutting, the green triangles (inside  $V_{sweep}$ ) and yellow triangles (too small) are removed, blue triangles are reserved.

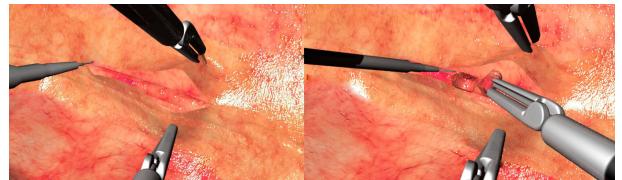
intersection on the boundary as Figure 13 shows. Then we subdivide the intersected triangle. Compared with previous works, our method is suitable for mesh at any resolution. For the tearing of the surface, we utilize the method proposed by Müller [Müller et al. 2007].



**Figure 13:** Combination and subdivision. (a)The green points are the intersections on the shape's boundary, the purple point is the intersection on the sweep surface. (b)The boundary intersections on the same edge are combined into a median point. (c)The triangle primitive is subdivided according to these intersections.

### 5.3 Volumetric Mesh Dissection

We use volumetric mesh dissection to exsect a polyp on the tissue surface. The volumetric mesh is composed of tetrahedra elements. When dissecting the volumetric mesh, line is selected as the implicit shape for surgical instrument, and the sweep volume turns into a sweep surface. The dissection algorithm for volumetric mesh is similar to surface cutting: finding the intersections and subdividing the original primitives. There are two types of intersections between the tetrahedral mesh and sweep surface: edge intersections and face intersection. We encode the edges and faces of tetrahedron in binary, and the intersecting state can be represented by a tuple of edge code and face code when there is an intersection. Then the intersected tetrahedra are split into small elements according to the intersecting state. We employed the method in [Mor and Kanade 2000] to subdivide the tetrahedron. Figure 14 shows the dissection of fascia and volumetric mesh in the simulator.



**Figure 14:** The dissection of fascia and volumetric mesh.

## 6 Realistic Rendering

There are several models of surface BRDFs. The empirical model (Blinn, Phong etc.) and microfacet model are the most widely used.

The empirical model is a fast computational model adjustable by parameters, but without considering the physics behind it. The microfacet model is inspired by real physical processes. Due to the complex anatomical structures and their glistening effect under light, our material rendering is based on the microfacet model.

## 6.1 Material Model

The microfacet model can be decomposed into the diffuse part ( $f_d$ ) and specular part ( $f_r$ ). This model can be described in the general Equation 7, each component and parameter can be modified according to practical needs:

$$f_{d/r} = \frac{\int_{\Omega} (f_m(\mathbf{v}, \mathbf{l}, \mathbf{h}, \mathbf{S}) G(\mathbf{v}, \mathbf{l}, \mathbf{S}) D(\mathbf{S}, \alpha)) (\mathbf{v} \cdot \mathbf{S})(\mathbf{l} \cdot \mathbf{S})) d\mathbf{S}}{|\mathbf{n} \cdot \mathbf{v}| |\mathbf{n} \cdot \mathbf{l}|} \quad (7)$$

$\mathbf{n}$  is the surface normal,  $\mathbf{l}$  is the lighting direction,  $\mathbf{v}$  is the viewing direction,  $\mathbf{h}$  is the half vector, the integration  $\int_{\Omega} d\mathbf{S}$  indicates an integral over a hemisphere of all directions,  $D(\mathbf{S}, \alpha)$  is the normal distribution (NDF) function,  $G(\mathbf{v}, \mathbf{l}, \mathbf{S})$  is the geometry attenuation term and  $f_m(\mathbf{v}, \mathbf{l}, \mathbf{h}, \mathbf{S})$  is the Fresnel term. The difference between  $f_d$  and  $f_r$  is dependent on the Fresnel term  $f_m$ . For the specular term  $f_r$ , it can be written as the general Cook-Torrance form:

$$f_r(\mathbf{v}, \mathbf{l}, \mathbf{h}, \alpha) = \frac{F(\mathbf{v}, \mathbf{l}) G(\mathbf{v}, \mathbf{l}, \mathbf{h}, \alpha) D(\mathbf{h}, \alpha)}{4 \langle \mathbf{n} \cdot \mathbf{v} \rangle \langle \langle \mathbf{n} \cdot \mathbf{l} \rangle \rangle} \quad (8)$$

The term  $D$  determines the appearance of surfaces. To capture the real world better, we use the GGX distribution [Burley 2012] which can produce realistic "long tailed" effect.

$$D(\mathbf{h}, \alpha) = \frac{\alpha^2}{\pi (\langle \mathbf{n} \cdot \mathbf{h} \rangle^2 (\alpha^2 - 1) + 1)^2} \quad (9)$$

Where  $\alpha = \text{roughness}^2$ . The term  $G$  is used for describing how much the microfacet is occluded by others. Heitz et al.[Heitz 2014] point out that approximating Smith visibility function using Schlick model [Schlick 1994] and Disney's modification [Burley 2012] are not accurate enough. The height-correlated Smith function which can model the correlation between the masking and shadowing according to the height of the microfacet is a good choice.

$$G(\mathbf{v}, \mathbf{l}, \mathbf{h}, \alpha) = \frac{\chi^+(\mathbf{v} \cdot \mathbf{h}) \chi^+(\mathbf{l} \cdot \mathbf{h})}{\Lambda(\mathbf{v}) + \Lambda(\mathbf{l})} \quad (10)$$

Where  $\Lambda(\mathbf{x}) = \frac{-1 + \sqrt{1 + \alpha^2 \tan^2(\theta_x)}}{2}$ ,  $\theta_x$  is the angle between normal and  $\mathbf{x}$ ,  $\chi^+(a)$  is the Heaviside function: return 1 if  $a > 0$ , return 0 if  $a \leq 0$ .

The typical choice for the Fresnel term  $F$  is Schlick's approximation [Schlick 1994]. We utilize a spherical Gaussian approximation method proposed in [Tsai and Shih 2006] which is more computationally efficient for hardware. We modify the term  $F$  as:

$$F(\mathbf{v}, \mathbf{h}) = F_0 + (1 - F_0) \times 2^{-6.745372(\langle \mathbf{v} \cdot \mathbf{h} \rangle)^2 + 35.324156(\mathbf{v} \cdot \mathbf{h})} \quad (11)$$

For the diffuse term  $f_d$ , we take the form of Disney's empirical method which takes the roughness of material into account and generates retro-reflection at grazing angles. This model can approximate the main features of the MERL database's material.

$$f_d(\mathbf{v}, \mathbf{l}, \mathbf{h}, \alpha) = \frac{(1 + (F_{D90} - 1)(1 - \langle \mathbf{n} \cdot \mathbf{v} \rangle)^5)(1 + (F_{D90} - 1)(1 - \langle \mathbf{n} \cdot \mathbf{l} \rangle)^5)}{\pi / \rho} \quad (12)$$

Where  $F_{D90} = 0.5 + \cos(\langle \mathbf{l} \cdot \mathbf{h} \rangle)^2 \alpha$ . In real world, the energy received by material is no less than the reflected energy. The energy conservation can cope with the effect at grazing angle where the light is tend to be scattered more. To keep the computation efficiency, we adopt the Disney method which multiply the Fresnel reflectance  $(1 - F(\mathbf{n}, \mathbf{l}))(1 - F(\mathbf{l}, \mathbf{v}))$  to Equation 12.

## 6.2 Lighting Model

Under the effect of light, some soft tissues will reflect the surrounding environment. Image based lighting (IBL) allows us to represent the incident light surrounding a point by an environment map, making the object fit into the environment without expensive lighting computation. To achieve this, incident light should be in consistent with BRDF equation especially for layered material. However, the computation of the interaction between IBL ( $L$ ) and BRDF material ( $f$ ) is a costly operation which needs to integrate all the direction of the light:

$$L(\mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) L(\mathbf{l}) d\mathbf{l} \quad (13)$$

To improve efficiency, pre-integration can be used for IBL computation. For the specular lighting computation, the Equation 13 can be approximated using the method proposed in [Kasyan 2013]:

$$L_r(\mathbf{v}) \approx \sum_i^N (F(\mathbf{v}, \mathbf{h}) G(\mathbf{l}, \mathbf{v}, \mathbf{h}, \alpha) \langle \mathbf{v} \cdot \mathbf{h} \rangle) \sum_i^N (L(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle) \quad (14)$$

Where  $N$  is the number of light probes. We can see that  $L_r(\mathbf{v})$  is approximated by two summation term, called DFG term and LD term respectively, both can be precomputed separately. Similarly, for diffuse material, it can be calculated as:

$$L_d(\mathbf{v}) \approx \sum_i^N \frac{f_d(\mathbf{v}, \mathbf{h}, \mathbf{l}, \alpha)}{\pi} \sum_i^N (L(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle) \quad (15)$$

$\sum_i^N (L(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle)$  can be efficiently calculated using spherical harmonic method [Ramamoorthi and Hanrahan 2001]. Spherical harmonics uses a frequency space to represent an image over a sphere, which is continuous and rotationally invariant. The usage of this representation can produce accurate diffuse reflection from a surface using nine spherical harmonic basis functions [Ramamoorthi and Hanrahan 2001]. Figure 15 is the demonstration of objects using different material under different roughness.

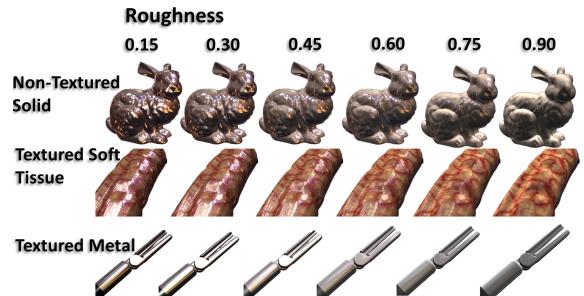
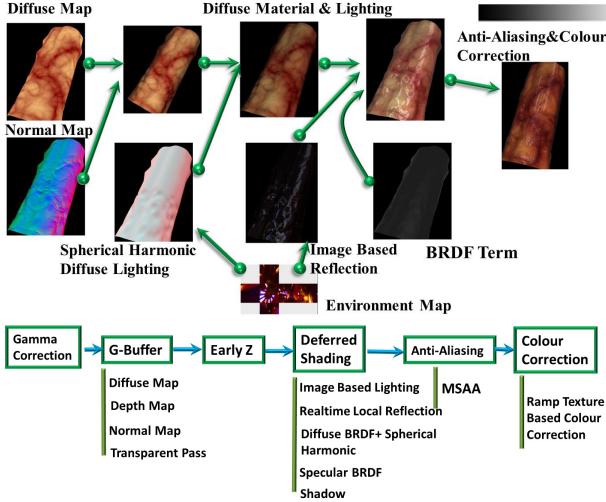


Figure 15: Demonstration of objects using different material

## 6.3 Rendering Pipeline

For the rendering component of a game engine, the most time consuming part is the post-processing stage, such as SSAO, depth of

field, HDR, soft shadow and motion blur etc. Those post-processing techniques have one thing in common: expensive pixel operation. Excessive pixel sampling operation may cause the sampler stall phenomenon which means that GPU is too busy to handle all the sampling instruction. Considering the practical scene and lighting complexity of laparoscopic surgery, our reduced deferred shading based rendering pipeline focuses on the lighting and material rendering. Our pipeline will not consume much computational resource but provide high quality rendering effect for laparoscopic surgery. Figure 16 shows an example of the rendering procedure and the structure of the whole rendering pipeline.

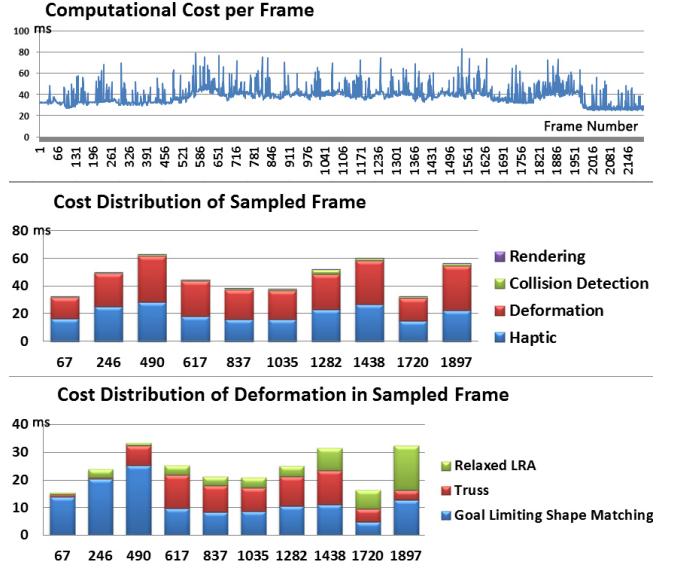


**Figure 16:** Overview of the rendering procedure and pipeline

## 7 Implementation and Results

Implementing the above discussed techniques, we have built a rectum cancer surgery simulator using C++, OpenGL and OpenHaptics which runs on a desktop with a Xeon 5 CPU and Quadro K2000 GPU. We use two Omni Phantoms haptic devices running at the rate of 1KHz. To determine the optimal values of the control parameters of the physical simulation, we have tested our system with the help of experienced practitioners. Our aim was to achieve the same or very similar effect in the simulation of the operations: goal limiting factor (section 3.1)  $\eta = 0.7$ , level of relaxation (section 3.2)  $\beta = 0.1 \sim 0.4$ , bending threshold (section 3.3)  $\theta_{threshold} = \pi/3$ , minimum stiffness factor (section 3.3)  $k_{min} = 0.05$ , safe angle (section 4.2)  $\theta = \pi/6$ , coarse uniform factor (section 4.2)  $\psi = 3$ , eliminating threshold (section 5.2)  $v_{threshold} = 0.15$ .

To evaluate the computational cost, we have designed an experiment to include all the key operations for the rectum surgery. We recorded the time cost while performing the surgical operation (see Figure 17). We analyzed the cost distribution of each component such as graphical rendering, collision detection, deformation and haptic rendering. In detail, we also measured the cost of major deformation algorithms such as relaxed LRA, truss and goal limiting shape matching. The computational cost for deformation is dependent on the current surgical operation and operative site. In the initial stage of the simulation (frame 1 to 450), the main interaction is between the large volumetric organ and surgical tools so that the goal limiting shape matching is the major deformation algorithm. From frame 450 to 1500, the operative site focused more on the separation of tubiform and glandular soft tissues. The truss based deformation algorithm begins to play its role at this stage.



**Figure 17:** Computational cost distribution analysis

From frame 1500 to 2146, mobilizing bilateral sides of the colon from its connected fascia is the main operation, which is mainly simulated by relaxed LRA. Successive over-relaxation technique [Macklin et al. 2014] has been used to average each kind of deformation constraint in a soft manner. Such over-relaxation propagates constraint corrections faster which will benefit convergence speed.

## 8 Future Works

To further improve our system, there are still a few aspects needing to be developed in the future. Currently, our constraint solving procedure consumes most of the computation resources. Migrating this procedure onto GPU using parallel computing could further improve the performance of the system. Yet, this system does not support smoke and fluid simulation which can further enhance the visual effects making the simulation more immersive and realistic.

## References

- ALLARD, J., COTIN, S., FAURE, F., BENSOUSSAN, P.-J., POYER, F., DURIEZ, C., DELINGETTE, H., AND GRISONI, L. 2007. Sofa—an open source framework for medical simulation. In *MMVR 15—Medicine Meets Virtual Reality*, vol. 125, IOP Press, 13–18.
- BENDER, J., ERLEBEN, K., AND TRINKLE, J. 2014. Interactive simulation of rigid body dynamics in computer graphics. *Comput. Graph. Forum* 33, 1, 246–270.
- BRADSHAW, G., AND O’SULLIVAN, C. 2002. Sphere-tree construction using dynamic medial axis approximation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 33–40.
- BRADSHAW, G., AND O’SULLIVAN, C. 2004. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics (TOG)* 23, 1, 1–26.
- BURLEY, B. 2012. Physically-based shading at disney. In *SIGGRAPH ’12: ACM SIGGRAPH 2012 Courses*, ACM.

- COLES, T., MEGLAN, D., AND JOHN, N. W. 2011. The role of haptics in medical training simulators: a survey of the state of the art. *Haptics, IEEE Transactions on* 4, 1, 51–66.
- COURTECUISSE, H., ALLARD, J., KERFRIDEN, P., BORDAS, S. P., COTIN, S., AND DURIEZ, C. 2014. Real-time simulation of contact and cutting of heterogeneous soft-tissues. *Medical Image Analysis* 18, 2, 394 – 410.
- ELHELW, M. A., LO, B. P., DARZI, A., AND YANG, G.-Z. 2004. Real-time photo-realistic rendering for surgical simulations with graphics hardware. In *Medical Imaging and Augmented Reality*. Springer, 346–352.
- ERICSON, C. 2004. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- HEITZ, E. 2014. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques* 3, 2, 32–91.
- HOFFMAN, N. 2010. Background: Physically-based shading. In *SIGGRAPH '10: ACM SIGGRAPH 2010 Courses*, ACM.
- JAMES, D. L., AND PAI, D. K. 2004. Bd-tree: output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (TOG)* 23, 3, 393–398.
- KASYAN, N. 2013. Playing with real-time shadows. In *SIGGRAPH '13: ACM SIGGRAPH 2013 Courses*, ACM.
- KIM, T., CHENTANEZ, N., AND MÜLLER-FISCHER, M. 2012. Long range attachments - A method to simulate inextensible clothing in computer games. In *Proceedings of the 2012 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2012, Lausanne, Switzerland, 2012*, 305–310.
- LIM, Y.-J., JIN, W., AND DE, S. 2007. On some recent advances in multimodal surgery simulation: A hybrid approach to surgical cutting and the use of video images for enhanced realism. *Presence* 16, 6, 563–583.
- MACIEL, A., HALIC, T., LU, Z., NEDEL, L., AND DE, S. 2009. Using the physx engine for physics-based virtual surgery with force feedback. *International Journal of Medical Robotics and Computer Assisted Surgery* 5, 3, 341–353.
- MACKLIN, M., MÜLLER, M., CHENTANEZ, N., AND KIM, T. 2014. Unified particle physics for real-time applications. *ACM Trans. Graph.* 33, 4, 153:1–153:12.
- MENDOZA, C., AND OSULLIVAN, C. 2006. Interruptible collision detection for deformable objects. *Computers & Graphics* 30, 3, 432–438.
- MOR, A., AND KANADE, T. 2000. In *Medical Image Computing and Computer-Assisted Intervention C MICCAI 2000*, vol. 1935. Springer Berlin Heidelberg, 598–607.
- MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. In *ACM Transactions on Graphics (TOG)*, vol. 24, ACM, 471–478.
- MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (Apr.), 109–118.
- NEALEN, A., MLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum* 25, 809–836.
- PAN, J. J., CHANG, J., YANG, X., ZHANG, J. J., QURESHI, T., HOWELL, R., AND HICKISH, T. 2011. Graphic and haptic simulation system for virtual laparoscopic rectum surgery. *International Journal of Medical Robotics and Computer Assisted Surgery* 7, 3, 304–317.
- PAN, J., BAI, J., ZHAO, X., HAO, A., AND QIN, H. 2015. Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics. *Computer Animation and Virtual Worlds* 26, 3-4, 321–335.
- PAN, J., ZHAO, C., ZHAO, X., HAO, A., AND QIN, H. 2015. Metaballs-based physical modeling and deformation of organs for virtual surgery. *The Visual Computer*, 1–11.
- RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 497–500.
- SCHLICK, C. 1994. An inexpensive BRDF model for physically-based rendering. *Comput. Graph. Forum* 13, 3, 233–246.
- SÉBASTIAN LAGARDE, C. d. R. 2014. Moving frostbite to physically based rendering. In *SIGGRAPH '14: ACM SIGGRAPH 2014 Courses*, ACM.
- SMITH, J., HODGINS, J., OPPENHEIM, I., AND WITKIN, A. 2002. Creating models of truss structures with optimization. In *ACM Transactions on Graphics (TOG)*, vol. 21, ACM, 295–301.
- SURAZHSKY, T., MAGID, E., SOLDEA, O., ELBER, G., AND RIVLIN, E. 2003. A comparison of gaussian and mean curvatures estimation methods on triangular meshes. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1, IEEE, 1021–1026.
- SZÉKELY, G., BRECHBÜHLER, C., HUTTER, R., RHOMBERG, A., IRONMONGER, N., AND SCHMID, P. 2000. Modelling of soft tissue deformation for laparoscopic surgery simulation. *Medical Image Analysis* 4, 1, 57–66.
- TESCHNER, M., KIMMERLE, S., HEIDELBERGER, B., ZACHMANN, G., RAGHUPATHI, L., FUHRMANN, A., CANI, M.-P., FAURE, F., MAGNENAT-THALMANN, N., STRASSER, W., ET AL. 2005. Collision detection for deformable objects. In *Computer graphics forum*, vol. 24, 61–81.
- TSAI, Y., AND SHIH, Z. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.* 25, 3, 967–976.
- WELLER, R., AND ZACHMANN, G. 2009. A unified approach for physically-based simulations and haptic rendering. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, ACM, 151–159.
- WELLER, R. 2013. A brief overview of collision detection. In *New Geometric Data Structures for Collision Detection and Haptics*. Springer, 9–46.
- WU, C., AND TAI, X. 2010. A level set formulation of geodesic curvature flow on simplicial surfaces. *Visualization and Computer Graphics, IEEE Transactions on* 16, 4 (July), 647–662.
- WU, J., WESTERMANN, R., AND DICK, C. 2014. Physically-based simulation of cuts in deformable bodies: A survey. In *Eurographics 2014 - State of the Art Reports, Strasbourg, France, 2014*, 1–19.