ESSENTIAL TECHNIQUES FOR IMPROVING VISUAL REALISM OF LAPAROSCOPIC SURGERY SIMULATION

Kun Qian

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of

Doctor of Philosophy

Bournemouth
University

Sep 2017

Copyright Statement

Abstract

With the prevalence of laparoscopic surgery, the request for reliable training and assessment is becoming increasingly important. The traditional way of training is both time consuming and cost intensive, and may cause ethical or moral issues. With the development of computer technologies, virtual reality has entered the world of consumer electronics as a new way to enhance tactile and visual sensory experiences. Virtual reality based surgical skill training gradually becomes an effective supplementary to the traditional laparoscopic skill training in many surgical theatres.

To provide high fidelity virtual surgery training experiences, the presentation of the virtual world should have the same level of realism as what surgeons see and feel during real operations. However, the weak computing power limits the potential level of details on the graphics presentation and physical behaviour of virtual objects, which will further influence the fidelity of tactile interaction. Achieving visual realism (realistic graphics presentation and accurate physical behaviour) and good user experience using limited computing resources is the main challenge for laparoscopic surgery simulation.

The topic of visual realism in laparoscopic surgery simulation has not been well researched. This topic mainly relates to the area of 3D anatomy modeling, soft body simulation and rendering. Current

researches in computer graphics and game communities are not tailored for laparoscopic surgery simulation. The direct use of those techniques in developing surgery simulators will often result in poor quality anatomy model, inaccurate simulation, low fidelity visual effect, poor user experience and inefficient production pipeline, which significantly influence the visual realism of the virtual world. The development of laparoscopic surgery simulator is an interdiscipline of computer graphics, computational physics and haptics. However, current researches barely focus on the study of tailored techniques and efficient production pipeline which often result in the long term research cycle and daunting cost for simulator development.

This research is aiming at improving the visual realism of laparoscopic surgery simulation from the perspective of computer graphics. In this research, a set of tailor techniques have been proposed to improve the visual realism for laparoscopic surgery simulation. For anatomy modeling, an automatic and efficient 3D anatomy conversion pipeline is proposed which can convert bad quality 3D anatomy into simulation ready state while preserving the original model's surface parameterization property. For simulation, a soft tissue simulation pipeline is proposed which can provide multi-layer heterogeneous soft tissue modeling and intuitive physically editable simulation based on uniform polynomial based hyperelastic material representation. For interaction, a collision detection and interaction system based on adaptive circumphere structure is proposed which supports robust and efficient sliding contact, energized dissection and clip. For rendering, a multi-layer soft tissue rendering pipeline is proposed which decomposed the multi-layer structure of soft tissue into corresponding material asset required by state-of-art rendering techniques. Based on this research, a system framework for building a laparoscopic surgery simulator is also proposed to test the feasibility of those tailored techniques.

Contents

List of Figures

List of Publications and Awards

Conferences

Qian, K., Bai, J., Yang, X., Pan, J., Zhang, J., Virtual reality based laparoscopic surgery simulation[C], Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology. ACM, 2015: 69-78. **Cited by**: 4

Qian, K., Yang, X., Zhang, J., An Adaptive Spherical Collision Detection and Resolution Method for Deformable Object Simulation[C], Proceedings of the 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics). IEEE, 2015: 8-17. **Cited by**: 1

Journal

Jiang, T., **Qian, K.**, Liu, S., Wang, J., Yang, X., Zhang, J., Consistent as-similar-as-possible non-isometric surface registration[J]. The Visual Computer, 2017: 1-11.

Qian, K., Bai, J., Yang, X., Pan, J., Zhang, J., Essential techniques for laparoscopic surgery simulation[J]. Computer Animation and Virtual Worlds, 2017, 28(2).

Qian, K., Jiang, T., Wang, M., Yang, X., Zhang, J. Energized soft tissue dissection in surgery simulation, Computer Animation and Virtual Worlds, 2016, 27(3-4): 280-289.
Cited by: 1

Wang, M., Sun, Y., Zhang, H., **Qian, K.**. Digital relief generation from 3D models[J]. Chinese Journal of Mechanical Engineering, 2016, 29(6): 1128-1133.

Awards

Winner (1st) of Animation and Games Development Specialist Group's 2017 Student and Open Competitions (British Computer Society)

Winner (1st) of Animation and Games Development Specialist Group's 2016 Student and Open Competitions (British Computer Society)

Acknowledgement

I would like to express my special appreciation and thanks to my advisor Professor Xiaosong Yang and Professor Jianjun Zhang, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless.

I am grateful to my familiy and my girl friend, who have provided me through moral and emotional support in my life. I am also grateful to my other family members and friends who have supported me along the way.

A very special gratitude goes out to Bournemouth University for helping and providing the funding for the work.

Thanks for all your encouragement!

Declaration

I, Kun Qian, declare that this report has been created by myself. It is being submitted for the degree of Doctor of Philosophy (PhD) at the Bournemouth University and has not been submitted in any previous application for any degree.

Chapter 1

Introduction

1.1 Background

Laparoscopy surgery is a popular minimally invasive operation. It allows the surgeons to access the inside of the human body without having to make large incisions on the skin. However, due to the limitation of small manipulation space and viewing angle, there is a higher risk of damaging the internal organs, nerves and major arteries. The qualified surgeons should have good eye-hand coordination and three-dimensional space perception skills. The traditional training approach of performing surgery under the supervision of experienced surgeons can not keep up with the ever increasing demand from surgical profession and public. On the other hand, methods relying on animals and cadavers usually cause ethical and moral issues.

To train surgeons in a safe, controlled and standardised environment without jeopardising patient's safety, virtual reality based simulation emerges as a very effective complementary learning tool. As a new way to enhance tactile and visual sensory experiences, it enables the trainees to obtain valuable experiences in an immersive virtual environment. The reason that virtual reality has not been widely used in surgery simulation in early ages is due to the hardware computing power limitation. Recently, virtual reality technology has finally entered the world of consumer electronics and shown great potential of utilities in the laparoscopic surgery simulation field.

In early age, the basic training modules provided by laparoscopic surgical simulators suffered from low fidelity sensory experience due to the limitation of hardware computational power. Although current surgery simulators can perform more complex simulations such as the anatomical variations and various pathological conditions, the hardware computing power is still the bottleneck limiting the further potential improvement towards high fidelity. Within the limited computing power, finding the balance between accuracy and efficiency is the main challenge for laparoscopic surgery simulation.

For a high fidelity surgery simulator, it should provide immersive experiences from both *visual* and *tactile* perspectives. The immersive *visual* experience regards to the realism of virtual objects generated by computer graphics technologies. The immersive *tactile* experience refers to the high fidelity haptic feedback from the virtual environment [KFN06]. There are many laparoscopic surgery simulators on the market which have been widely deployed around the world. Although those commercial simulators have undergone substantial developments in both graphics performance and haptics fidelity, most of them still only provide key procedure based training under guided routine due to the limited computing power. From the *visual* aspect, they generally suffered from the drawbacks of restricted operation degree of freedom, inaccurate soft tissue physical behaviour, limited type of medical instrument manipulation, unrealistic soft tissue presentation. From the *tactile* aspect, current surgery simulators still can not provide physically accurate haptic feedback like the real surgery due to the inaccuracies originate from the visual aspect and hardware limitations [KFN06]. Due to those drawbacks, current laparoscopic surgery simulator can only be used for education and training purpose. There is still a long way to make a surgery simulator that can provide the same level of realism as what surgeons see and feel during real operation.

Compare to the tactile fidelity, the visual realism of a simulator plays a more significant role in influencing the user's experience because visual representation is the most intuitive way to influence the user's sensory experience and all the tactile interactions are actually dependent on the visual representation of the virtual world. With the fast development of computer graphics community, state-of-art technologies which aim at improving the visual realism of graphics have been widely used in game and movie industries. Virtual surgery can be considered as a graphic based application for medical use. However, different from traditional graphics applications such as game or movie, the content of medical application mainly focuses on accurate anatomy representation and physics simulation which has not received enough attention in the main trend of computer graphics research. So designing a set of tailored techniques for surgery simulation use based on state-of-art research is an essential issue to improve the visual realism of laparoscopic surgery simulation.

1.2 Motivation

The motivations of this research are:

- Less training time for trainees and increasing resistance to use of cadavers and animals for training (totally banned in the UK).
- Wide realism gap between surgery simulator and real surgery.

- No standard and efficient production pipeline for the development of laparoscopic surgery simulation, like game or visual effect industry.

- Fast developing computer graphics and visualization technologies which have potential usage for medical applications.

Although the gap between simulation and real surgery still exists, this gap can be narrowed with the help of the state-of-art computer graphics researches. Due to the visual realism can directly influence the tactile fidelity, improving the visual realism becomes the top priority in narrowing the gap between real and virtual surgery.

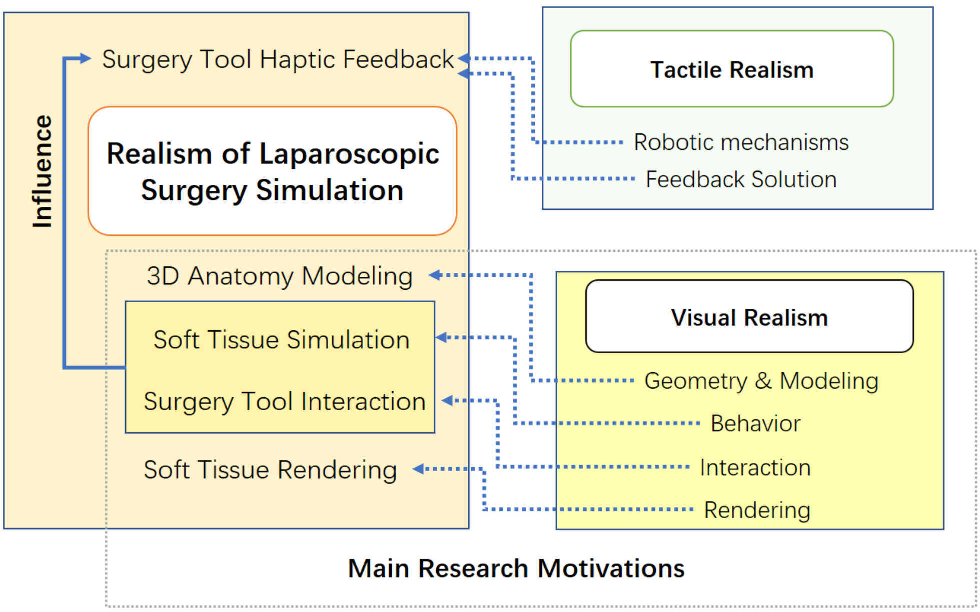

Figure 1.1: The factors that influence the realism of laparoscopic surgery simulation.

From the computer graphics perspective, the visual realism can be influenced by geometry modeling, physical behavior, interaction and rendering. As can be seen in Figure 1.1, in laparoscopic surgery simulation, those realism factors are corresponding to the realistic representation of complex anatomy structure, physically based soft tissue simulation, high fidelity tactile surgical tool interaction and realistic soft tissue rendering. As many of these techniques originated from theoretical computer graphics, computational physics, mechanical engineering, it is challenging for a practitioner with modest theoretical exposure or familiarity with these fields to navigate the most established reference textbooks in those areas, especially if their goal is to acquire a high-level understanding of the basic tools needed for implementing a simulation system. As an interdiscipline of computer graphics, computational physics, software and mechanical engineering, the development of surgery simulator always faces the dilemma of long research cycle and daunting budget, which finally results in the high cost and slow development process. A set of tailored techniques and an efficient development pipeline is in urgent need for improving the visual realism and development efficiency of laparoscopic surgery simulation.

1.3 Research Questions

By considering the aforementioned facts, this research introduces a set of tailored computer graphics techniques for improving the visual realism of laparoscopic surgery simulation. This research tries to answer the following questions:

- **Q1** How to design an efficient and realistic modeling method for complex anatomy structure ?
- **Q2** How to improve the realism of soft tissue simulation and make it more easily editable to achieve user desired effect ?
- **Q3** How to improve the stability, efficiency and accuracy of interaction between surgery tools and soft tissues ?

- **Q4** How to achieve realistic rendering of soft tissue ?

1.4 Aims and Objectives

This research is aiming at improving the visual realism of laparoscopic surgery simulation from the perspective of computer graphics. A set of tailor techniques have been proposed from the perspective of efficient anatomy modeling, accurate physically based soft tissue simulation, robust surgical tool interaction and realistic presentation of soft tissue. To achieve this aim, the specific objectives are as followed:

- **OBJ1** Review current laparoscopic surgery simulators and surgery simulation frameworks. Compare and analyse the pros and cons of existing systems and summarize the factors that influence the realism of laparoscopic surgery simulation.

- **OBJ2** Investigate state-of-art computer graphics researches which are suitable for improving the realism and efficiency of laparoscopic surgery simulator.

- **OBJ3** Design an efficient geometric pipeline which can take advantage of the already existing 3D anatomic models on the market and converting it to simulation ready model.

- **OBJ4** Propose a physically based soft tissue modeling method which can reflect the muti-layer nature of soft tissue and provide intuitive physics based user control for soft tissue property.

- **OBJ5** Develop a robust surgical tool interaction system which can provide stable sliding contact, energized dissection and clipping operations.

- **OBJ6** Design an efficient pipeline for realistic multi-layer soft tissue rendering which includes material asset generation, rendering and post-processing.

- **OBJ7** Design a laparoscopic surgery simulator to justify the feasibility of this research in improving realism of laparoscopic surgery simulation.

1.5 Contributions

This research dedicates to improving the visual realism of laparoscopic surgery simulation and the efficiency of development. It focuses on improving the current techniques from the perspective of modeling, simulation, rendering and pipeline development.

For the modeling, the 3D anatomy model used in surgery simulator should meet the standard and requirement of physics simulation (called simulation ready model). The traditional simulator development heavily depends on manual based digital anatomy asset creation. The accuracy and quality of the model is dependent on the 3D artist's skill, which may influence the accuracy and stability of physics simulation. To create anatomy models that can well adapt to the needs of physics simulation, inefficient manual or half-automated model healing and processing procedure are required before sending the models into physics simulation pipeline.

To solve this problem, this research proposed a **simulation ready anatomic model generation** pipeline which can automatically convert poor quality 3D anatomical models into simulation ready state while preserving the original model's surface parameterization attribute. The pipeline includes two stages:

- **C1** The voxelization and remesh based stage which can keep the shape of original 3D surface model but eliminate the ill shaped and degenerate polygons without influencing existing artistic pipeline.

- **C2** The cutting based surface mesh parameterization transfer stage which can transfer the original surface parameterization (UV mapping) to the simulation ready model without distortion in the parameterization space.

For the simulation, the traditional simulator mainly depends on single layer soft tissue simulation for the consideration of efficiency and pipeline compatibility. However, it does not agree with the fact that most soft tissues are multi-layer. Considering the fact that game based production pipeline is widely used in the production of surgery simulator, although there are some multi-layer soft tissue modeling strategies, their special mesh structures have not been widely used in the production of surgery simulator due to the issue of pipeline compatibility and simulation efficiency. For most of current surgery simulators, they can only provide soft tissue simulation based on certain type of material. However, the fact is that the physical property of soft tissue is complex and heterogeneous which can not be easily described using certain type of material. Current simulators lack a more general and physically accurate material property control mechanism which can uniformly represent various materials and be intuitively controlled.

In this research, **a multi-layer based heterogeneous soft tissue simulation pipeline** is proposed. This method does not need to change the traditional graphics pipeline and has no special requirement on the input mesh format. Different from other simulators which control soft tissue behaviour using certain types of material constitutive, the physical behaviour of the soft tissue in this research is controlled by a more general and polynomial based material model which can simulate various widely used hyperelastic materials using one uniform representation. Based on this material model, a heterogeneous soft tissue simulation strategy has been proposed to simulate complex physical behaviour of soft tissue. The contributions can be summarized as:

- **C3** A multi-layer soft tissue modeling method which generalizes the soft tissue into fascia, fatty and embedded tissues and provides geometric based connection strategy for different types of tissues.

- **C4** A Valanis-Landel hypothesis based soft tissue simulation framework which can simulate various hyperelastic materials via a uniform polynomial based equation and can be controlled by intuitive curve editing system.

- **C5** A heterogeneous soft tissue generation strategy which can produce heterogenerous stiffness distribution on soft tissue via a skeleton based painting system.

For the surgical tool interaction, the traditional simulators often use guided routine to limit the flexibility of surgical tools to avoid frequent contact in case of producing simulation artifact. They lack an effective and robust collision detection and resolution system to support the frequent contact especially the sliding contact between surgical tool and soft tissue (which is a common operation in laparoscopic surgery). So the contact types and details are also limited especially the energy based dissection and clip operations, which have not been well research in previous works.

In this research, **an efficient and robust surgical tool interaction system** based on adaptive circumsphere mesh representation is proposed, which supports robust and efficient sliding contact. Based on the interaction system, an energized soft tissue dissection and a geometric based clipping method have been proposed. The contributions are summarized as:

- **C6** A circumsphere based collision detection and resolution method which outperforms both the existing sphere-based and polygon-based methods in overall performance and reduces collision tunnelling effectively.

- **C7** A local geometry feature and energy based adaptivity strategy which dynamically adapt the location and size of the circumspheres surface for better approximation and provide stable and robust collision response.

- **C8** A computationally efficient dissection model based on heat transfer which provides physically based modeling for the dissected area. It will not change the topology of the virtual objects and can keep the efficiency and stability of the simulation.

- **C9** An efficient geometric based soft tissue clip method which can keep the clip attaching to the vessel and produce visual plausible clip result without introducing the rigid body simulation.

For the rendering of the soft tissue, the existing simulators can only provide single layer soft tissue rendering which neglects the structure of the multi-layer soft tissue. The rendering techniques used in existing simulators are mainly from game industries. The direct use of those techniques can not well meet the request level of realism in surgery simulation, which often result in low fidelity rendering artifact in existing simulators.

In this research, **a realistic multi-layer soft tissue rendering pipeline** is proposed. It decomposed the multi-layer structure of soft tissue into corresponding material asset required by state-of-art rendering techniques. This research proposed tailored rendering model and post processing pipeline to further improve the realism of laparoscopic surgery simulation. The contributions are:

- **C10** A procedural organic material generation pipeline which enables the users to efficiently generate organic look texture based on real images.

- **C11** A multi-layer soft tissue rendering pipeline which includes multi-layer soft tissue shading, optimized lighting effect modeling and post-processing.

For the system development, currently there are not any standard develop framework or pipeline existing in the industry of surgery simulation because the structure of system framework highly depends on the techniques used in the system.

In this research, **a laparoscopic surgery development framework** based on the techniques in this thesis is proposed. It shows the mechanism of how to organize all the techniques into one framework. The contribution can be summarized as:

- **C12** A system framework for building a laparoscopic surgery simulator which includes the hardware setup, GPU based simulation pipeline, deferred shading based rendering pipeline and performance analysis.

1.6 Scope and Limitations

This research focus on the tailored techniques which target at improving the efficiency, realism, stability and accuracy of current surgery simulation from the perspective of computer graphics. This research is based on the existing features and functions in current simulators. The system validation is not the focus of this research but intuitive interfaces are provided for potential use in the clinic validation. Detailed system validation can be performed according to individual manufacture's rule or under the guidance of assessment methodology of manual dexterity (such as Direct Observation of Procedural Skills (DOPS) [BA07], Objective Structured Assessment of Technical Skills (OSATS) [ABH+07]).

1.7 Thesis Outline

This thesis is organised into eight chapters. The thesis milestones include the relation between each chapter with the research questions, objectives and contributions as shown in figure 1.2:

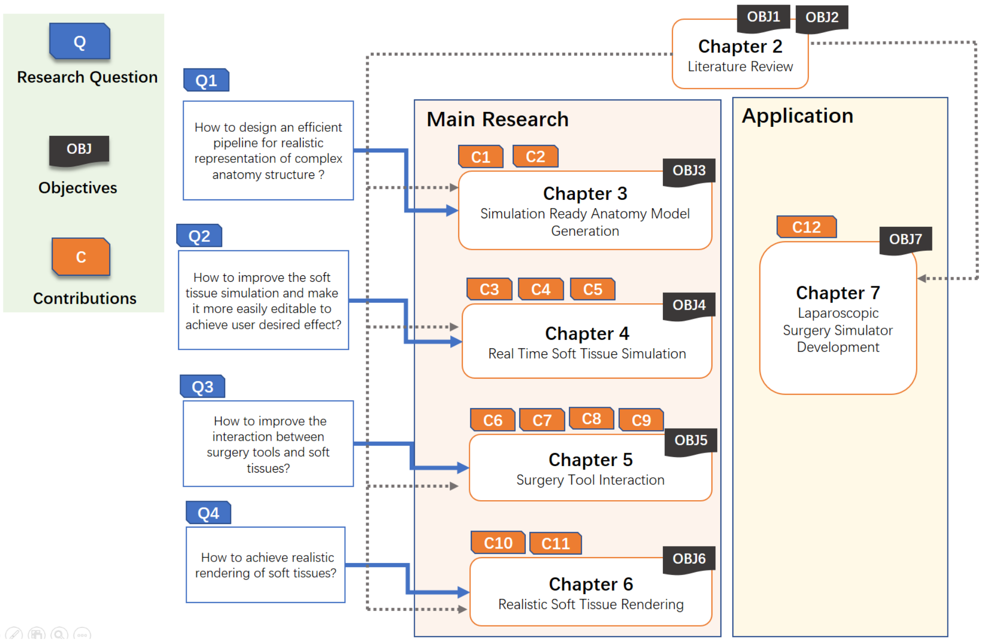

Figure 1.2: The structure of the thesis.

- In chapter 2, a literature review on the laparoscopic surgery simulation and its related computer graphics research have been made. It gives an overview of current laparoscopic surgery simulators and in depth technique analysis of factors that influencing the realism of laparoscopic surgery simulation from the perspective of computer graphics research (**OBJ1**). After that, a detailed review on the related computer graphics researches have been made which include the topics of modeling, deformation simulation, collision detection and rendering (**OBJ2**).

- In chapter 3, an automatic simulation ready model generation pipeline is introduced which can convert the poor quality anatomic model into simulation ready model while preserving the original model's surface parameterization property during the transfer procedure. (**OBJ3**).

- In chapter 4, a soft tissue simulation pipeline has been proposed which can provide multi-layer heterogeneous soft tissue modeling and intuitive physically editable simulation based on uniform polynomial based hyperelastic material representation.(**OBJ4**).

- In chapter 5, an adaptive circumsphere based surgical tool interaction system has been developed which support robust collision (especially the sliding contact) and other types of frequently performed operations such as energized dissection and clipping. (**OBJ5**).

- In chapter 6, a pipeline for realistic multi-layer soft tissue rendering has been proposed. It includes procedural based material assets generation, physically based multi-layer soft tissue rendering and post processing. (**OBJ6**)

- In chapter 7, a framework for the development of laparoscopic surgery simulator has been proposed, which includes the techniques proposed in this research (**OBJ7**). Detailed performance analysis of the system in each component has been made in this chapter.

Chapter 2

Literature Review

2.1 Overview of Laparoscopic Surgery Simulator

Laparoscopy has become the standard approach in most surgical specialties [LSG$^+$09]. It is evident that laparoscopy is associated with a longer operation time and a higher rate of surgical complications during the learning curve of the surgeons. The primary obstacles in learning laparoscopy are psychomotor determinants and perceptual speed. To train those skills, simulation emerges as a complementary tool to traditional training, allowing the reduction of learning curves in a safe and controlled environment [CIRM06].

"A skillfully performed operation is about 75% decision making and 25% dexterity[Spe78]." Judgement, knowledge and dexterity are three critical elements of safe surgery [DDM01]. The main purpose of surgery simulator is to enhance the knowledge of procedure, practice decision making, improve technical skills by performing critical steps in real surgery. To achieve these goals, current simulators on the market can be generally classified into two groups: *cognitive based training* and *dexterity based training.*

The simulators for cognitive based training are used for the education and demonstration purpose, which do not require realtime simulation and high accuracy. They show the general overview and key steps of certain surgical procedure. Those simulators focus on the modelling of complex disease and demonstrating their surgical solution steps. There are many companies developing the cognitive based medical education applications such as Touch surgery [1], ,Visible Body[2], LibroScience [3] etc. Touch Surgery used cognitive mapping techniques, cutting edge AI and 3D rendering technology to codify surgical procedures. Visible Body mainly develops applications for medically accurate content such as biomedical visualization. LibroScience released an interactive CT and MRI visualization tools for cell phone.

The simulators for dexterity training focus on transferring skills from the simulated environment to the operating theater. They have high demand on the simulation efficiency and accuracy. Those simulators mostly have extended simulation platform with many essential features specifically designed for medical simulation and haptic feedback. There are many companies developing such training systems such as 3D system (formerly Simbionix)[4] , Surgical Science [5], Sense Graphics [6], CAE healthcare's LapVR[7], MEDICAL-X's LAP-X[8]. The products of those companies not only cover wide range simulation scenarios for dexterity training but also cognitive training modules.

From the computer animation perspectives, compared to the simu-

[1] https://www.touchsurgery.com/
[2] https://www.visiblebody.com/
[3] https://www.libroscience.com/
[4] http://simbionix.com/
[5] https://surgicalscience.com/
[6] http://sensegraphics.com/
[7] https://caehealthcare.com/
[8] https://www.medical-x.com/

lators for dexterity training, the simulators for cognitive based training is relatively easy to develop because they heavily rely on pre-computed or key-framed animation rather than real-time simulation. The key of developing the cognitive based simulators is how to interactively demonstrate an animation based surgical procedure to users. This kind of simulators are not the focus of the research in this thesis because it is an artistic driven medical application.

The development of dexterity training based simulator is an interdiscipline of computer graphics, computational mechanics and mechanical hardware design. The goal of dexterity training based simulator is to allow the surgeons to have an experience that similar to the real surgery. To achieve this, accurate physical behaviour, photo realistic anatomical 3D presentation and high fidelity haptic feedback are very important to an immersive surgery simulator. Those key factors are corresponding to physics simulation, rendering and geometry modeling problems in computer graphics research. However, due to the realtime performance requirement of laparoscopic surgery simulator, compromises have to be made on the model resolution, simulation accuracy and render quality which will often result in the undesired visual artifact (see Figure 2.1 [9] [10] [11] [12]). Finding a balance between the visual plausibility and computational efficiency is the key when developing the surgery simulator. In the following part, related works regarding to the development of laparoscopic surgery simulator will be discussed from the perspectives of 3D anatomy modeling, soft tissue simulation, surgical tool interaction and soft tissue rendering.

[9] https://www.youtube.com/user/surgicalscience
[10] https://www.youtube.com/user/SimbionixUSA
[11] https://www.youtube.com/channel/UCxvlvWkIv9bgmbY0Iz5K6IA
[12] http://sensegraphics.com/

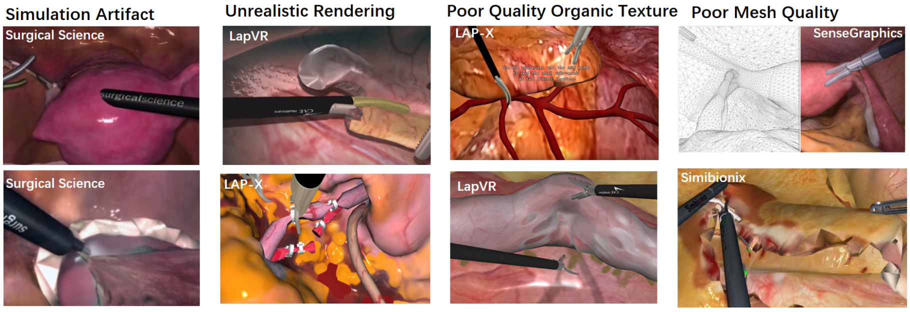

Figure 2.1: The artifacts exist in current laparoscopic surgery simulators.

2.2 Anatomy Modeling

For the anatomy modeling, the ideal pipeline is to obtain 3D anatomy model from patient specific CT or MRI data. However, the labeling of the interested anatomy area from CT or MRI data is normally manual based and inaccurate especially for laparoscopic surgery because the soft tissues such as rectum are hard to be captured. Digital modeling based on coarse CT and MRI data is still the main trend in developing laparoscopic surgery simulator but the quality of the mesh highly dependents on the 3D artist's skill. Irregular polygon distribution, self-intersected or interpenetrated primitives caused by manual based modelling process will influence the simulation accuracy and stability. There are a lot of commercial 3D human anatomical models on the market but most of them can only be used for visualization purpose. Most of those models can not be directly used for physics simulation (see chapter 3 for more details). How to efficiently obtain a nice and clean simulation ready 3D anatomy model is still challenging in the development of surgery simulator.

The process of optimizing a poor quality non-manifold 3D surface model into a simulation ready model is a geometric processing problem.

Traditional geometric processing techniques such as split, collapse, fuse operation on vertex, edge, face etc [BLP+13] can be helpful in fixing the geometric degeneracies but it often requires the human intervention to get desired result. Instead of working on the trivial polygon representation level, those non-manifold geometry optimization problems can also be solved from the perspectives of voxel representation, which can facilitate the detection of self-intersections and other degeneracies using signed distance field (SDF). It is non-trivial to get rid of the non-manifold geometry by performing the topological operations (such as union, intersection and difference etc.) based on the SDF. The volumetric data can be represented on Cartesian, unstructured, Octree grid [JLSW02]. The Cartesian grid is convenient for fast interpolation, level set schemes such as marching based method [Set96], which has been widely used in game and VFX industries[Mus13].

After the voxelization step, an isosurface mesh can be extracted based on the SDF. However, the original source mesh's surface parameterization information will be lost in the isosurface mesh. The source mesh's surface parameterization property should be completely inherited for the consideration of production efficiency. To preserve the source mesh's surface attribute, the transfer of the surface properties between the source and the target mesh is the key technique during this process.

Transferring surface properties from source mesh to target mesh is an important research topic in computer graphics, which includes detail synthesis [WHRO10], shape analysis[GF08], texture synthesis[LH06] and surface editing [SCOL+04] etc. In the case of converting existing poor quality 3D anatomy models into good quality models, the input mesh is a non-manifold geometries with self-intersections and other degeneracies, the output mesh is a high quality simulation ready polygon model. To match the original mesh to the newly generated mesh, finding correspondences between two surfaces is needed before transferring surface properties. The task of finding shape correspondence can range from the shape identification of portions which geometrically similar[JQL+17], to the pairing of elements that represent the same parts or serve the same function on the shape [XKH+16]. Van et al [VKZHCO11] gives a recent survey on the details of finding shape correspondence. The classical shape correspondence applications involves two steps: shape alignment [DBL01] and feature matching[YMYK14]. In the voxelization based model optimization pipeline, the target mesh is generated from the voxelization and remesh result of the source mesh. The target mesh has already preserved the shape of the original mesh but with different polygon representations so that shape alignment step is not necessary. Only the features of source and target mesh are needed to be matched and then the texture color and surface parametrization can be transferred.

For the texture color transfer problem, there are mainly two general approaches. The first one builds a common parameterization which is used as a mapping from the source surface to the target surface [Ale02]. The smoothness of the mapping will affect the scale of local distortion. This method will preserve large scale pattern in textures but suffer from local scale distortion especially when the shape variation is large between source and target mesh. Texture synthesis, as the second approach, is a process of constructing large images from exemplar by preserving its structural content and detail feature. The traditional methods in texture synthesis is a process of pixel-based neighborhood-matching which finds the best matching pixel between partially synthesized neighborhoods and exemplar neighbours [LH06]. This method is able to reproduce the small scale details of the texture but can not well preserve the large scale patterns. The above methods will inevitably produce distortion.

Beside the distortion problem, texture discontinuity across the UV seam (chart boundary) is another issue when transferring texture color. UV seam of the texture atlases will produce discontinuities artifact because the neighbouring points on the mesh surface may be far away from each other in the parameterization space (UV space). There are many works on solving the problem of UV seam discontinuity. Nicolas et al. [RNLL10] proposed a solution to make the UV seam invisible by aligning texel grids across UV seam based on grid-preserving parameterization. Such alignment ensures the interpolated color on both sides of UV seam precisely matched. Lefebvre et al. [LH06] synthesized best matching pixel on both sides of UV seam, making it hard to percept. A two-phase continuity mapping has been proposed et al. [GP09] by Francisco. The first phase a bidirectional mapping is built between areas outside the UV seam and the corresponding areas inside, the second one generates a thin border of virtual triangles in texture space to correctly interpolate and filter texture values through the seams. Sheffer et al. [SP06] proposed a mesh cluster method. It creates rectangular patches which preserves a one-to-one texel correspondence across UV seam boundaries. However, those methods focus on healing the UV discontinuity problem based on the original mesh's polygon representation but they are not suitable for this research because this research focus on transfer surface parameterization property between two surfaces of different polygon representations. Instead of solving the UV discontinuity on original mesh, the source mesh's UV parameterization is needed to be preserved for the target mesh and ensure the original UV parameterization causes no artefact (UV space element stretch around UV seam) on the target mesh.

Compare to the existing methods, most of current texture transfer techniques can only maintain the texture color and inevitably cause distortion to some extent. In this research, the target mesh is needed to not only inherit the source mesh's texture color but also the parameterization because the parameterization information may be needed for other purposes in later stage of production pipeline, such as UV space bleeding, rain or sweating effect. However, the inheritance of the original parameterization will cause artifact due to the existence of parameterization seam. In this paper, the texture transfer problem is solved using a cutting and remesh based technique which retains the original mesh's surface property and eliminates the distortion problem during the transfer process.

2.3 Deformation Simulation

For simulation, there are mainly two choices: using existing engines or developing from scratch. Most of the realtime physics simulation engines are developed for game and visual effect industries, such as Havok, PhysX, Bullet etc. Those engines have been used in the development of some surgery simulators [MHL+09] [MWW07] but they do not well support soft body dynamics such as FEM based soft body simulation so they are not quite suitable for high fidelity laparoscopic surgery simulation. Nvidia recently released a particle based simulation engine (FLEX[13]) but it only supports robust geometric based soft body simulation[MMCK14] rather than FEM based. Different from game engines, SOFA [14]is an open-source framework primarily targeted at realtime simulation with an emphasis on medical simulation developed by INRIA. However, SOFA is a general physics engine which has not been

[13]FLEX, https://developer.nvidia.com/flex

[14]Software Open Framework Architecture, https://www.sofa-framework.org/.

tailored for any specific surgery simulation such as laparoscopic surgery. SOFA does not support the features that laparoscopic surgery needed such as multi-layer heterogeneous soft tissue simulation, energized dissection, embedded soft tissue simulation, clipping operation etc. Also, SOFA does not have an effective way for soft tissue property editing which is essential for improving user experience.

Different from common elastic materials such as rubber, synthetic fabric etc., soft tissues are mostly the composite of matrix material which is embedded with a single family of aligned fibers [GPM06]. Due to the complex structure of soft tissue, its physical behaviour is hard to characterize. While, for some well organized tissue, it can be described well using the classical continuum mechanics theory.

From the perspective of computer graphics research, deformation simulation is the core of developing a laparoscopic surgery simulator which can influence a simulator's fidelity and realism. The design of the physical property and choice of solver will influence the accuracy and stability of simulation. Realtime simulation of soft tissue is a process of solving dynamic system. To achieve realtime performance, compromise on accuracy has to be made which will influence the simulator's fidelity and user experience. Within the computation budget for realtime performance, balancing the accuracy and user experience is important which is a challenge for simulator tuning and evaluation.

The simulation of solid and deformable objects has been an active research topic in computer graphic research area for more than 30 years. There are huge amount of excellent works continue to emerge. Among these excellent works, a few good surveys [MtV05] [NMK+06] [MSJT08] [BET14] [BMM17] papers can help researchers get a comprehensive overview of related area quickly.

As the pioneer of deformation simulation, Terzopoulos et al. [TPBF87] proposed an elastic deformation model which used the penalty force derived from energy functions to simulate elastic object based on finite difference approach. Their work lead the early development of deformation simulation in physics based animation[BW98][EWS96][BHW94] but the realtime simulation was not mature at that time until the early 2000. The realtime physics simulation began to attract people's attention when Havok was used for game industries. The realtime deformation simulation method can be classified into two groups: geometric based and physics based. The geometric based methods apply geometric constraint to maintain the original shape (such as the shape matching methods [MHTG05a][SOG08], ARAP [BG07], Bounded Biharmonic Weights [JBPS11] etc). For laparoscopic surgery simulation, geometric based simulation can provide realtime performance for soft tissue simulation [QBY+15] but lacks of physical support which limits the range of physical behaviours. Different from geometric based method, the physics based methods are dependent on the physics laws. Finite element method (FEM) is the most widely used method in computational science which discretizes deformable objects into finite elements (such as edge, quad, triangle, tetrahedron etc.) and approximates the analytical solution of partial differential equations at discrete number of points over the domain. Mass spring system [GHF+07] [LBOK13] [SLF08] is a popular representation for deformable objects (especially thin shell such as cloth) which can be regards as a simplified finite element method. However, mass spring system cannot well capture the volumetric effect. Finite element method is physical accurate simulation method [ITF04][BWHT07] which is based on the continuum mechanics laws. However, the expensive computational cost makes finite element methods rarely used in realtime applications in the early 2000. However, with the development of software and hardware technologies, realtime

FEM based simulation of hyperelastic material is achievable but not physically accurate [SB12a][BKCW14][BML+14][WY16].

The physically properties of soft tissues are much more complex than hyperelastic materials. There are many good surveys on the biomechanics soft tissue simulation in both biomechanic engineering areas [Del98][Li16] and computer graphics research[MLM+05][LGK+12]. In biomechanic research, approximate the soft tissue behaviour using existing ideal models is a common approach. Mass spring theory and hyperelastic models are most widely used models for analysing the soft tissue physical behaviour. For example, Baier-Saip et al. [BSBSO] approximate the elasticity of artery by connecting discretized model nodes using three types of linear springs. Gasser et al. [GOH06] provides a more accurate hyperelastic model for the modeling of arterial layer. Li et al. [Li16] analysis and compares the damage models of soft tissue which highly dependent on hyperelastic models. Natali et al. [NCP+06] describes a procedure used to define constitutive parameters for hyperelastic soft tissue constitutive models.

In computer graphics research, mass spring theory and hyperelastic materials are also widely used models for soft tissue simulation. Mass spring based soft tissue simulation by organizing the structures of different types of springs (tetrahedral spring [MDM+02], anisotropic spring [BC00], volume preserve spring [AR08] etc.) is a widely used. Hyperelastic model based soft tissue simulation [BWHT][XLCB15][LXB16] is also an active research topics in computer graphics community especially in virtual surgery field [CAR+09][QBY+15][TW14]. Realtime performance is an important research topic in deformation simulation.

For laparscopic surgery simulation, realtime performance is the corner stone for the simulation system. Müller et al. [MHTG05b] proposed meshless shape matching which is a fast geometric based deformation simulation method. This method is the prototype of the popular position based dynamics. Position based dynamics (PBD) [MHHR07a] was developed rapidly in the past decade. It has been widely used in games [MMCK14] and visual effect industries [ATO15][MCK13]. PBD can be considered as an approximate implicit Euler integration scheme. The idea of PBD is its direct control of position via constraint projection, eliminating the overshoot problem in the tradition force based method and achieving unconditional stability. There are many works on improving PBD to a more robust and versatile method, such as the improvement on convergence [Mül08][KCM12][Wan15], continuum mechanics [BKCW14][MCKM14], parallel and unified solver [FP15] [MMCK14], fluid [MM13], rigid body[MCMJ17][DCB14][MCK13], robust elastic simulation [CMM16][MC11][MC10][USS14].

However, the simulation result of PBD is dependent on the time step, iteration count and constraint traversal order. To solve this problem, a new interpretation of PBD as an approximate solver for implicit Euler has been proposed in [LBOK13], which presents a local/global solve based mass spring system. This method has been further generalized into the theory of Projective Dynamics [BML$^+$14]. To overcome the disadvantage that PBD is not totally derived from continuum mechanical theory, Projective Dynamics builds a bridge between the nodal finite element method and PBD by formulating a convex quadratic energy, which can be solved using the local/global strategy. Wang et al.[Wan15] proposed a chebyshev semi-iterative approach to accelerate the convergence rate of Projective Dynamics and PBD. ADMM as a more general form of Projective Dynamics has been proposed in [NOB16]. Those methods are based on the global solve of a prefactorized linear system. When the mesh topology changes each frame, the efficiency of the global solve methods will be affected significantly.

According to the above analysis, for realtime laparoscopic surgery simulation, physically accurate continuum mechanics based soft tissue simulation is still challenging. PBD is not rigorously derived from implicit integration which means it can not accurate describe the concept of force. It only provides a visual plausible simulation. The Projective Dynamics formulate the energy as the quadratic form which can not be used to simulate the most basic continuum mechanics materials such as St. Venant Kirchhoff, Neo-Hookean etc. Liu et al. [LBK17] interpreted the projective dynamics as a quasi-Newton method which can simulate the mechanics based materials faster than accurate Newton method but its realtime performance still not that good and also surfer from the topology change inefficiency problem. Thus the realtime physically accurate simulation of soft tissue is not achievable yet. So combining the physical plausible simulation with effective user-control is a way of remedy for inaccuracy, which has been used in the development of most surgery simulators. In this thesis, a continuum mechanics based soft tissue simulation method derived from implicit integration is proposed, which can provide physically plausible simulation and physically based intuitive user control.

2.4 Surgical Tool Interaction

In laparoscopic surgery simulation, there are multiple types of tool interactions such as sliding contact[HPPMI10], poke, clip and dissection etc[BM94]. To reduce the risk to the patient, the contact between soft tissue and sharp instruments should be minimized purposefully . Tools used to clip, poke, push and stabilize soft tissue are normally blunt [Mis13]. Instead of using a scalpel, mechanical energy-based dissection systems [AFA+16] have been incorporated into modern laparoscopic surgery, such as electricity, diathermy, and ultrasound. Before the dissection is performed, important blood vessels should be clipped in case of losing too much blood. In a laparoscopic surgery simulator, those operations rely on a robust collision detection system.

2.4.1 Surgical Tool Collision Detection and Resolution

Due to the resolution of 3D anatomy model is usually high for good rendering quality, it poses great challenges to the performance of interaction accuracy, stability and efficiency. A robust collision detection and response system is of great importance to laparoscopic surgery simulation. Most of the early work concentrated on the interaction between rigid bodies where most computation can be completed in the preprocessing stage. The collision handling of deformable objects is much more complex, as the collision responses should be dynamically fed to the deformation computation. This brings greater challenges to the efficiency, stability and accuracy of the collision detection and resolution algorithm. The realtime performance of collision detection and resolution with soft body largely relies on the efficiency of localising the potentially colliding geometry and calculating the polygon intersections.

In current surgery simulators, frequent tool soft tissue sliding contact is avoided because it may cause collision tunneling artifact when not using accurate but computationally expensive collision detection methods (see figure 2.2[15] [16]). Improving the efficiency of collision detection while not affecting the accuracy and stability is the goal of a robust collision detection system for laparoscopic surgery simulator. Generally, the collision detection process can be accelerated from the perspective of the broad phase and narrow phase.

[15]https://www.youtube.com/user/surgicalscience/featured

[16]https://www.youtube.com/channel/UCxvlvWkIv9bgmbY0Iz5K6IA

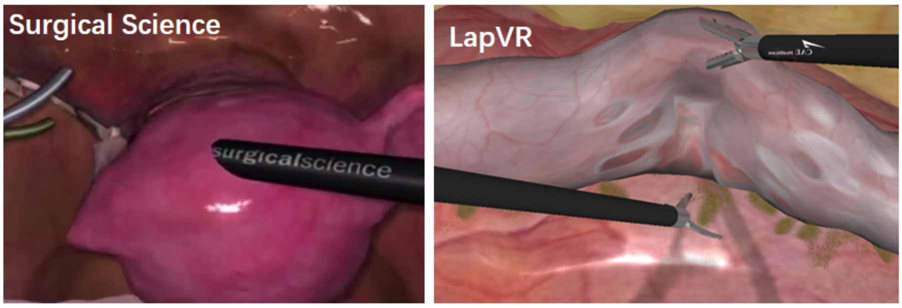

Figure 2.2: Simulation artifact in current surgery simulators.

For the broad phase optimization, highly efficient spatial data structures, such as bounding volume hierarchies (BVH), spatial hash, distance fields and image based method are proposed. A comprehensive summary of these techniques has been made in [TKH+05]. Among those methods, BVH and spatial hash are two most used spatial data structures. For constructing BVH, many types of bounding primitive can be used such as OBB [GLM96], AABB [BFA02] [BWK03], k-DOP [KHM+98] [MKE03], Sphere [Hub96] [JP04]. For most BVH based collision detection method, polygon intersection test in the leaf node is inevitable. Although fast BVH updating methods are proposed in [LAM01] [MKE03] [JP04], updating of BVH in each frame is still time consuming and not easy for parallel computation. For spatial hash method [THM+03], uniform grid hash is used for identifying potential pairs and easy for parallel computation [EL07] [PKS10]. Those structures have ready been well researched. The method proposed in this thesis focuses on the finest level collision detection and resolution, which is compatible with the hierarchical spatial data structures and their acceleration strategies.

Compare to rigid object, the cost of soft body collision detection is more expensive due to the update of hierarchy spatial data structure and the test of polygon intersection test in each frame. Although Ericson [Eri04] has efficiently optimized basic primitive(vertex, line, face) intersection test based on Voronoi regions, polygon intersection based method remains computationally expensive. Using simplified mesh representation is an alternative to handle collision detection and resolution with a reduced cost. Sphere based structure is the most widely used simplified representation for collision detection.

As to simplified mesh representation for rigid body collision detection, Hubbard [Hub96] proposed a time critical collision detection method using tightly fitting sphere hierarchies built from medial-axis surface. Medial axis has been used frequently in the construction of sphere tree or sphere packing for rigid body [DO00] [Hub96] [BO02] [WZ09] [TMC10]. However, as the computation of the above methods is expensive and mostly occurs in the initial configuration, they are not suitable for dynamic updating of deformable objects.

As to simplified representation for deformable body collision detection, Mendoza[MO06] proposed a time-critical collision detection algorithm for deformable objects based on a sphere tree constructed using an adaptive medial-axis approximation of original mesh. However, it updated the bounding sphere only using coarse tetrahedral mesh which is used for FEM simulation. This method can not handle detailed collision because bounding sphere only exists in the coarse mesh level. To improve its accuracy, Mendoza[MO06] also presented an idea of placing sphere on all triangle surfaces ensuring full coverage. However, this sphere representation does not approximate the original surface well so that it will incur obvious visual artefacts when handling detailed collision detection and resolution. BD-tree(Bounded Deformation Tree) [JP04] proposed an output sensitive collision detection for reduced deformable models. Although BD-Tree provides effective output-sensitive collision detection that can be much faster than hierarchical updating, it will suffer from over conservative bound, computational and memory overhead.

For collision resolution, it depends on the dynamic simulation system used. For soft body simulation, constraint projection based methods are very popular in recent years [MHHR07b][BML+14]. When performing the constraint projection, normally the constraint sets (such as length, volume, collision constraints etc.) are not feasible with each other. Solving one type of constraint is possible to violate the previously solved constraints. This will cause the oscillation artifact on the collided surface especially when sliding contact occurs. While, sliding contact is quite common in laparoscopic surgery simulation. In this thesis, a circumsphere based collision detection and resolution method is proposed which dynamically adapt the location and size of the implicit circumspheres surface. The method proposed in this thesis can provide good approximation to the original surface and stable collision response especially for sliding operation.

2.4.2 Surgical Tool Dissection

Dissection and clipping are also frequently performed operations in laparoscopic surgery. To reduce the risk to the patient, the contact between soft tissue and sharp instruments should be minimized. Unlike scalpel based cutting, energized dissection will not produce heat and clean incision which often modeled by geometric cutting methods in computer graphics research. The dissection of the soft tissue is caused by the energy at the tip of the tool rather the blade of a scalpel. Different from the incision cut by scalpel, the incision dissected by energized tool will subject to a certain degree of shrink due to the heat. In computer graphics research, the energized tool-based dissection has not been widely studied yet.

Wu et al. [WWD15] gives a comprehensive introduction to the recent cutting techniques used in physics based simulation. The cutting process can be described as finding elements (triangles, tetrahedrons, hexahedral etc.) that intersected by the cutting surface and then refine the intersected elements to conform to the cutting surface. How to handle the refinement process for the intersected elements is the key which will influence the performance of cutting algorithm significantly. Refine the intersected elements [BMG99] [SHGS06] [BGTG04] or duplicate them via virtual node [MBF04] [SDF07] [WJST14] according to the cutting surface can produce better incisions that conform to the cutting surface. However, those methods will introduce new elements and change mesh topology which will heavily influence simulation performance especially the global solve based methods. Directly removing the intersected elements is the easiest way to incorporate cuts into deformable object without introducing new elements [CDA00] but it will generate jagged surface on the newly exposed area, as can be seen in figure 2.3[17] [18]. However, this method has been widely adopted to realtime applications due to its simplicity. To reduce the jagged surface artifact, remedy methods such as vertex snapping [NvdS01] can be applied.

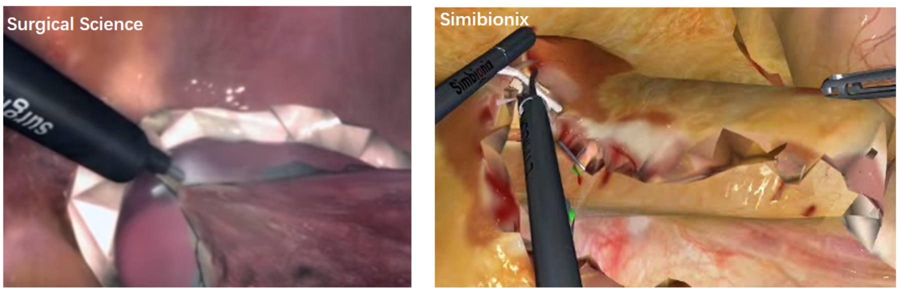

Figure 2.3: Dissection artifact in current surgery simulators.

Most of the researches focus on refinement based geometric cutting

[17] https://www.youtube.com/user/surgicalscience
[18] https://www.youtube.com/user/SimbionixUSA

which will generate neat and regular incisions on the cutting areas. Such cutting modes are not suitable for the laparoscopic surgery simulation because the energized dissection tools are not sharp and will not generate regular incisions like using scalpel etc. There is no need to follow the refinement based dissection methods. To guarantee the realtime performance and stability, some simulators used pre-computed incisions but it keeps the users from arbitrary cutting which will greatly influence the user experiences (see figure 2.4[19]). In this thesis, a computationally efficient energized tool based dissection is proposed based on heat transfer model. The method proposed in this thesis will not introduce new elements while provide physically based modeling method for the dissected area.

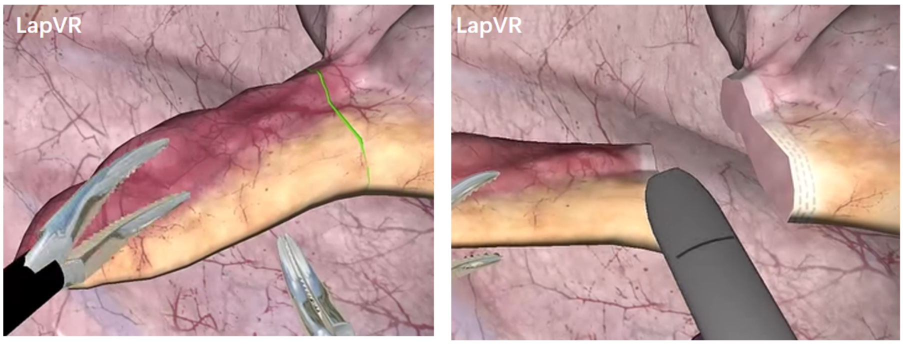

Figure 2.4: Pre-computed dissection in current simulators.

2.5 Soft Tissue Rendering

For the rendering, most of the techniques used in the development of simulators are derived from the game and visual effect industries. There are not so many specially designed rendering techniques yet for realistic soft tissue rendering. Most of the simulators render the soft tissue us-

[19]https://www.youtube.com/channel/UCxvlvWkIv9bgmbY0Iz5K6IA

ing simplified rendering pipeline from game industries which may easily cause unrealistic plastic appearance, as can be seen in figure 2.5 [20] [21]. Game engines such as Unreal, Unity etc are not suitable for the development of laparoscopic surgery simulation yet. Although they provide powerful rendering techniques and user-friendly material editing tools, the lack of physically based soft body simulation and haptic device interface make them not suitable for surgery simulator development. There is a large gap between the realism of soft tissue base appearance for different simulators, which is mainly dependent on the artistic pipeline and the artist's personal strength.

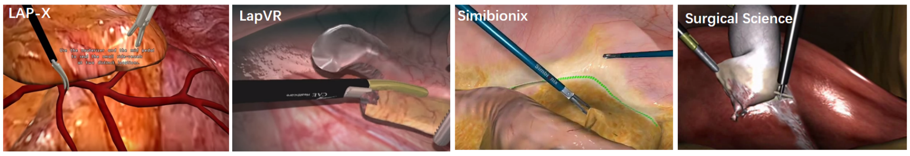

Figure 2.5: Rendering quality between different simulators varied a lot.

In laparoscopic surgery, the scene inside the patient 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. The realism of the simulator is mostly dependent on the quality of the textures, material models and lighting. In laparoscopic surgery, the position of the camera is very close to the surrounding anatomies so that it poses high requirement on the quality of the soft tissue's close rendering quality. Low quality textures, inappropriate material and lighting models will all influence the quality of the final rendering. As can be seen in figure, the quality of soft tissue texture varied a lot between different simulators and the plastic rendering effect is common in many simulators.

[20] https://www.youtube.com/user/SimbionixUSA

[21] https://www.youtube.com/user/surgicalscience

Most previous works on the rendering of surgical simulation use the traditional shading models [ELDY04] or video image based method [LJD07]. 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 is 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 [Hof10][SL14]. Physically based rendering (PBR) refers to the concept of using realistic shading models along with measured surface values to accurately represent real-world materials. PBR has been widely used in game and visual effect industries but there are not many works on soft tissue rendering in surgery simulation. A PBR based deferred shading rendering pipeline for soft tissue has been proposed in [QBY+15] but this model has not reflected too many properties of soft tissue. Nunes et al.[NMCW17] proposed BRDF model whose parameters are determined by reference laparoscopic video. Similar works have been proposed in [CDS+06][ENC+08] which approximate the BRDF model based on image for virtual bronchoscope simulation.

Due to the complex environment in laparoscopic surgery, there are many tissues of different properties in camera view. Measuring the BRDF for different tissues are difficult. In fact, the plastic rendering artifact in figure 2.5 is not totally caused by the incorrect BRDF model. Soft tissue is a multi-layer object. Using traditional single layer rendering techniques can not achieve correct result. In this thesis, a procedural and physically based multi-layer soft tissue rendering pipeline has been proposed, which can efficiently generate organic procedural texture for soft tissue and provides realistic soft tissue rendering.

Chapter 3

Simulation Ready Anatomy Model Generation

3.1 Introduction

A realistic anatomic model is of great importance to medical applications such as surgery simulation. In the current market, there are a lot of 3D human anatomic models which often includes complete set of human anatomies such as muscles, organs, skeletons and nerve system etc. Those models are mostly created by 3D artists who work under the guidance of medical practitioners or human anatomy structure books. However, when artists create those models, they focus more on the visual appearance. To achieve complex shapes and structures especially for junior artists, they may use irregular primitives to approximate the shape of the real anatomy structure. Due to the complexity of anatomic structure, self intersections, inter-penetrations are always inevitable during the modeling process. Although such artifact will not influence the rendering effect, it makes the anatomic model can not be directly used in physics simulation.

In surgery simulation, continuum mechanics based soft body simulation is widely used. It requires the structure of the 3D anatomic model to be a solid object. Tetrahedron and hexahedron are common used discrete forms of solid object. To convert a 3D surface mesh into a solid model, it requires the surface mesh to be a *simulation ready model*.

Simulation ready model is defined as a surface mesh which satisfies the model requirement for physics simulation and can be directly used for solid mesh conversion. To convert the surface mesh into solid mesh, it requires the surface mesh having the properties of watertight, no self-intersection and no degenerate element. The surface mesh that has those properties is called *simulation ready model*. The *simulation ready model* can be directly tetrahedralized or hexahedralized to generate a discretized solid representation of the object which is required in continuum mechanics based simulation.

For the development of surgery simulator, using existing 3D anatomic models on the market rather than modeling everything from scratch is a cost saving choice especially for small companies or research groups which have inadequate professional digital artists. To convert the existing anatomic models into a simulation ready model, there are mainly two challenges during this process. The first is how to generate a simulation ready model which can efficiently get rid of the ill shaped and degenerate polygons in original model without manual fixing. The second is how transfer the original mesh's surface parameterization attribute to the newly generated simulation ready model (target mesh) without any distortion. Although the first challenge can be fixed using existing geometry processing tools (Meshlab etc.) and 3D content creation tools (Blender, Houdini etc.), manual intervention is needed because the conversions require the combination of various geometry algorithms. For the second challenge, only some digital content creation tools such as

Maya, Houdini support simple attribute transfer feature. Although surface parameterization attribute can be transferred, it only works well when the topology of two meshes are similar. If the topology of two meshes varied too much and UV seams exist, obvious artifact will appear near UV seams.

Transferring surface properties from original mesh to target mesh is an important research topic in computer graphics, which includes detail synthesis [WHRO10], shape analysis[GF08], texture synthesis[LH06] and surface editing [SCOL$^+$04] etc. In this research, there is no need to transfer surface attribute between meshes of different shapes. The shape of the target mesh is similar to the source mesh. The traditional texture transfer techniques such as parameterization mapping based method [Ale02] and texture synthesis [LH06] will inevitably cause distortion. The parameterization mapping based method will preserve large scale pattern in textures but suffer from local scale distortion. The texture synthesis method is able to reproduce the small scale details of the texture but can not well preserve the large scale patterns. Although those methods may transfer color from source mesh to the target mesh, the source mesh's UV parameterization pattern may not be preserved which will influence the existing pipeline or workflow in game or visual effect industries. During the texture transfer procedure, obvious texture stretch across the UV seam will occur because the neighbouring points on the source mesh's surface may be far away from each other in the target mesh's parameterization space (UV space) which is the main difficulty for this challenge. Although there are many researches focusing on the solving the artifact caused by the UV seam [RNLL10][LH06][GP09][SP06], most of them only solve the discontinuity around UV seam on single mesh rather than the texture stretch problem during attribute transfer between different meshes.

In this research, an automatic simulation ready model generation pipeline is introduced which can convert the poor quality anatomic model into simulation ready model while preserving the original model's surface parameterization property during the transfer procedure. The pipeline can be treated as a blackbox and does not influence traditional artistic workflow. Also, it will liberate artists from tedious works and improve the working efficiency quite a lot. The pipeline is composed of two stages:

- A voxelization and remesh based pipeline which can keep the shape of original 3D surface model but eliminate the ill shaped and degenerate polygons (self intersection, inter-penetrations etc.) without influencing existing artistic pipeline.

- A surface mesh parameterization transfer algorithm which can transfer the original surface parameterization (UV mapping) to the simulation ready model without distortion in the parameterization space.

3.2 Voxelization Based Mesh Optimization

The common way of constructing the volume representation is to convert the explicit geometry representation into a signed distance function $\phi(\mathbf{x})$. The signed distance function computes the minimum distance from a given point $\mathbf{x}$ to the boundary ($\phi(\mathbf{x}) = 0$) on the mesh, with a positive value sign outside the domain and negative value sign inside the domain. The signed distance function can be obtained through the propagation in the form of :$\frac{\partial\phi(\mathbf{x})}{\partial t} = -\mathbf{v}\nabla\phi(\mathbf{x})$, which means propagate a surface $\phi(\mathbf{x})$ with the speed $\mathbf{v}$ in the direction of $\nabla\phi(\mathbf{x})$ [ZOF01].

For the input non-manifold geometries with self-intersections and degenerate element, they are firstly converted from the explicit poly-

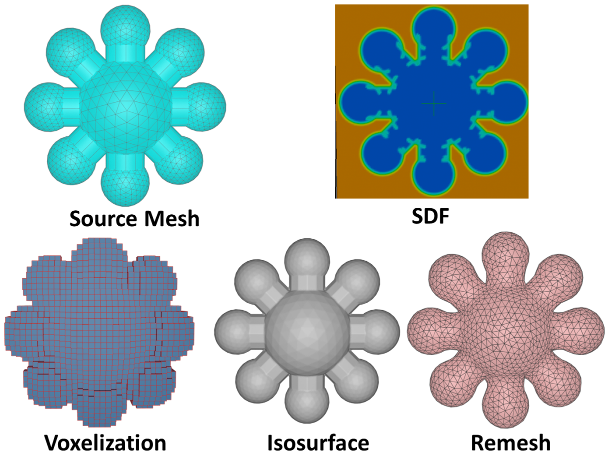

Figure 3.1: Voxelization and remesh based simulation ready model generation pipeline.

gon presentation into a signed distance function $\phi(\mathbf{x})$ (level set) using scan conversion [CCI08][HNB$^+$06]. The distance field is everywhere in the space. The gradient of the signed distance function can provide geometric information. For example, on the geometry surface, $\nabla\phi(\mathbf{x})$ represents the normal to the surface. The resolution to sample the SDF will determine how well the volumetric data represent the shape of the input polygon geometry. In the center of each voxel data, a distance field sampled value is stored.

After building the signed distance function for input geometries (see SDF result in Figure 3.1), the SDF can easily get rid of self-intersections and degenerate elements of the input geometries (called *source mesh*) using the simple and fast topology operations between distance fields such as the union, difference and intersection (see Figure 3.2).

After the topology modification, the volumetric representation of the desired object's shape is obtained. Then the volumetric data needs to be converted back into the polygon mesh, which is a process of surfacing the volumetric data along the specified iso value. However, there are

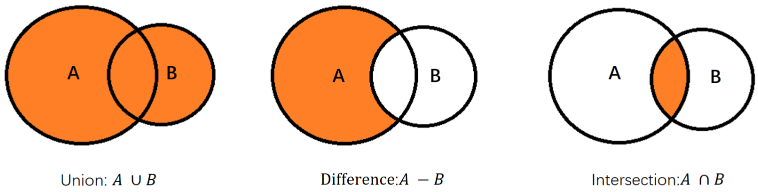

Figure 3.2: Illustration of SDF operations: union, difference and intersection.

two challenges for the output isosurface.

- The quality of the isosurface is dependent on the resolution of the voxel size.
- The new mesh will lost the texture coordinate information (UV coordinate) in the original mesh.

In order to maintain the source mesh's shapes and details of the input polygon geometries, small voxel size is often used, which means the polygon number of the extracted isosurface normally will have high density polygons (see the isosurface in Figure 3.1). Here the remesh technique is applied to reduce the polygon count while preserve isosurface's shape. This remeshed model can be used as a simulation ready model, which is called as the *target mesh*.

However, the generation of the target mesh will result in the useless of the source mesh's parameterization information. Unwrapping the target mesh and create new texture parameterization for it is quite inefficient which is no easier than fixing mesh degeneracies piecewise. In the following part, a surface attribute transfer method will be introduced which can transfer the source mesh's surface parameterization information to the target mesh.

3.3 Parameterization Transfer

3.3.1 Basic Notion

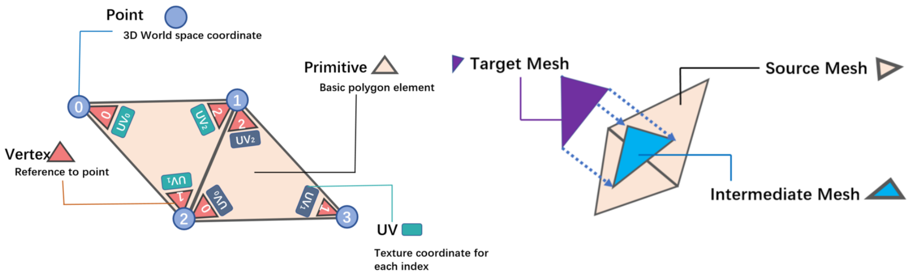

Figure 3.3: Basic Notion of Geometry

Before introducing the method, some basic geometric notions are firstly introduced. As can be seen in Figure 3.3, point is simply a point in world space as defined by a 3D coordinate. Vertex is an integer reference to the point. Primitive consists of a group of vertices that indicate the basic element of the mesh. The point can be shared among primitives but the vertices are unique in each primitive which means a point can be referenced by several different vertices in different primitives. In each primitive, each vertex has its own texture coordinate (UV coordinate). The original input mesh is defined as the *source mesh* and the remeshed isosurface generated from the voxelization step is defined as *target mesh.* The projection of the target mesh onto the source mesh's surface is defined as the *intermediate mesh* which is used to transfer attribute from source mesh to the target mesh. For each vertex on the intermediate mesh, it lays inside a primitive on the source mesh. This corresponding primitive on the source mesh is defined as the *host primitive* for intermediate mesh's vertex.

3.3.2 Method Overview

To transfer the surface parametrization (UV mapping) information from the source mesh to the target mesh, it is natural to transfer the surface attribute from the source model to the closest location on the target mesh. However, this method will also transfer the UV seam from the source mesh to the target mesh, which will result in the UV space stretch for the polygons that cross the UV seam on the target mesh. The idea is projecting the target mesh onto the source mesh and dissecting the target mesh based on how the intermediate mesh is dissected by the UV seam's corresponding edges (defined as *UV seam edges*) on source mesh, which can completely eliminate the UV space stretch artifact. Then use the newly generated dissected edge as the hard edges and feed the dissected target mesh into a remesher to generate the final good quality simulation ready model, which can completely inherit the surface parameterization of the source mesh without distortion.

3.3.3 UV Island Based Surface Parameterization Transfer

When performing the surface transfer operation, identify the location where the surface attribute is transferred from is important. Due to the shape of the target mesh is conform to that of the source mesh, it is natural to transfer the surface attribute from the source mesh to its closest location on the target mesh. The transfer process is performed by iterating the source mesh's vertices and piece-wisely transferring attribute to each vertex on the target mesh. For the i-th vertex on the target mesh, its corresponding point $\mathbf{x}_i$ is projected to the source mesh's surface (projection position $\mathbf{x}_i^p$). $\mathbf{x}_i^p$ is used as the intermediate for the transfer which means surface parameterization attribute is firstly transferred from source mesh to intermediate $\mathbf{x}_i^p$ and then this attribute is copied to its corresponding vertex on the target mesh. The transfer operation is based on barycentric coordinate. For each $\mathbf{x}_i^p$, a barycentric coordinate $\mathbf{w}_i$ can be computed from the *host primitive* it is projected onto, $\mathbf{w}_i \in \mathbf{R}^k$ where k is the number of vertices in this primitive (mostly $k = 3$ or 4). If k is larger than 4, the source mesh can be triangulated or quadrangulated to meet this standard. Then $\mathbf{x}_i^p$'s surface parameterization attribute ($\mathbf{uv}_i \in \mathbf{R}^2$) is computed by barycentrically interpolating the corresponding vertices' parameterization attribute on source mesh. Then assign $\mathbf{uv}_i$ to the corresponding i-th vertex on the target mesh.

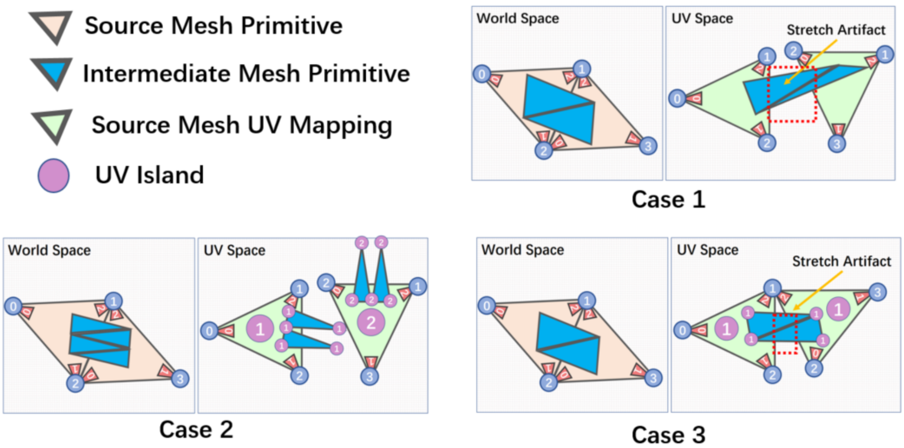

Figure 3.4: The UV space stretch artifact.

However, this method will generate artifact. As can be seen in Figure 3.4 (case 1, World Space), the *intermediate mesh's* primitive may cross the source mesh's UV seam in the world space. If the UV seam separate the source mesh into several isolated UV islands, then the primitive that cross this UV seam will be stretched in the UV space, as can be seen in Figure 3.4 (case 1, UV space). This problem is caused by the fact that the intermediate mesh's vertices in the same primitive belong to different UV islands, which means the neighbouring points on the intermediate mesh surface are far from each other in the parameterization space (UV space). When performing the barycentric interpolation for those cross UV seam primitives, their vertices will be interpolated to different UV islands.

To solve this crossing UV island stretch artifact, ensuring each intermediate mesh's primitive is barycentrically interpolated into the same UV island will effectively alleviate this artifact. Before performing the barycentric based attribute transfer, which UV island each primitive should be interpolated into is needed to be identified. For the intermediate mesh, the UV island (κ_j) is firstly labeled for each vertex by assigning the host primitive's UV island ID to it. For the labeling of the primitives that cross the UV seam, the area portion split by the UV seam is computed and the largest portion's vertex UV island ID is assigned to all the vertices in this primitive, as can be seen in Figure 3.4(case 2). Although this method may work for some models and artifact may not be obvious, it still generates zigzag UV pattern near the UV seam. However, if the UV seams do not split the mesh completely into two separate UV islands which means the area crossing the UV seam share the same UV island ID. The above method will still generate the UV space stretch artifact near UV seams, as can be seen in Figure 3.4 (case 3).

In general, when transferring UV attribute, artifact can not be avoided if the intermediate mesh's primitives cross the UV seam of the source mesh. To completely eliminate the UV space stretch artifact, there should not be any intermediate primitives cross the source mesh's UV seam. To achieve this, the intermediate mesh's primitives that cross the UV seams must be split or adjusted to align with the UV seam.

3.3.4 UV Seam Cutting Based Parametrization Transfer

To split the primitives on the intermediate mesh that cross the source mesh's UV seam, intersection is tested between the primitives and the corresponding edges of the UV seam. If the intermediate primitive intersects with the UV seam edges, corresponding primitive on the *target mesh* will be dissected into refined primitives according to how the intermediate primitive is dissected by the UV seam edges. This process is called as UV seam based *cutting transfer*. When performing the UV seam dissection, the idea proposed in [SHGS06] is used which combines the vertex snapping with the element refinement to avoid small or ill shaped primitives. For the intermediate mesh's primitive which cross the UV seam, the polygon area is measured on both side of the UV seam. If the polygon area ratio is too large or too small, the vertex will be snapped onto the seam instead of being split.

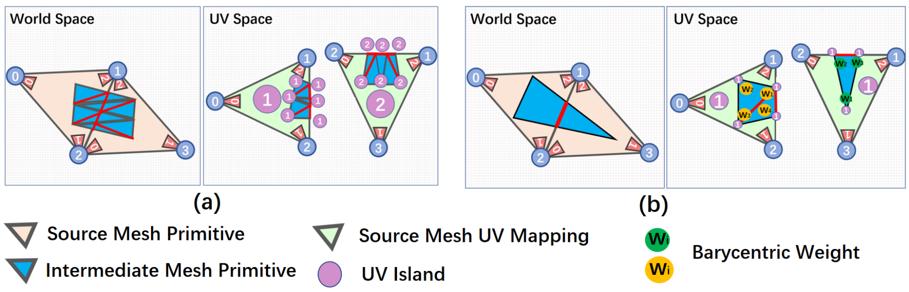

Figure 3.5: UV seam based primitive cutting transfer.

After this operation, there will be no primitives crossing the UV seam. And then the above UV island based surface parameterization transfer operation is performed (see section 3.3.3), there will be no artifact near the seam between two different UV islands, see Figure 3.5 (a). For the seam that splits the same UV island, when computing the barycentric coordinates for the newly generated vertices that lay on the UV seam, the rule that the vertices in the same primitive (on the target mesh) should use the same *host primitive* (on source mesh) to compute the barycentric coordinate and transfer attribute is used, can be seen in Figure 3.5 (b). If not use this rule, the vertices in the same primitive will receive attribute from two different primitives on the source mesh, resulting in UV space stretch again.

Then the remesh technique [Per06] is applied to optimize the shape of polygon by using the edge that lay on the UV seam as hard edge. The hard edge may be subdivided by remesher according to the user-specified edge length for remesher, but the hard edge shape will be preserved which can avoid the UV space stretch. Till now the 3D anatomy models with self-intersections and degenerate elements can be converted into a simulation ready model using this pipeline while preserving the original mesh's surface parameterization. The whole procedure of the cutting based surface parameterization transfer has been summarized in algorithm 1.

3.4 Experiment and Results

3.4.1 Voxelization

The anatomy model used is a human kidney which includes many parts such as renal pelvis, adrenal gland, pyramid, artery and vein. The quality of this kidney anatomy model is fair for rendering. However, it does not meet the requirement of physics simulation because it includes the irregular and ill shaped polygons, self-intersection and not watertight mesh etc., see Figure 3.6.

The voxelization can be performed separately on each part of the source mesh or the whole mesh according to the requirement of simulation. Due to the fact that each part of the model may have different resolutions, perform voxelization on the whole mesh may cause the loss

Algorithm 1 Cutting Based Surface Parametrization Transfer

Definition: target mesh (**T**), source mesh (**S**), barycentric coordinate (**w**), uv coordinate (**uv**).
procedure UVTransfer((**T**, **S**))
  Project **T** onto **S** to get intermediate mesh **I**
  for all primitive $\pi_i \in \mathbf{I}$ **do**
    if π_i intersect with UV seam **then**
      Split π_i into π_i^k $(k = 1, 2)$
    if π_i^k $(k = 1, 2)$ is ill shaped **then**
      Perform vertex snapping
    for all vertices $\mathbf{x}_j \in \pi_i^k (k = 1, 2)$ **do**
      Compute $\mathbf{w}_j$ for $\mathbf{x}_j$
      Interpolate $\mathbf{uv}_j$ based on $\mathbf{w}_j$
      Transfer $\mathbf{uv}_j$ back to corresponding primitive on **T**
    for all edge $\mathbf{e}_m \in \pi_i^k (k = 1, 2)$ **do**
      if $\mathbf{e}_m$ coincides with uv seam **then**
        Mark $\mathbf{e}_m$ as hard edge
    Feed **T** and all $\mathbf{e}_m$ into remesher
    Return simulation ready model **T**

of detailed geometric features and change the source mesh's geometry group information which will influence the art pipeline. Perform voxelization on each part of the source mesh will capture the detail feature and make each part of the model simulation ready. For the purpose of better illustration, the experiment is only performed on one part of the mesh (renal pelvis). For other part of the mesh, the same operation can be performed. In Figure 3.7, high resolution voxel can capture the detail of mesh well which can be seen from the SDF and corresponding isosurface. The quality of the isosurface will directly influence the final

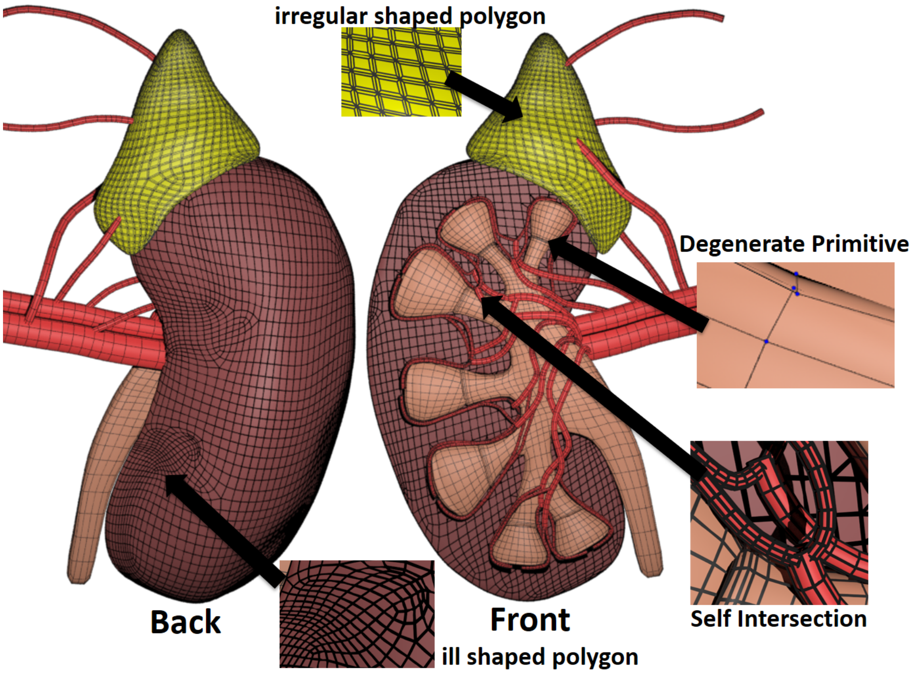

Figure 3.6: Illustration of the artifacts in the experiment model.

result of the remesher. When sending isosurface to remesher [Per06], the edge length of remesher can be fixed or adaptive. For the adaptive remesh, the quality of the result is controlled by the gradation [Per06] which means the rate that edge lengths are allowed to change from one primitive to another. The more accurate the isosurface is, the better the remeshed model will approximate the source model. It can be seen from Figure 3.7 that both fixed and adaptive edge length can well capture the shape of the source mesh even choose large edge length for remesher. Although the shape of the remeshed model is not completely the same as the source mesh, such small variance can satisfy the need of most virtual surgery simulator for training purpose. The subtle shape different can be solved using a rendering technique like displacement mapping. It will compute the displacement from the target mesh to the source mesh along the local surface normal. When rendering, the

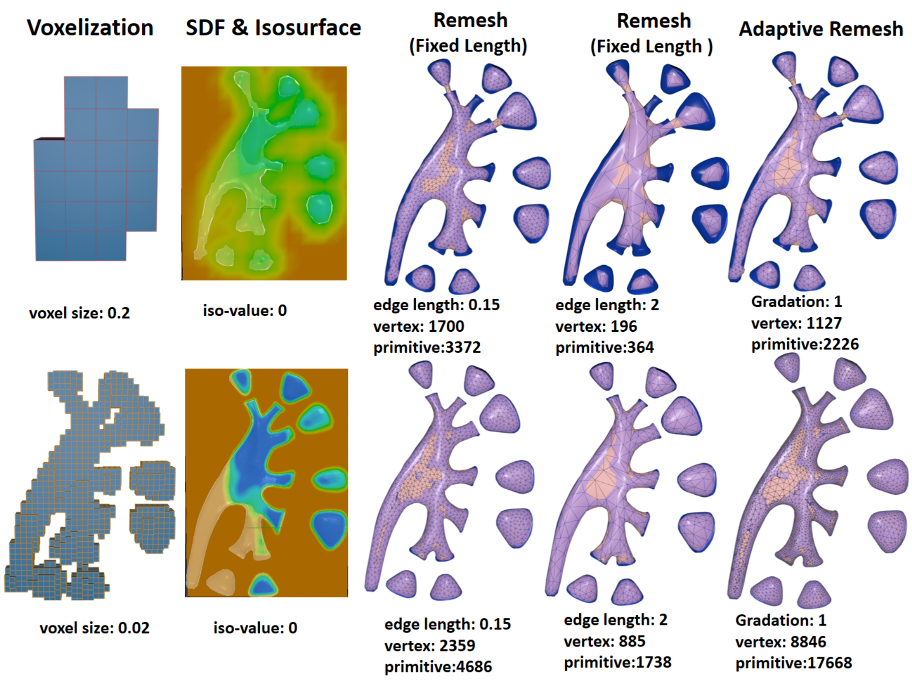

Figure 3.7: voxelization based mesh optimization pipeline and remeshed results. In the remeshed result, the shape of source mesh (translucent purple layer) and remeshed model (solid pink layer)

is compared.

target mesh will be displaced to the shape of source mesh.

3.4.2 Surface Attribute Transfer

It can be seen from Figure 3.8 (b) that transferring the UV parameterization without using the UV island information will generate the UV space stretch and obvious object space artifact. By taking into consideration the UV island information (Figure 3.8 (c)), the UV space stretch artifact can be alleviated but the zigzag artifact near the UV seam and stretch artifact still exist on the same UV island. In Figure 3.8 (d), it can be seen that the proposed method can well eliminate the UV space stretch. Also, the proposed method can preserve the original mesh's UV density without any detail loss and distortion, as can be seen in Figure 3.9.

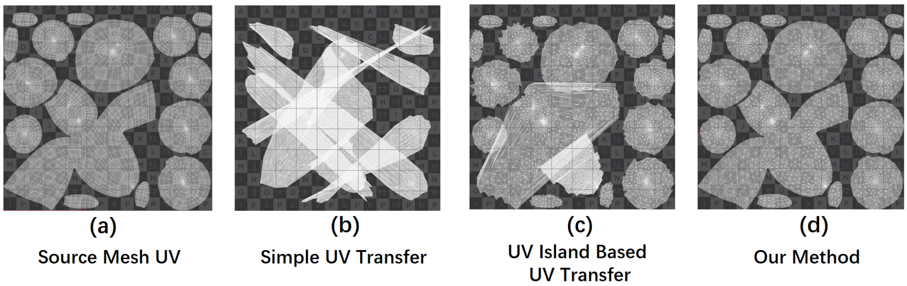

Figure 3.8: Surface attribute transfer results.

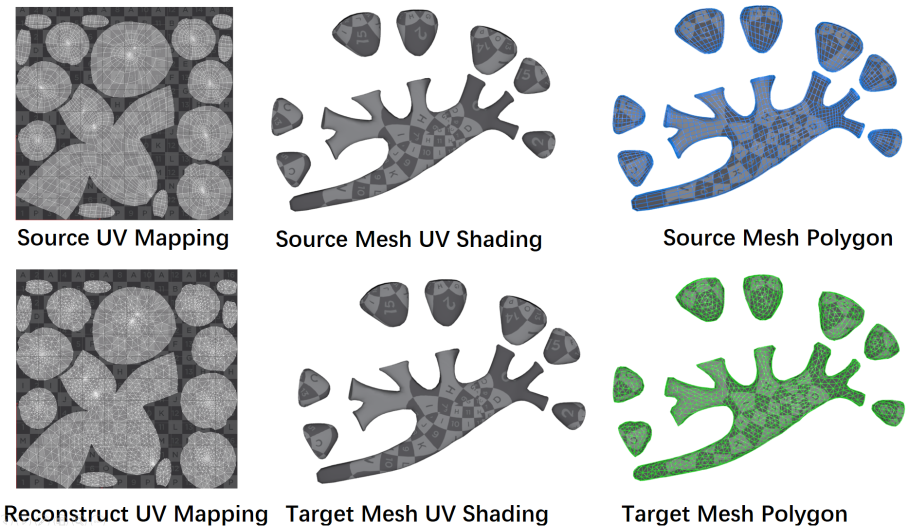

Figure 3.9: UV shading and polygon quality comparison.

In Figure 3.10, when applying the proposed method on each part of the kidney model, a simulation ready model can be generated, which has good quality polygon discretization and completely inherit the source model's surface parameterization. Figure 3.10 also show the result of simulating the target mesh using finite element method [SB12a].

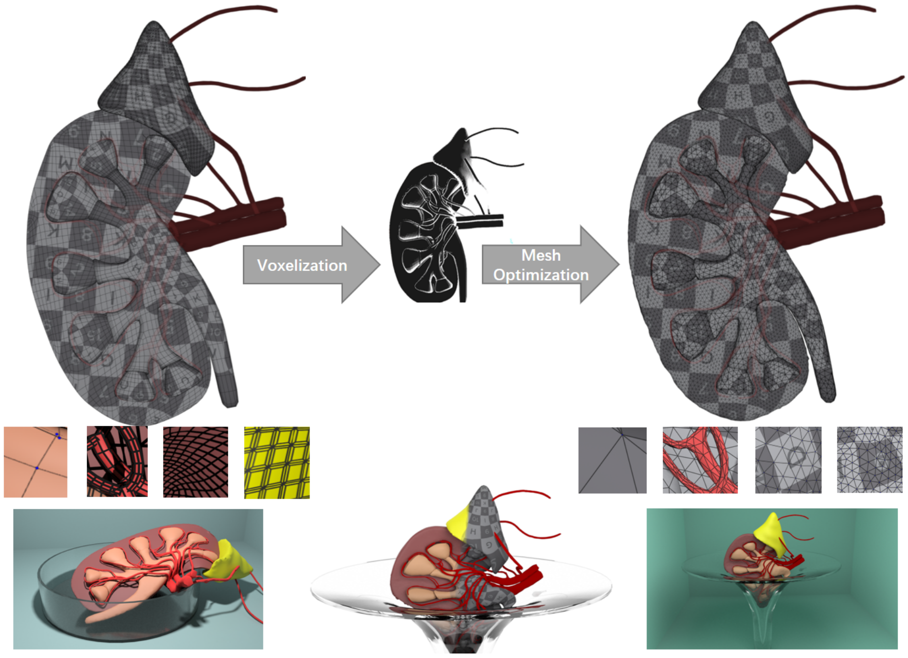

Figure 3.10: Applying the proposed method on each part of the kidney model and the simulation result.

3.5 Summary

In this chapter, a simulation ready model generation pipeline is proposed which can automatically convert poor quality 3D anatomical models into simulation ready state while preserving the original model's surface parameterization attribute. This pipeline includes two stages. The first stage is a voxelization and remesh based simulation ready model generation pipeline which can keep the shape of original 3D surface model while eliminate the ill shaped and degenerate polygons (self intersection, inter-penetrations etc.) without influencing existing artistic pipeline. The second stage is a surface mesh parameterization transfer algorithm which can transfer the original surface parameterization (UV mapping) to the simulation ready model without distortion in the parameterization space.

Chapter 4

Realtime Soft Tissue Simulation

4.1 Introduction

Soft tissue simulation, as the core of laparoscopic surgery simulator, has been an active research topic in computer graphics and continuum mechanics for decades. Due to the realtime performance requirement of surgery simulator, the top priority of choosing simulation method is the realtime performance for the consideration of fluent user experiences. This limits the use of physically accurate but computationally expensive simulation methods which can provide high fidelity simulation result. The common strategy used in most simulator is to compromise simulation accuracy to efficiency but find a acceptable balance point which the user can accept the simulation result while keeping the realtime performance of the system. The compromise on accuracy makes the tuning of soft tissue's physical behaviour is very important because achieving the user desired simulation effect is the goal of surgery simulation.

There are many successful realtime physics simulation engines (such as Havok, PhysX, Bullet, FLEX etc.) on the market for game and visual effect industries. Although there are some cases of developing surgery simulator based on those engines [MHL+09] [MWW07] , they do not well supported soft body dynamics such as FEM based soft body simulation which limit their potential in physically accurate high fidelity simulation. The soft body simulation in existing physics engine are mainly based on the simulation framework of position based dynamics which is an iterative based constraint solving system [MHHR07a][MMCK14][BML+14]. The type of constrain will determine the physical behaviour of the objects. The spring based constrains[KCM12] [MCMJ17] (such as distance, bending, shear etc.) and shape based constraint (such as shape matching, volume preservation) [MHTG05b][CMM16][MC11] are the main constraint types in those engines for simulating elastic behaviour. Compared to continuum mechanics based materials, those simulation engines only have very few material models which limited the types of soft tissue behaviour that can be tuned. The users have to feedback their desired result to manufactures and wait for system update. Compares to game engines, SOFA [1] is targeted at realtime simulation with an emphasis on continuum mechanics based soft body simulation. However, it only provides a few constitutive models but does not have an intuitive material tuning strategy and a mechanism for simulating complex physical behaviour such as heterogeneous property of soft tissue which is common in real surgery.

In this chapter, a multi-layer based soft tissue simulation pipeline is proposed. Different from other simulators which control soft tissue behaviour using certain types of material constitutive, the physical behaviour of the soft tissue in this research is controlled by a more general

[1] Software Open Framework Architecture, https://www.sofa-framework.org/.

and polynomial based material which can represent various widely used hyperelastic materials. Based on this material model, a heterogeneous soft tissue simulation strategy has been proposed to simulate complex physical behaviour of soft tissue. The contributions can be summarized as:

- A multi-layer soft tissue modeling method has been proposed which generalizes the soft tissue into fascia, fatty and embedded tissues and provide geometric based connection strategy for different types of tissues.
- A Valanis-Landel hypothesis based soft tissue simulation framework has been proposed. It enables the user to simulate various types of soft tissue behaviours using one uniform polynomial based material representation and tuning the physical behaviour via an intuitive control system based on curve editing.
- A heterogeneous soft tissue generation strategy is proposed. It can produce heterogeneous stiffness distribution on soft tissue via a skeleton based painting system.

4.2 Multi-Layer Soft Tissue Modeling

Soft tissues play the role of connecting, supporting and wrapping other structures and organs of the body as can be seen in Figure 4.1. It normally includes tendons, ligaments, fascia, skin, fibrous tissues, fat, membranes (connective tissue), muscles, nerves and blood vessels (non-connective tissue) etc.[Mar07]. In laparoscopic surgery, the fascia, membrane and fat will be dissected to set a clear path to the cancer part. During the dissection process, some important nerves, blood vessels, ligaments, glands must be avoided, which normally lay beneath the

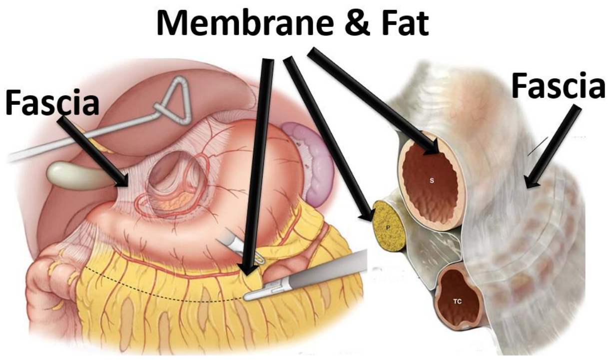

Figure 4.1: Multi-layer soft tissue structure.

membrane or embed inside the fatty tissue. According to these facts, the soft tissue is generalized into three groups in this research for simulation: *fascia*, *fatty tissue* and *embedded tissue*.

For the *fascia*, it refers to a band of connective tissues beneath the skin that attach, stabilize, enclose, and separate muscles and other internal organs, such as ligaments, aponeuroses, and tendons. The fascia is mainly composed of fibrous connective tissue which encloses packed bundles of collagen fibers parallel to the direction of pull. The elastin in the fascia confers stiffness to it and store most of the energy. Due to the inextensibility of the collagen fibers, the tension caused by the taut fascia will grow when the collagen is gradually pulled and stretched in one direction. The fascia mainly can be categorized into three types: superficial fascia, deep fascia and visceral fascia. The superficial fascia can be found in the subcutis in many parts of the body. Deep fascia is the dense fibrous connective tissue that interpenetrates and surrounds the muscles, bones, nerves and blood vessels of the body [MH07]. Visceral fascia is responsible for the suspension of organs and wraps them in layers of connective tissue membranes.

For the *fatty tissue*, it mainly includes water, carbohydrate and protein which makes it like a hyperelastic object combined with damping resistance. When the stretch of fatty soft tissue exceed certain limit, it will not recover to its initial shape.

For the *embedded tissue*, it can be nerves, blood vessels and glands that travel through the connective tissues or fat. Those embedded tissue can provide an avenue for communication and transport among other tissues.

For the modeling of *fascia*, triangular mesh is used to represent its structure which is normally a thin film with lubricating fluid on the surface. For the *fatty tissue* which is wrapped up by the connective tissues, it is solid object and its volumetric behaviour is needed to be captured during simulation. Thus, the tetrahedral mesh is utilized as the discretization form to represent the *fatty tissue*. For the *embedded tissue*, tetrahedral mesh is also used to represent its tubiform shape.

The multi-layer soft tissue structure will cause a problem when performing tetrahedralization to the mesh because the boundary of the embedded tissue is inside the boundary of the fatty tissue. When performing the tetrahedralization, it needs to be conformal to the boundary of both tissue. However, this often incur high amounts of tetrahedral elements if the embedded tissue has high resolution mesh. To solve this problem, the idea of sharing tetrahedron between different types of tissues is used. The fatty tissue and embedded tissue are tetrahedralized separately. Some of the fatty tissue's tetrahedrons will be occupied by the embedded tissue. When simulating the embedded tissue, those occupied tetrahedrons are treated as cages just like the free-form deformation [MRT$^+$10]. The simulation result of each tetrahedron is weighted by the contribution of different tissues that share this tetrahedron, as can be seen in Figure 4.2.

For the fascia that connect and support other tissue, a connection strategy between different surface meshes is needed. Due to the fascia

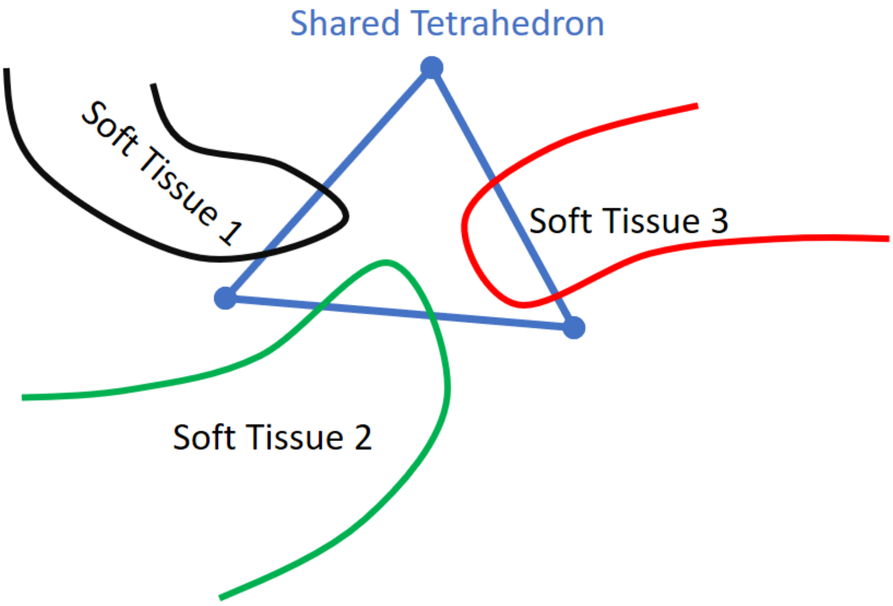

Figure 4.2: The idea of sharing tetrahedron between different types of tissue.

and boundary of fatty tissue are triangular mesh, a barycentric based mapping method is proposed to connect them. Firstly, each point $\mathbf{x}_i$ on the fascia's mesh is projected onto the boundary of fatty tissue (the projected position $\mathbf{x}_i^p$). Assuming $\mathbf{x}_i^p$ belongs to a triangle t with position $\mathbf{X}_j^t \in \mathbf{R}^3$ at vertex $j = 1, 2, 3$. A barycentric coordinate $\mathbf{w}_i \in \mathbf{R}^3$ between $\mathbf{x}_i^p$ and $(\mathbf{X}_1^t, \mathbf{X}_2^t, \mathbf{X}_3^t)$ can be computed.

$$\mathbf{w}_i = \mathbf{A}^{-1}\mathbf{x}_i \tag{4.1}$$

Where $\mathbf{A} = (\mathbf{X}_1^t, \mathbf{X}_2^t, \mathbf{X}_3^t)$. When the fascia moves, the displacement of $\mathbf{x}_i$ can be barycentrically distributed to the points of its corresponding triangle $(\mathbf{X}_1^t, \mathbf{X}_2^t, \mathbf{X}_3^t)$ and vice versa. The simulation of those tissues are based on continuum mechanics which treats the soft tissues are solid object. More details can be found in the following section.

4.3 Physics Based Soft Tissue Property Modeling

4.3.1 Hyperelastic Based Soft Tissue Modeling

It is axiomatic in mechanics that the response of a material to applied loads depends upon its internal constitution, that is, the distributions, orientations and interconnections of its microstructural components [Hum03]. Similar to continuum mechanics, soft tissue respects the basic postulates of mechanics (such as energy, momentum, conservation laws), and the basics concepts such as stress strain also apply. The physical property of soft tissue is different from traditional engineering materials such as metal, concrete etc. Most soft tissue exhibit nonlinear, anisotropic, heterogeneous characteristic that varied from point to point, time to time and individual to individual. Those physical behaviours are similar to the behaviours exhibited by elastomers. Many advances in soft tissue mechanics come from advances in rubber elasticity.

Hyperelasticity is a constitutive model which is used to describe the relation of ideally rubber-like elastic material. The hyperelastic material models the stress and strain relationship using strain energy function. The stress in a hyperelastic material is independent of the path or history of deformation. The relationship of stress and strain can be defined as isotropic, anisotropic, orthotropic and incompressible. In the biomechanics, the material identification of soft tissue is often made under the assumption that the tissue is hyperelastic material [WWM79][Hum98]. In computer graphic research, biological tissues are also often modeled via the hyperelastic idealization [SB12b]. In this thesis, the hyperelastic material is chosen to model the physical property of soft tissues.

For hyperelastic material, the relationship of stress and strain can be defined as isotropic, anisotropic, orthotropic and incompressible. St. Venant-Kirchhoff, Neo-Hookean, Mooney–Rivlin, Corotated linear etc. are typical hyperelastic materials. The isotropic material shows equal resistance to deformation along all possible directions. Rotational invariant is an important property of isotropic material. For anisotropic material, it demonstrates different properties in different directions. The orthotropic material is a subset of anisotropic material.

To mathematically describe a soft body, the soft body object is put into a 3D coordinate system. The space occupied by the object is donated as Ω. This domain will be referred to as the reference configuration. $\mathbf{X} \in \mathbf{R}^3$ is used to represent the individual material point in the rest state of the soft object. When the object undergoes deformation, $\mathbf{X}$ is displaced into a new state $\mathbf{x}$, deformed configuration. A deformation function $\phi : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ maps the elastic object in rest configuration to deformed configuration can be defined as $\mathbf{x} = \phi(\mathbf{X})$. To better describe the deformation, a physical quantity called deformation gradient $\mathbf{F}$ can be derived from the deformation mapping function $\phi(\mathbf{X})$:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \tag{4.2}$$

Deformation is a procedure of potential energy accumulation. This accumulated energy, referred to as strain energy, is a function of the deformation mapping function $E(\phi(\mathbf{X})) \in \mathbf{R}$. To measure the strain energy per unit volume according to unit undeformed volume, an energy density function is defined as $\Psi(\phi(\mathbf{X}))$. The deformation energy inside a deformed object can be calculated as:

$$E(\mathbf{X}) = \int_{\Omega} \Psi(\phi(\mathbf{X})) d\mathbf{X} \tag{4.3}$$

The elastic force for a material point $\mathbf{x}$ can be calculated as the negative gradient of elastic energy.

For a hyperelastic material, the energy density function Ψ is integrated over the entire domain Ω of the elastic body which can be discretized into elements such as tetrahedron, hexahedron etc. Due to the mesh used in research are triangular, tetrahedron based discretization is used in this thesis which is a relative simple element with non-planar surfaces and has been widely used in computer graphics research [ITF04][BWHT07][SB12a]. So the energy can be expressed as the function of the discretized degree of freedom. For a tetrahedron element, the discrete deformation mapping for its four nodes can be expressed as a piece linear function (see Figure 4.3):

$$\mathbf{x}_i = \mathbf{F}\mathbf{X}_i + \mathbf{b}, i = 1, 2, 3, 4 \tag{4.4}$$

Where $\mathbf{X}_i$ and $\mathbf{x}_i$ are the rest and deformed state position respectively,

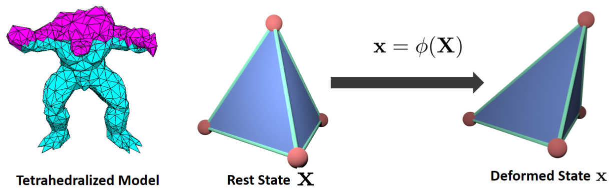

Figure 4.3: Demonstration of tetrahedralized model and deformation mapping on tetrahedron element.

$\mathbf{F} \in \mathbf{R}^{3\times3}$ is the deformation gradient and $\mathbf{b} \in \mathbf{R}^3$ is the element translation. It can be also reformulated as:

$$(\mathbf{x}_1 - \mathbf{x}_4|\mathbf{x}_2 - \mathbf{x}_4|\mathbf{x}_3 - \mathbf{x}_4) = F(\mathbf{X}_1 - \mathbf{X}_4|\mathbf{X}_2 - \mathbf{X}_4|\mathbf{X}_3 - \mathbf{X}_4) \tag{4.5}$$

By donating $(\mathbf{x}_1 - \mathbf{x}_4|\mathbf{x}_2 - \mathbf{x}_4|\mathbf{x}_3 - \mathbf{x}_4)$ as $\mathbf{D}_s$ and $(\mathbf{X}_1 - \mathbf{X}_4|\mathbf{X}_2 - \mathbf{X}_4|\mathbf{X}_3 - \mathbf{X}_4)$ as $\mathbf{D}_m$, the deformation gradient can be calculated as:

$$\mathbf{F} = \mathbf{D}_s\mathbf{D}_m^{-1} \tag{4.6}$$

The strain energy E for a tetrahedron element T is determined by the energy density function Ψ which will determine the physical property of object:

$$E = \int_T \Psi(\psi) d\mathbf{X} = \Psi(\psi) \int_T d\mathbf{X} = W \cdot \Psi(\psi) \tag{4.7}$$

Where W is the volume of T. The force applied on the vertex $\mathbf{x}_i$ is computed as the negative gradient of the energy:

$$\nabla_{\mathbf{x}_i} E = W \frac{\partial \mathbf{E}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{x}_i} = W \mathbf{P}(\mathbf{F}) \frac{\partial \mathbf{F}}{\partial \mathbf{x}_i} \tag{4.8}$$

where $i = 1, 2, 3$, $\mathbf{P}(\mathbf{F})$ is the first Piola-Kirchhoff stress tensor which is dependent on the constitutive model. $\mathbf{x}_i^{(j)}$ is donated as the jth element of $\mathbf{x}_i$. Then the derivative of $\mathbf{F}$ with respective to $\mathbf{x}_i^{(j)}$ can be computed as:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}_i^{(j)}} = \mathbf{e}_j \mathbf{e}_i^T \mathbf{D}_m^{-1} \tag{4.9}$$

Where $\mathbf{e}_i \in \mathbf{R}^3$ is a selection vector. The ith element of $\mathbf{e}_i$ is 1 while other elements is zero. For example, $\mathbf{e}_1 = (1, 0, 0)^T$. $\nabla_{\mathbf{x}_i^{(j)}} E$ can be computed as:

$$\nabla_{\mathbf{x}_i^{(j)}} E = \mathbf{P}(\mathbf{F}) \colon \mathbf{e}_j \mathbf{e}_i^T \mathbf{D}_m^{-1} = tr(\mathbf{e}_j^T \mathbf{P}(\mathbf{F}) \mathbf{D}_m^{-T} \mathbf{e}_i) = \mathbf{e}_j^T \mathbf{P}(\mathbf{F}) \mathbf{D}_m^{-T} \mathbf{e}_i \tag{4.10}$$

$\nabla_{\mathbf{x}_i^{(j)}} E$ equals to the element of $\mathbf{P}(\mathbf{F}) \mathbf{D}_m^{-T} \in \mathbf{R}^{3\times3}$ which locates at the ith row and jth column. So the force on the tetrahedral element node can be expressed as:

$$\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3] = -W \mathbf{P}(\mathbf{F}) \mathbf{D}_m^{-T}, \mathbf{f}_4 = -\mathbf{f}_1 - \mathbf{f}_2 - \mathbf{f}_3 \tag{4.11}$$

The gradient of the elastic force is called the tangent space stiffness matrix, which needs to be computed in numerical solution (interpreted as the Hessian of energy). It can be computed as:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = W \left(\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}} \mathbf{D}_m^{-T} \right) \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \tag{4.12}$$

$\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ can be computed based on equation 4.9. For the computation of $\partial \mathbf{P}(\mathbf{F})/\partial \mathbf{F}$, it can be derived as:

$$\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}_{ij}} = \frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}} \mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^{\mathbf{T}} + \mathbf{U}\frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \mathbf{F}_{ij}}\mathbf{V}^{\mathbf{T}} + \mathbf{U}\mathbf{P}(\mathbf{F})\frac{\partial \mathbf{V}^{\mathbf{T}}}{\partial \mathbf{F}_{ij}} \tag{4.13}$$

where $\mathbf{P}(F) = \mathbf{U}\mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^{T}$ because $\mathbf{P}(\mathbf{F})$ is rotational invariant for isotropic constitutive model. In the above equation, $\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}}$, $\frac{\partial \mathbf{V}^{\mathbf{T}}}{\partial \mathbf{F}_{ij}}$ and $\frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \mathbf{F}_{ij}}$ are needed to be computed. Those variables can be derived from the differentiation of deformation gradient $\mathbf{F}$'s SVD ($\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^{\mathbf{T}}$).

$$\mathbf{U}^{\mathbf{T}}\frac{\partial \mathbf{F}}{\partial \mathbf{F}_{ij}}\mathbf{V} = (\mathbf{U}^{\mathbf{T}}\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}})\hat{\mathbf{F}} + \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{F}_{ij}} + \hat{\mathbf{F}}\frac{\partial \mathbf{V}^{\mathbf{T}}}{\partial \mathbf{F}_{ij}}\mathbf{V} \tag{4.14}$$

Due to $(\mathbf{U}^{\mathbf{T}}\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}})$ and $\frac{\partial \mathbf{V}^{\mathbf{T}}}{\partial \mathbf{F}_{ij}}\mathbf{V}$ are antisymmetric matrix which means the diagonal elements are zeros. So the $\frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{F}_{ij}}$ equals to the diagonal elements of $\mathbf{U}^{\mathbf{T}}\frac{\partial \mathbf{F}}{\partial \mathbf{F}_{ij}}\mathbf{V}$. $\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}}$ and $\frac{\partial \mathbf{V}^{\mathbf{T}}}{\partial \mathbf{F}_{ij}}$ can also be solved from a 2×2 symmetric matrix.

Now the computation methods for all the necessary variables $(\nabla E, \nabla f)$ for computing the physical behaviour of isotropic material have been listed. There are many widely used isotropic materials which has been summarized in Figure 4.1.

For the simple model, it is not rotationally invariant which means it will treat rotation as deformation (when $\mathbf{F} = \mathbf{R}$, $\frac{1}{2}k\|\mathbf{F} - \mathbf{I}\|_F^2$ is non-zero). For the linear elasticity model (also called small strain model), ε just provides a linear approximation of Green Strain which is not completely rotational invariant and only acceptable under small scale rotation. For the St. Venant-Kirchhoff model, it is rotational invariant but not reflection invariant because $\mathbf{F}^T\mathbf{F} = \mathbf{U}\Sigma^2\mathbf{V}^T$ (Σ^2 will eliminate the negative value in Σ, $\mathbf{U}$ and $\mathbf{V}$ are rotation matrices by performing SVD to $\mathbf{F}$). When element is inverted during simulation, the energy will remain the same which will cause inversion artifact. For the co-rotated linear elasticity, it is both rotational invariant and reflection

Constitutive Model			
Model	Strain Measurement (ε)	Energy Density	First Piolar Kirchhoff Stress Tensor
Simple Model	$F - I$	$\frac{1}{2}k\|F-I\|_F^2$	$K(F-I)$
Linear Elasticity	$\frac{1}{2}(F+F^T) - I$	$\mu\|\varepsilon\|_F^2 + \frac{\lambda}{2}tr^2(\varepsilon)$	$2\mu\varepsilon + \lambda tr(\varepsilon)I$
St. Venant-Kirchhoff model	$\frac{1}{2}(F^TF - I)$	$\mu\|\varepsilon\|_F^2 + \frac{\lambda}{2}tr^2(\varepsilon)$	$F[2\mu E + \lambda tr(E)I]$
Cororated Linear Elasiticity	$\Sigma - I$ $(F = U\Sigma V^T)$	$\mu\|\varepsilon\|_F^2 + (2/\lambda)tr^2(\Sigma - I)$	$2\mu(F - R) + \lambda tr(R^TF - I)R$
Neohookean elasticity	$I_1 = tr(F^TF), I_3 = \|F\|^2, J = \|F\|$	$\frac{\mu}{2}(I_1 - log(I_3) - 3) + \frac{\lambda}{8}log^2(I_3)$	$\mu(F - \mu F^{-T}) + \lambda log(J)F^{-T}$

Table 4.1: The summary of the most used constitutive models.

invariant. However, it needs to perform polar decomposition for the deformation gradient when computing the strain and the energy. For the Neohookean elasticity, polar decomposition of deformation gradient is still needed. When element is heavily compressed, the energy will increase dramatically to penalize such compression and react strongly to it. This will result in numerical issue during simulation.

4.3.2 General Material Modeling

To meet the requirement of realtime simulation, compromise on accuracy has to be made. For laparoscopic surgery simulation, physically accurate soft tissue simulation is not top priority compare to the realtime performance requirement. Instead of being physically accurate, physically plausible simulation is more appealing to realtime application. The physically plausible simulation means the result of simulation is not converged but have the tendency of convergence when given enough computation iteration. Due to the fact that physically plausible simulation result is not physically accurate, the procedure of system tuning is used as a necessary supplementary to achieve user desired effect. However, tuning the physical behaviour based on the continuum mechanic equations is not an intuitive way especially for users from medical field.

In laparoscopic surgery simulation, an intuitive physically based user control is very important. The soft tissue's physical behaviour is determined by the strain-stress relation which are often described by curves when performing the material property identification experiment[Fun13]. Working directly on the energy density function Ψ is cumbersome and not intuitive. Valanis-Landel hypothesis separates the strain energy function into separate functions of principle stretch [BCCB06]. This offers the possibility of linking curve based editing technique to the editing of energy function in each direction of principle stretch. The curve based editing technique is widely used in computer animation industry which can provide animators and artists with intuitive control to achieve desire effect. Valanis-Landel hypothesis is defined as:

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{i=1}^{i=3} f(\lambda_i) + g(\lambda_1\lambda_2) + g(\lambda_2\lambda_3) + g(\lambda_3\lambda_4) + h(\lambda_1\lambda_2\lambda_3) \tag{4.15}$$

where f, g, h are scalar energy functions for length (uniaxial), area(biaxial), and volume (triaxial strain) respectively. $\lambda_1, \lambda_2, \lambda_3$ represent the principle stretch which is the diagonal elements of $\hat{\mathbf{F}}$ ($\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T$). The widely used St.Venant-Kirchhoff, Neo-Hookean, corotated linear material can all be expressed using the Valanis-Landel hypothesis.

For isotropic material, the density function Ψ is invariant to any permutation of the principle stretch and the Piolar-Kirchhoff is invariant under rotations of material space:

$$\mathbf{P}(\mathbf{F}) = \mathbf{P}(\mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T) = \mathbf{U}\mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^T \tag{4.16}$$

where $\mathbf{P}(\hat{\mathbf{F}})$ can be computed as:

$$\mathbf{P}(\hat{\mathbf{F}}) = \begin{bmatrix} \frac{\partial \Psi}{\partial \lambda_1} & 0 & 0 \\ 0 & \frac{\partial \Psi}{\partial \lambda_2} & 0 \\ 0 & 0 & \frac{\partial \Psi}{\partial \lambda_3} \end{bmatrix} \tag{4.17}$$

4.3.3 Curve Based Material Editing

The stress and strain relationship is normally reflected by curves in material identification process. For example, the stress and strain relationship of connective tissue usually demonstrates a multi-phase feature. As can be seen in Figure 4.4[2], the curve can be divided into toe region, linear region and failure region. The toe region is within the elastic range of the tissue. The temporary straightening of collagen fibers will restore to its rest configuration. In the linear region, it is still within the physiologic limit but the collagen fibres are permanently lengthened through internal microfailure. In the failure region, irreversible lengthening and failure of ligament or tendon occurs.

Curve based stress and strain editing can provide more intuitive way of designing soft tissue physical property. However, there is a problem when editing the separate function curves of main stretch. As can be seen in equation 4.15, the energy density should equals to zero in the rest configuration:

$$\Psi(1,1,1) = \sum_{i=1}^{i=3} f(1) + 3g(1) + h(1) = 0 \tag{4.18}$$

Here take editing $f'(\lambda)$ for example. When the spline of $f'(\lambda)$ is edited, the form of $f(\lambda)$ will change and $\Psi(1,1,1)$ will not equal to

[2]This Figure is excerpted from [Ham07] .

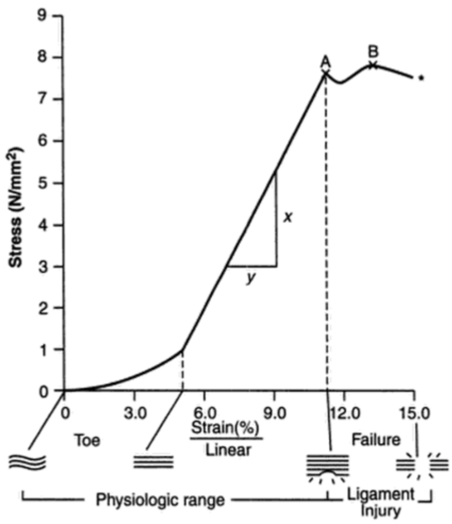

Figure 4.4: Strain and stress curve for connective tissue.

zero in the rest configuration, which will easily crash the simulation due to the accumulation of energy in the rest state. The same is true to g and h term. Non-zero $\Psi(1,1,1)$ will still cause problem for global solved based method (such as Newton Method) when performing the line search. So simple editing of the spline of separate main stretch function will not work. However, although $\Psi(1,1,1)$ is non-zero, it is still the local minimum by keeping $\Psi'(1,1,1) = 0$. When $\Psi'(1,1,1)$ is a strictly monotonically increasing function, then $\Psi(1,1,1)$ will be the global minimum, see Figure 4.5.

Based on this fact, $f'(1), g'(1), h'(1)$ are kept unchanged during the spline editing, which will keep $\Psi'(1,1,1) = 0$ unchanged. $\Psi'(1,1,1) = 0$ does not mean $f'(1), g'(1), h'(1)$ has to be zero. Due to the fact that the increasing of the stress goes with the increasing of the strain (no plastic deformation happens), so $f'(\lambda), g'(\lambda), h'(\lambda)$ should be a strictly increasing function. Let's assume $\Psi(1,1,1) = c$ after editing separate function curves. To keep $\Psi(1,1,1) = 0$, $\Psi(1,1,1)$ can be set as $\Psi(1,1,1) = \Psi(1,1,1) - c$ (see Figure 4.5, the third column). As to

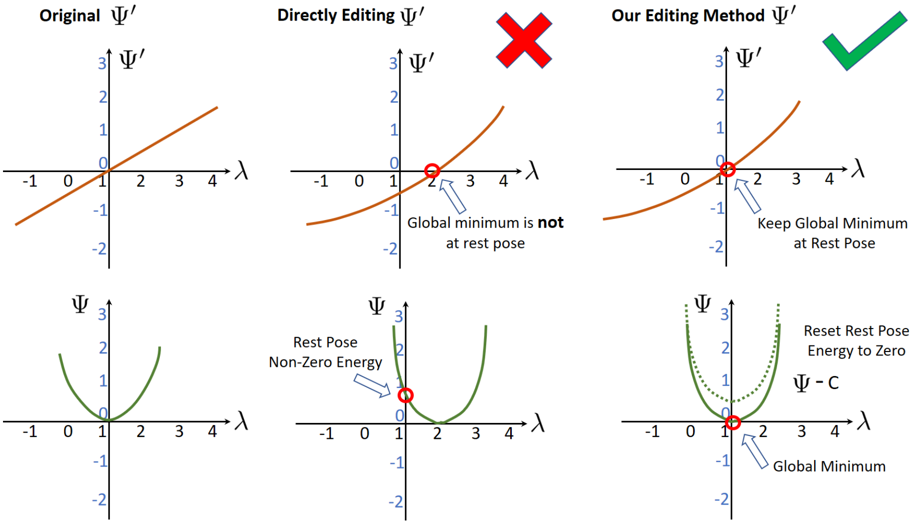

Figure 4.5: Illustration of spline editing method. The energy function is simplified into $\Psi(\lambda)$ form for easy demonstration. The first column is the original form of the function. The second column show that editing $\Psi(\lambda)'$ will influence global minimum value of $\Psi(\lambda)$. The third column keep $\Psi'(1) = 0$ when editing the curve to keep the global minimum value still at the rest position for $\Psi(\lambda)$.

edited separate functions $f(1) = c1, g(1) = c2, h(1) = c3$, they can be updated as $f(1) = f(1) - c1$, $g(1) = g(1) - c2$, $h(1) = h(1) - c3$ to keep the energy in the main stretch direction zero at rest configuration.

When designing the curve, there are many choices for editing. The B-spline interpolation and extrapolation is used. Control points $\mathbf{D}_i$ are manually added to the curve. This system can easily control the soft tissue's respond speed to tension and compression. In the following experiment, the material based on Neo-Hookean model is edited. Under the Valanis-Landel hypothesis, the Neo-Hookean model can be expressed as:

$$f(x) = \frac{\mu_{Láme}}{2}(x^2 - 1) \tag{4.19}$$

$$g(x) = 0, h(x) = -\mu_{Láme} \log x + \frac{\lambda_{Láme}}{2} (\log x)^2 \tag{4.20}$$

where $\mu_{Láme}$ and $\lambda_{Láme}$ are *Láme* coefficients. As can be seen in Figure 4.6, different physical properties are modelled by editing $f'(x)$. In the first row, the material has small resistance to both stretch and compress. In the second row, the material is stiffer than the first row's material in both stretch and compress. In the third row, the material is stiffer than the second row's material when stretched but it has very small resistance to compression. This curve based editing system can also fit the material to experimental data which is generated from uni-axial tension and compression test. Figure 4.7 shows the curve editing system interface in the system. The users can plot on the curve editing panel and insert node to change the shape of the curve.

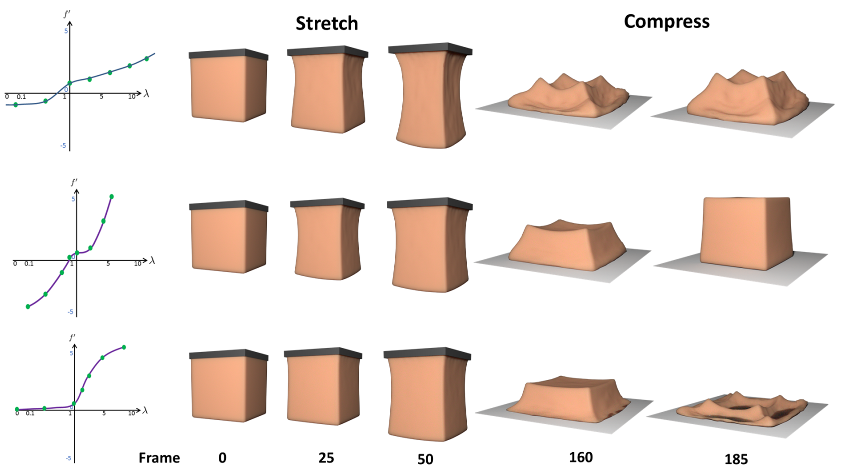

Figure 4.6: Result of different curve representations and physical properties.

4.3.4 Anisotropic Strain Limiting

Strain limiting is necessary for the modeling of soft tissue behaviour. It enables to model high stiffness materials which is compliant to small

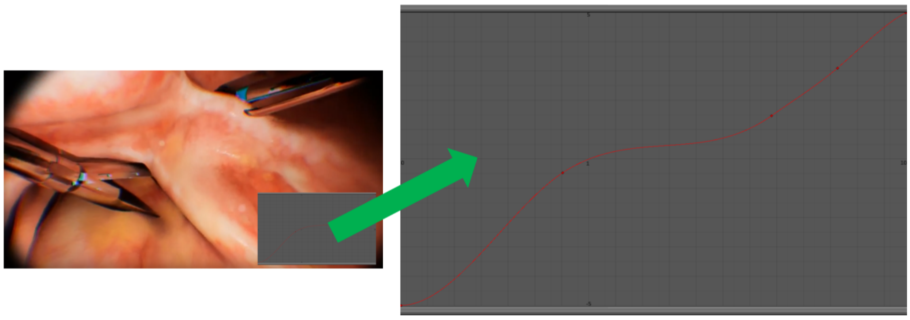

Figure 4.7: Illustration of curve editing interface in laparoscopic surgery simulator.

strain but resistant to large ones. For the soft tissue, the fiber direction will influence the resistance direction to strain.

When applying strain limiting, constraint on each entry of the strain tensor ($\epsilon = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}), \epsilon \in \mathbf{R}^{3\times3}$) can be set:

$$C = \epsilon_{ij} - s_{ij} \tag{4.21}$$

where $i, j = 1, 2, 3$ and s_{ij} is the constraint value in the stretch or shear direction. This constraint can be unilateral or bilateral. For unilateral constraint, $C < 0$ means the material can be stretched but cannot exceed s_{ij}. For bilateral constraint, $C = 0$ means the material is not stretchable.

Let's assume $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3) = \mathbf{D}_s\mathbf{D}_m^{-1} = (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$. For a tetrahedron element $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$, the gradient of the constraint C can be computed as:

$$\nabla C = \nabla\epsilon_{ij} = (\nabla_{\mathbf{x}_1}\epsilon_{ij}, \nabla_{\mathbf{x}_2}\epsilon_{ij}, \nabla_{\mathbf{x}_3}\epsilon_{ij}) = \mathbf{f}_j\mathbf{c}_i^{\mathbf{T}} + \mathbf{f}_i\mathbf{c}_j^{\mathbf{T}}. \tag{4.22}$$

For the fascia, it is composed of triangular element rather than tetrahedron elements. When computing the deformation gradient, $\mathbf{F} = \mathbf{D}_s\mathbf{D}_m^{-1}$ becomes invalide because the inverse of $\mathbf{D}_m \in R^{3\times2}$ does not exist.

To model the strain limiting for fascia and membrane, finding the direction of limiting is important. The texture of fascia and membrane can provide an intuitive and visual reflection to the fibers direction so the strain is limited in the direction of its material coordinate. The strain limiting is performed in the surface parameterization space (UV space), which means using the UV direction to determine the strain limiting direction. This way can provide an intuitive way of controlling the fascia's strain limiting direction because the UV direction can be easily modified in the modern 3D content creation tools (such as Maya, Houdini etc).

For each triangle, a local frame can be computed using the UV coordinate. Let assume the positions of triangle vertices as $(\mathbf{x}_k, \mathbf{x}_j, \mathbf{x}_i)$ and material coordinate as $\mathbf{u}_k, \mathbf{u}_j, \mathbf{u}_i \in \mathbf{R}^2$. For this triangle, a coordinate system can be built using the normal ($\mathbf{N}$), tangent ($\mathbf{T}$) and bitangent ($\mathbf{B}$) as basis. The tangent and bitangent are conforms to the material UV direction (see Figure 4.8), which can be computed as:

$$(\mathbf{T}, \mathbf{B}) = (\mathbf{x}_k - \mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i)(\mathbf{u}_k - \mathbf{u}_i, \mathbf{u}_j - \mathbf{u}_i)^{-1} \tag{4.23}$$

The stretch along the material uv direction can be measured by:

$$\mathbf{D}'_m = (\mathbf{T}, \mathbf{B})^T (\mathbf{x}_k - \mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i) \tag{4.24}$$

where $\mathbf{D}'_m \in \mathbf{R}^{2\times 2}$ is similar to $\mathbf{D}_m$ in the deformation gradient computation (in equation 4.6). The deformation gradient $\mathbf{F} \in \mathbf{R}^2$ of triangle primitive can be computed as:

$$\mathbf{F} = \mathbf{D}_s \mathbf{D}'^{-1}_m \tag{4.25}$$

Then the Green strain can be computed and the strain limiting constraint can be set and solved as equation 4.21 and 4.22.

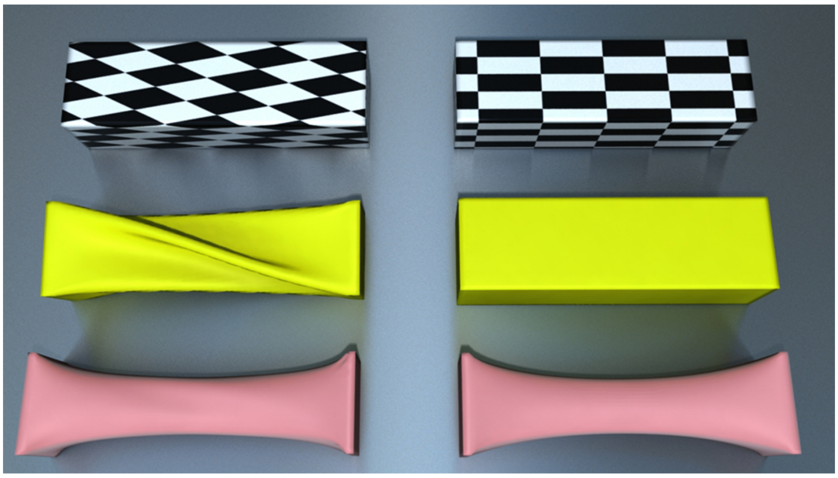

Figure 4.8: Strain limiting comparison (triangular surface model). First row: strain direction illustration (UV direction), second row: strain limiting in u -axis, third row: strain limiting in both u and v axes.

4.3.5 Heterogenous Soft Tissue Modeling

The anatomical structures in abdominal cavity is complex. Each organ is connected with various artery, nerve, ligament, gland etc. For example, in the Figure 4.9, it can be seen that the complex anatomical structure of ano-rectal is connected with hypogastric plexus (superior, inferior), sacrat splanchnic nerves, rectal vein (superior, middle, inferior) etc. The important glands such as urine, sex gland etc are near the rectum which are needed to be protected during surgery.

To apply different stiffness to different parts of the mesh according to the real physical property of the anatomy, painting, as the most widely used user interaction technique in computer animation, is an intuitive and convenient way for achieving this goal. Similar to paint the skin weight for skeleton animation, the idea is to add specified stiffness to the painted vertices of the mesh. However, due to the fact that the anatomical structure is complex and some tissues are embedded inside, the direct painting is not easy to perform. What's more, some

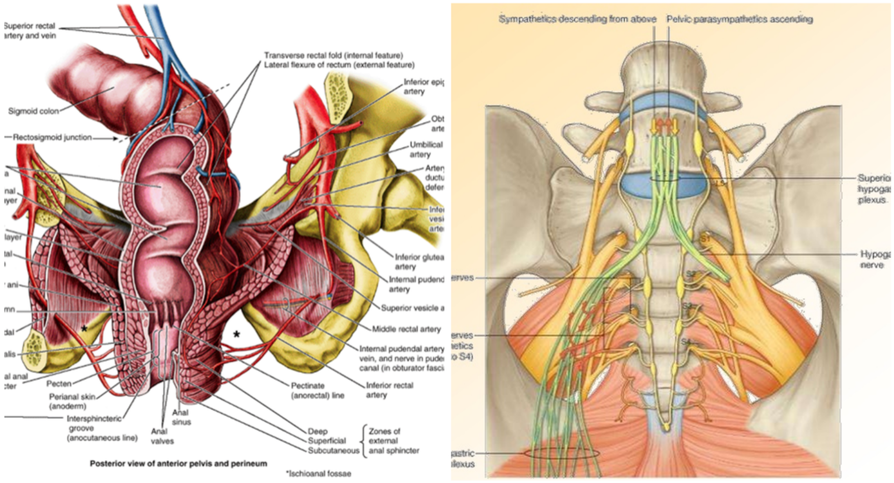

Figure 4.9: Anatomical structure around rectum. The left is the vessel and gland distribution. The right shows the nerve structure map.

soft tissues are tetrahedral mesh so the surface painting will not work because there are vertices inside the boundary surface. Also, if the resolution of the mesh is high, the painting will become a trivial work. Finding a good and compact representation of the mesh will facilitate the painting process efficiently.

As most of the vessels, glands, nerves, organs (such as rectum, colon) are tubiform shape object, the idea is to extract the skeleton of the anatomy model and accelerate the painting procedure using skeleton based paint. Skeleton, as a simple structure, provides a good abstraction to the original shape of the mesh. Medial axis is one of the well known skeleton structure but it is sensitive to surface perturbations and has complex topology. Compare to the complex medial axis method, curve skeleton is a popular skeletal representation in computer graphics and can be a good candidate structure for painting. The curve skeleton is extracted by surface evolution [ATC+08]. The process of surface evolution is contracting the boundary surface of the mesh into a zero-volume degenerated mesh, curve skeleton. Tagliasacchi et al.[TAOZ12] proposed a mean curvature skeleton which improves Au's work [ATC+08] by contracting surface via mean curvature flow and centering the skeleton based on Voronoi poles. The mean curvature flow skeleton can provide a good representation of the anisotropy of the shape and homotopic to the original mesh. Through edge collapse and connectivity surgery, a mapping between skeleton node and surface vertices will be generated.

As the skeleton is a shape abstraction of the original mesh, painting the skeleton and its corresponding vertices instead of the original mesh is a non-trivial but effective way of assigning user-defined stiffness to anatomical mesh. However, curve skeleton only maintains the mapping between skeleton point and boundary surface vertices. Due to the mesh used is tetrahedralized, mapping between skeleton and the vertices which are not on the boundary surface is needed to be built.

For a mesh composed of n points and m skeleton nodes, assemble all the vertices $\mathbf{x}_i \in \mathbf{R}^3$, $i = 0, 1, ..., n-1$ and skeletons $\mathbf{p}_j$, $j = 0, 1, ...m-1$ into the column state vector $\mathbf{x} \in \mathbf{R}^{3n}$ and $\mathbf{p} \in \mathbf{R}^{3m}$ respectively. Let's donate Π_j as the set of points corresponding to the jth skeleton vertex $\mathbf{p}_j$. To ensure all the points in the tetrahedralized mesh has corresponding skeleton nodes, the following method is performed. As can be seen in Figure 4.10, for each skeleton node $\mathbf{p}_j$, the largest distance r_j between $\mathbf{p}_j$ and its corresponding points in Π_j is computed. Then iterate all the vertices in the tetrahedral mesh and measure their distances to $\mathbf{p}_j$. If the distance is less equal than r_j, that point is put into Π_j. Thus, after iterating all the skeleton nodes, all the tetrahedral nodes have their corresponding skeleton nodes.

After finding correspond skeleton nodes for all the tetrahedral nodes, the paint procedure can be started. All the vertices' weights are assem-

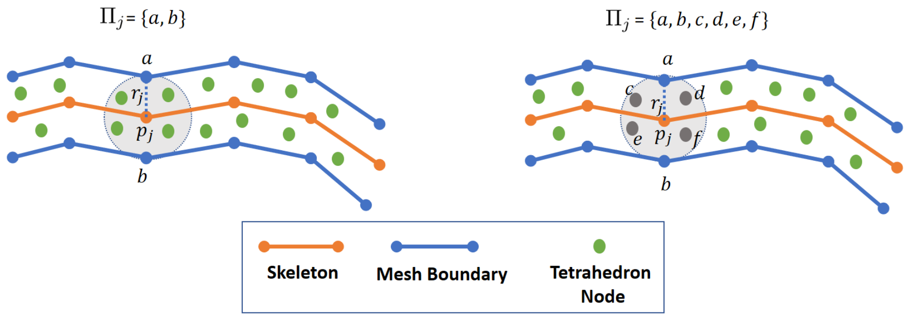

Figure 4.10: Illustration of the skeleton based painting.

bled into a column state vector $\mathbf{W}$. The painting brush is defined as an implicit shape. When the painting procedure start, the state of vertices which fall inside the implicit shape is set as painted. Then a corresponding kernel based weight value is computed (see equation 4.26) and added to those painted vertices. The state vector of weight for the mesh $\mathbf{W} \in \mathbf{R}^n$ can be computed as:

$$\mathbf{W} = \mathbf{W} + \mathbf{W}_{kernel}(\mathbf{c}, \mathbf{v}, h) \tag{4.26}$$

$$\mathbf{W}_{kernel}(\mathbf{c}, \mathbf{v}, h) = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix} \tag{4.27}$$

$$w_i(v_i) = \begin{cases} \dfrac{315}{64\pi h^9}(v_i^2 - h^2)^3, v_i < h \\ 0, v_i \geq h \end{cases} \tag{4.28}$$

where $\mathbf{c}$ is the geometrical center of the paint brush, $\mathbf{v} \in \mathbf{R}^n$ is a vector which measures the distance (v_i) between tetrahedral nodes and $\mathbf{c}$, h is the kernel support radius, such as the *poly*6 kernel in equation 4.28. In Figure 4.11, curve based skeleton painting system and deformed result are demonstrated.

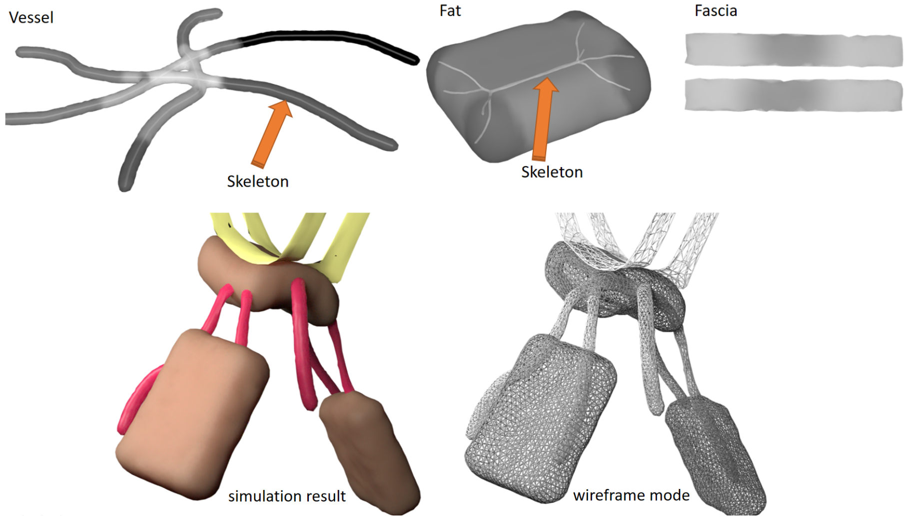

Figure 4.11: Skeleton based painting system. The top row is the skeleton based painting on vessel and fat (tetrahedral mesh). The fasica is painted via polygon paint (triangular mesh). The second row is the combination of those meshes and the simulation result.

4.4 Soft Tissue Simulation

4.4.1 Basics

Newton's second law of motion is the basic of physics simulation. For a simple particle position $\mathbf{x}$, the law can be expressed as $\mathbf{f} = m\ddot{\mathbf{x}}$. This equation can be used to compute the acceleration $\ddot{\mathbf{x}}$ based on the force $\mathbf{f}$ applied on the particle of mass m. The second order ODE can be separated into two first order equation:

$$\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x})/m \tag{4.29}$$

$$\dot{\mathbf{x}} = \mathbf{v} \tag{4.30}$$

Instead of solving the analytical solution, the above equations are normally solved numerically with finite difference.

$$\dot{\mathbf{v}} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^{n}}{h} \tag{4.31}$$

$$\dot{\mathbf{x}} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{h} \tag{4.32}$$

where h is the time step. The way of using $\mathbf{f}(\mathbf{x}^n)$ or $\mathbf{f}(\mathbf{x}^{n+1})$ to represent force will determine the integration type: explicit or implicit. The explicit integration uses current state ($\mathbf{x}^n$) to approximate the force while the implicit integration uses the unknown status in next step ($\mathbf{x}^{n+1}$) to approximate the force which will be linearized as $\mathbf{f}(\mathbf{x}^{n+1}) = \mathbf{f}(\mathbf{x}^n + \Delta\mathbf{x}) = \mathbf{f}(\mathbf{x}^n) + \nabla\mathbf{f}\Delta\mathbf{x}$ during solution. The explicit integration suffers from instability (overshoot) problem under large time step which is not robust enough for developing surgery simulation application. In the following part, only the implicit integration based simulation method is discussed.

Firstly, let's briefly review the implicit integration's variational form. Assuming that a mesh is consisted of n points which are stacked into a position state vector $\mathbf{x} \in \mathbf{R}^{3n\times1}$ and velocity state vector $\mathbf{v} \in \mathbf{R}^{3n\times1}$. The system evolves according to Newton's law through discrete time interval $t_1, t_2, ..., t_n$. $\mathbf{x}^n$ and $\mathbf{v}^n$ represent the position and velocity state at time t_n. The implicit integration can be described as:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1}h \tag{4.33}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + h\mathbf{M}^{-1}(\mathbf{f}_{int}(\mathbf{x}^{n+1}) + \mathbf{f}_{ext}) \tag{4.34}$$

For the simplicity and notion convenience, the external force are ignored when solving equation 4.33 and 4.33. The solution of the implicit integration is:

$$(\mathbf{I} - \Delta t^2\mathbf{M}^{-1}\nabla\mathbf{f})\Delta\mathbf{v} = \Delta t\mathbf{M}^{-1}(\mathbf{f}(\mathbf{x}^n) + h\nabla\mathbf{f}\mathbf{v}^n) \tag{4.35}$$

The global based method solves this implicit integration using the Newton's Method or local/global, or Quasi Newton based method. Details analysis of those method can be found in Appendix 9.1. The drawback of the global based solving is its realtime performance especially when topology changed (see section 2.3 for details). Compare to those global solving based approaches, Position Based Dynamics solves the constraint optimization problem using the iterative method such as Gauss-Seidel and Jacobian, which will produce a visually plausible result rather than a converged (physically accurate) result. The Projective Dynamics is actually a general form of PBD (see Appendix 9.2). The realtime performance, unconditional stability and physical plausible result makes PBD very popular. However, PBD's main drawback is that the stiffness is dependent on the iteration count and time step.

XPBD[MMC16] solves PBD's long-standing problem of iteration count and time step dependent constraint stiffness. For the laparoscopic surgery simulation, mesh topology change is frequent because dissection is the main operation in the surgery. By balancing the realtime performance and accuracy, the XPBD is chosen as the simulation framework.

4.4.2 XPBD Based Simulation Framework

In the XPBD simulation framework, it formulates the energy as

$$\boldsymbol{E}(\mathbf{x}) = \mathbf{\frac{1}{2}C(x)^T}\boldsymbol{\alpha}^{\mathbf{-1}}\mathbf{C(x)} \tag{4.36}$$

where $\mathbf{C}(\mathbf{x})$ is the vector of constraint set $\mathbf{C}(\mathbf{x}) = [\mathbf{C}_1(\mathbf{x}), \mathbf{C}_2(\mathbf{x})..., \mathbf{C}_m(\mathbf{x})]$, α^{-1} is the diagonal compliance matrix which stores the inverse stiffness.

$$\mathbf{f} = -\nabla E(\mathbf{x})^{\mathbf{T}} = \underbrace{-\nabla \mathbf{C}(\mathbf{x})^{\mathbf{T}}}_{direction}\underbrace{\alpha^{-1}\mathbf{C}(\mathbf{x})}_{scalar} \tag{4.37}$$

According to equation 4.33 and 4.34, the following equation can be derived:

$$\mathbf{M}(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}) = h^2\mathbf{f}(\mathbf{x}^{n+1}) \tag{4.38}$$

where the $\tilde{\mathbf{x}} = 2\mathbf{x}^n - \mathbf{x}^{n-1} = \mathbf{x}^n + \mathbf{v}^n h$ represents the inertial of the object. Then introducing the Lagrangian multipler by decomposing the force into its direction and scalar component $\boldsymbol{\lambda} = \tilde{\boldsymbol{\alpha}}^{-1}\mathbf{C}(\mathbf{x})$, where $\tilde{\boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}}{h^2}$. Then the equation 4.38 can be formulated as:

$$\underbrace{\mathbf{M}(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}) - \nabla\mathbf{C}(\mathbf{x}^{n+1})^{\mathbf{T}}\boldsymbol{\lambda}^{n+1}}_{\mathbf{g}(\mathbf{x},\boldsymbol{\lambda})} = \mathbf{0} \tag{4.39}$$

$$\underbrace{\mathbf{C}(\mathbf{x}^{n+1}) + \tilde{\boldsymbol{\alpha}}\boldsymbol{\lambda}^{n+1}}_{\mathbf{h}(\mathbf{x},\boldsymbol{\lambda})} = \mathbf{0} \tag{4.40}$$

Linearizing $\mathbf{g}(\mathbf{x},\boldsymbol{\lambda}), \mathbf{h}(\mathbf{x},\boldsymbol{\lambda})$ can get the following system:

$$\begin{bmatrix} \nabla_{\mathbf{x}}\mathbf{g} & \nabla_{\boldsymbol{\lambda}}\mathbf{g} \\ \nabla_{\mathbf{x}}\mathbf{h} & \nabla_{\boldsymbol{\lambda}}\mathbf{h} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}(\mathbf{x},\boldsymbol{\lambda}) \\ -\mathbf{h}(\mathbf{x},\boldsymbol{\lambda}) \end{bmatrix} \tag{4.41}$$

$\nabla_{\mathbf{x}}\mathbf{g}$ requires the computation of constraint Hessian. XPBD replaces $\nabla_{\mathbf{x}}\mathbf{g}$ with the constant mass matrix $\mathbf{M}$ like a quasi-Newton method. Miles et al.[MMC16] points out that although it will introduce an error on the order of $O(t^2)$ and change the rate of convergence, it will not change the global error. By setting the initial guess $\mathbf{x}_0 = \tilde{\mathbf{x}}, \boldsymbol{\lambda}_0 = 0$. The equation 4.41 will becomes:

$$\begin{bmatrix} \mathbf{M} & \nabla\mathbf{C}^{\mathbf{T}}(\mathbf{x}) \\ \nabla\mathbf{C}(\mathbf{x}) & \tilde{\boldsymbol{\alpha}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{h}(\mathbf{x},\boldsymbol{\lambda}) \end{bmatrix} \tag{4.42}$$

Then the final update becomes:

$$\Delta\boldsymbol{\lambda} = \frac{-\mathbf{C}(\mathbf{x}) - \tilde{\boldsymbol{\alpha}}\boldsymbol{\lambda}}{\nabla\mathbf{C}(\mathbf{x})\mathbf{M}^{-1}\nabla\mathbf{C}^{\mathbf{T}}(\mathbf{x}) + \tilde{\boldsymbol{\alpha}}} \tag{4.43}$$

$$\Delta\mathbf{x} = \mathbf{M}^{-1}\nabla\mathbf{C}^{\mathbf{T}}(\mathbf{x})\Delta\lambda \tag{4.44}$$

When $\tilde{\boldsymbol{\alpha}} = \mathbf{0}$, XPBD becomes the original PBD update rule. Otherwise, $\tilde{\boldsymbol{\alpha}}$ will regularize the constraint force according to the stiffness paramter applied(included in $\tilde{\boldsymbol{\alpha}}$). In the original PBD framework, the stiffness of the constraint influence the update like: $\Delta\mathbf{x}(1-k)^{n_s}$ where

n_s is the current iteration count. The relationship between stiffness and iteration count are nonlinear and hard to tune. The Lagrangian multiplier $\boldsymbol{\lambda}$ reflect the scale of the force which can be used to drive the force dependent devices.

4.4.3 Result and Analysis

In this section, comparison on the performance of different solvers has been made. When solving the dynamic system, the solver can be divided into direct and iterative. The direct solver can provide a robust and accurate solution which is mostly based on the matrix LU decomposition. The iterative solver will approach the solution gradually and convergence rate is dependent on the searching strategy.

As can be seen in Figure 4.12, the performance of direct and iterative solvers are compared. The relative error used in the comparison is defined as:

$$\frac{f(\mathbf{x}_k) - f(\mathbf{x}^*)}{f(\mathbf{x}_0) - f(\mathbf{x}^*)} \tag{4.45}$$

where $\mathbf{x}^*$ is the exact solution, $\mathbf{x}_0$ is the initial guess value and $\mathbf{x}_k$ is the value of k-th iteration. In this simulation test, it can be seen from (a) that Newton's method based on direct and iterative conjugate gradient (cg) solve will converge very fast but they are quite time consuming due to the expensive computation cost in each iteration (see (b)). Liu's method [LBK17] based on prefactorization and direct solve achieves good realtime performance and convergence rate. However, its reliance on prefactorization makes it not suitable for surgery simulation because topology change will happen frequently. The Projective Dynamics also suffers from this problem. When using iterative solving in Liu's method, it can be seen that its convergence rate is affected. For projective dynamics and XPBD, although their convergence rate is quite slow, their realtime performance is their advantage. They can achieve good converged result in realtime because each iteration in XPBD and Projective Dynamics is computationally cheap.

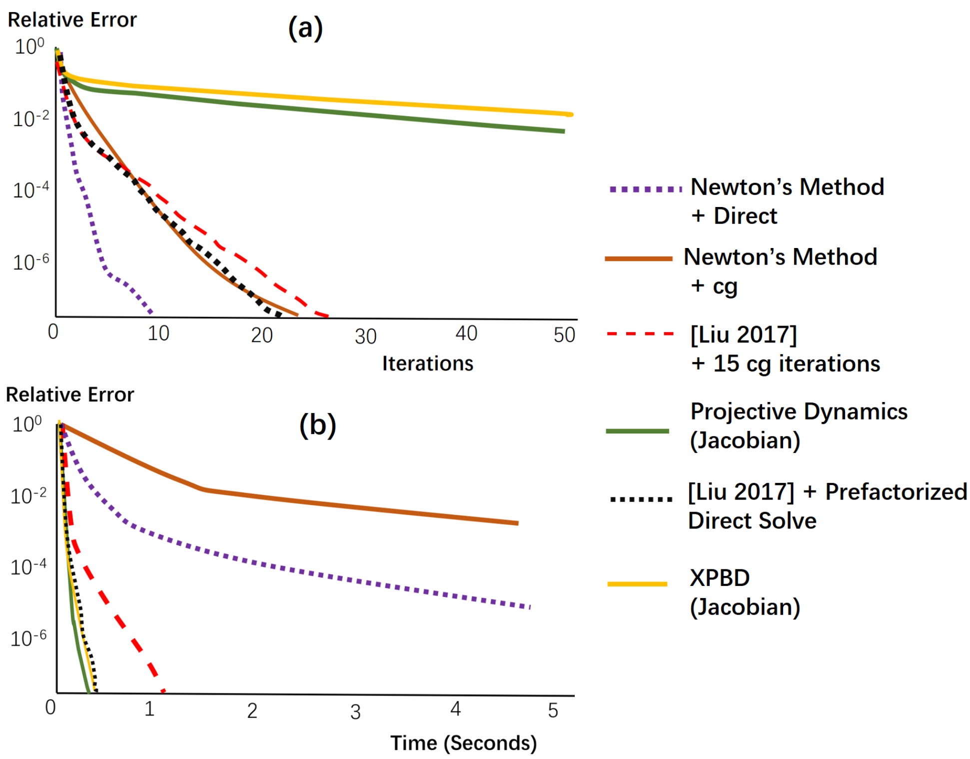

Figure 4.12: Performance comparison between solvers. (a). Relative error for a single frame. (b). Relative error evolves with time.

Although the result of XPBD is not physically accurate in realtime simulation, but it can generate physically plausible behaviour because the system evolves in the tendency of energy reduction (descent gradient direction). As can be seen in Figure 4.13, the liver in Figure is composed of 5350 tetrahedral elements and simulated based on the Neo-hookean material. It can be seen that XPBD (100 iterations) can produce physically plausible results (not converged but has the same tendency as the accurate result) compared to fast converged but time consuming direct solve based method (7 iterations). The relative error of XPBD based simulation is 10^{-4} and the direct solve is 10^{-5}. Although the simulation result of XPBD is not converged, the minor differences do not influence the visual plausibility.

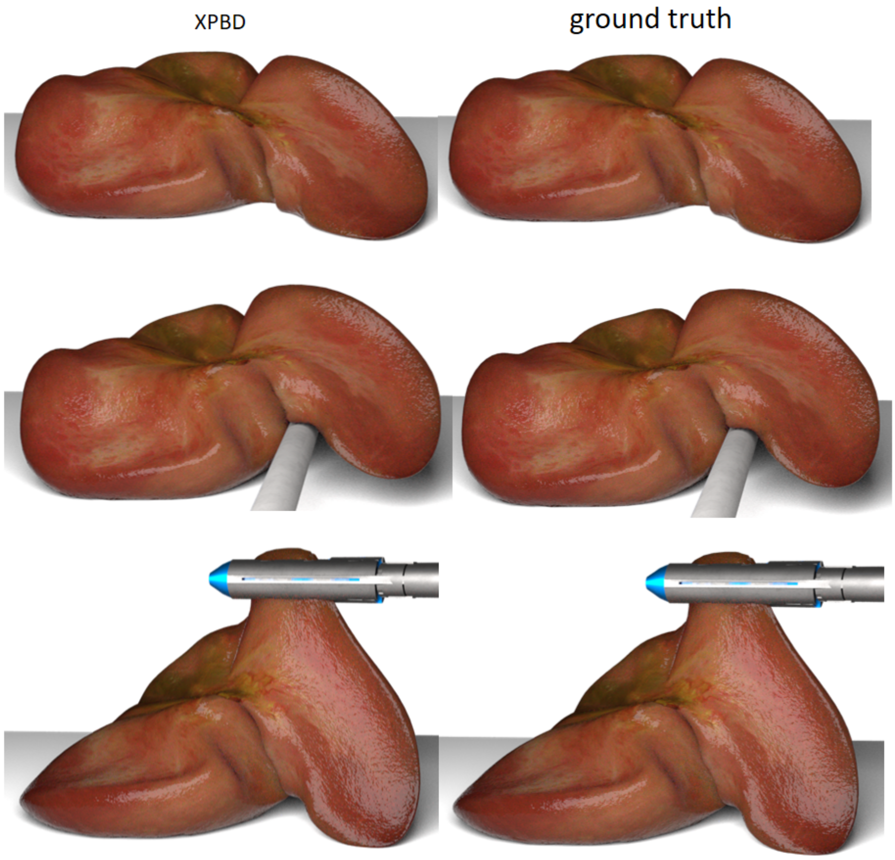

Figure 4.13: Deformation result comparison of XPBD based simulation and accurate direct solve based simulation.

4.5 Summary

In this chapter, a realtime physically based simulation pipeline is presented for multi-layer soft tissue. The pipeline includes multi-layer soft tissue modeling, physics based material editing, skeleton based heterogeneous property painting and realtime simulation. This pipeline is based on a multi-layer (fascia, fat and embedded tissues) soft tissue structure. The strain and stress relationship is modeled based on a curve based hyperelastic material editing system. To achieve heterogeneous material properties of soft tissue, a skeleton based stiffness painting system has been designed. This physically based soft tissue modeling method has been integrated into the XPBD framework to generate realtime physically plausible simulation.

Chapter 5

Surgery Tool Interactions

5.1 Introduction

In laparoscopic surgery simulation, there are multiple types of tool interactions such as sliding contact, poke, clip and dissection etc. To reduce the risk to the patient, mechanical energy-based dissection systems have been incorporated into modern laparoscopic surgery instead of using sharp scalpel. Before the dissection is performed, important blood vessels should be clipped in case of losing too much blood. Those are basic operations in laparoscopic surgery. In a laparoscopic surgery simulator, those operations rely on a robust collision detection system.

In current surgery simulators, frequent tool and soft tissue sliding contact is avoided because it may easily cause collision tunneling artifact when the kinematic movement of the surgical tool is fast. This artifact is caused by the collision detection method used. Considering the realtime performance requirement of surgery simulator, discrete collision detection method is preferred which will perform collision detection in discrete time step. Compare to the accurate but computationally expensive continuous collision detection methods [TTWM14] [WTTM15], the discrete collision detection method is not that robust and may miss potential collision but the low computational cost is its advantage. The main cause of the collision tunneling artifact can be the fast movement of objects or the large simulation timestep. To reduce this artifact, increasing the thickness of the mesh surface using methods such as sweep surface [Eri04] is the most common technique used. However, thick collision margin will cause poor surface approximatione especially for deformable objects which will affect the accuracy of the collision detection and resolution. To get a better approximation collision margin, simplified mesh representation (such as sphere based mesh representation) [MO06] [MO06] [JP04] is often used instead of polygon based representation. However, the good surface approximation provided by the simplified mesh representation will become poor when large deformation happens.

Based on a robust collision detection system, other important operations such as dissection and clipping can be performed. Dissection is the most challenge operation for a surgery simulator because it normally involves the changed of mesh topology which may influence the efficiency and stability of both physics simulation and rendering. In computer graphics, the mesh cutting [WWD15] is an active research field which is a process of finding elements (triangles, tetrahedrons, hexahedrons etc.) that intersected by the cutting surface and then refining the intersected elements to conform to the cutting surface[SHGS06] [BGTG04][SDF07] [WJST14]. However, those refinement based cutting methods will generate neat and regular incisions on the cutting areas which is not the type of cutting in laparoscopic surgery. Rather than using sharp scalpel, blunt and energized dissection tools are used in laparoscopic surgery for dissecting the soft tissue. This kind of dissection is caused by heat and will not generate regular and neat incision which has not been well researched in surgery simulation. Besides the dissection operation, other small but important operations such as clip is also essential for the completeness of the training framework. Minimizing the computational cost of trivial operations without influencing visual fidelity is also an important issue which has not been well researched.

To solve the problems mentioned above, the contributions of this chapter can be summarized as:

- A circumsphere based collision detection and resolution method which outperforms both the existing sphere-based and polygon-based methods in overall performance and reduces collision tunnelling effectively.
- A local geometry feature and energy based adaptivity strategy which dynamically adapt the location and size of the circumspheres surface for better approximation and provide stable and robust collision response.
- A computationally efficient dissection model based on heat transfer which provides physically based modeling for the dissected area. It will not change the topology of the virtual objects and can keep the efficiency and stability of the simulation.
- An efficient geometric based soft tissue clip method which can keep the clip attaching to the vessel and produce visual plausible clip result without introducing the rigid body simulation.

5.2 Adaptive Circumpshere Based Collision Detection and Resolution

Before introducing the detailed method, an overview of the collision detection and resolution pipeline is demonstrated in Figure.5.1. Ac-

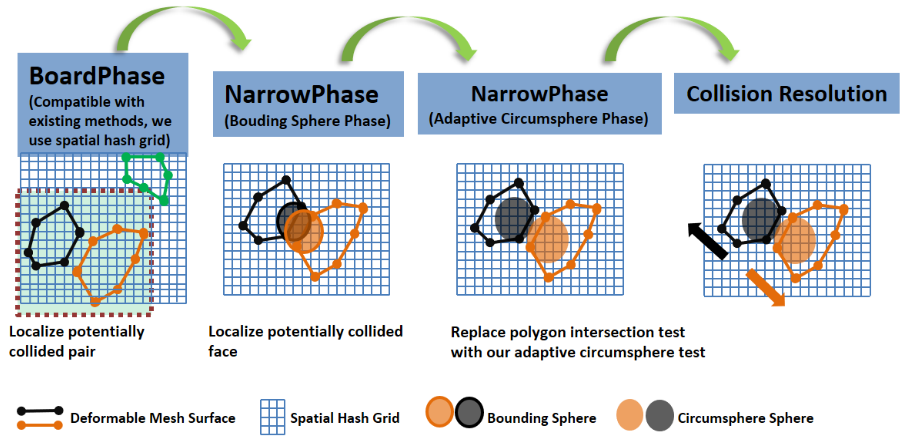

Figure 5.1: 2D Overview of the pipeline of collision detection.

cording to the analysis made in section 2.4.1, the spatial hash structure [THM+03] is used for the broad phase to localize the potentially colliding pairs due to its convenience for parallel computing. The proposed method concentrates on the subsequent narrow phase where it has been separated 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 the overlap of the dynamically generated circumsphere for the underlying primitives is tested and the circumsphere is used for the succeeding collision resolution.

5.2.1 Circumsphere Initialization

Having identified the potentially colliding pairs from the bounding sphere phase, the circumsphere is calculated to perform further intersection test. However, the circumsphere of 3D triangle is not unique. According to the basic geometrical principle, the center of the circumsphere

($\mathbf{C}_{cir}$) can be calculated as:

$$\mathbf{C}_{cir} = \mathbf{x}_c - \mathbf{N}_c\varphi \tag{5.1}$$

where $\mathbf{x}_c$ is the circumcenter of the corresponding triangle primitive which can be found as the intersection of the perpendicular bisectors of each edge, $\mathbf{N}_c$ is the normalized surface normal at $\mathbf{x}_c$ and φ is a global scale factor for translating $\mathbf{x}$ along $-\mathbf{N}_c$. Move $\mathbf{x}$ along the $-\mathbf{N}_c$ is to ensure the dynamically generated circumsphere surface can approximate its underlying triangle surface well. How far to move $\mathbf{x}$ along $-\mathbf{N}_c$ will influence the size of the circumsphere. If uniform scale φ is applied to all circumsphere centers, the circumsphere will not approximate the mesh surface well as shown in Figure 5.3.

To facilitate description of this method, some definitions are made first. As can be seen in Figure 5.2, the triangle split the circumsphere into two sections. The smaller section is called the convering surface while the bigger one is called the uncovering surface. The cone determined by the circumsphere center and the polygon surface called the safe cone. The angle of the cone is called safe angle.

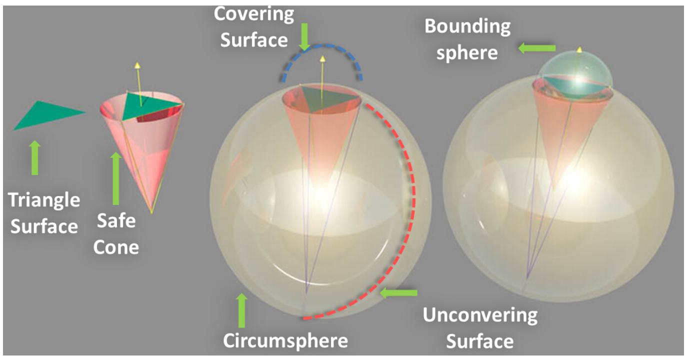

Figure 5.2: Basic notion and structure illustration

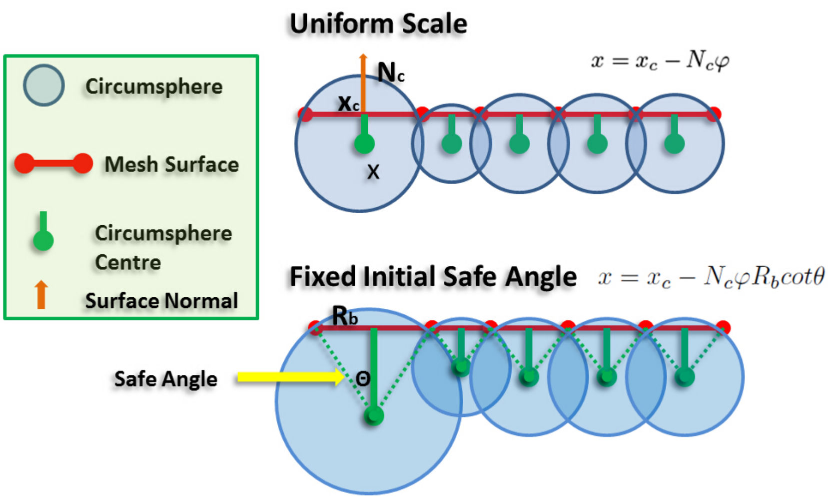

Figure 5.3: Difference between using uniform scale and fixed initial safe angle for circumsphere centre

5.2.2 Local Feature Based Circumsphere

To better approximate the mesh surface, the size of a circumsphere should relate to the geometric feature of its underlying triangle primitive. To make the size of the circumsphere related to the size of its underlying triangle, a fixed initial value θ (see Figure 5.3) is set for the safe angle so that the position of circumsphere center is then influenced by its underlaying triangle size. The approximation based on a fixed initial angle is better than uniform factor φ in equation 5.1.

$$\mathbf{C}_{cir} = \mathbf{x}_c - \mathbf{N}_c \varphi R_b cot\theta \tag{5.2}$$

where R_b is the circumcircle radius of the triangle primitive. In fact, not only the shape of the triangle primitive, but also the local geometry feature of the surface will influence the size of the circumsphere.

Surface curvature is a good reflection of the local geometry feature. Large curvature area can reflect the details of the surface. For those large curvature areas, too large circumsphere will covers too much potential collision areas which will cause the inaccuracy of circumsphere collision detection. Smaller circumsphere can more accurately capture collision and provide more degree of freedom for response direction. The larger the surface curvature is, the smaller the circumphere should be until it shrinks to the bounding sphere. the curvature of circumcenter is used to represent the curvature of the triangle primitive which can be computed via averaging the curvatures of all the vertices in the primitive. In order to apply the influence of local geometry feature to the size of circumsphere, the curvature factor $r = |1/K|$ is integrated into equation 5.1 as:

$$\mathbf{C}_{cir} = \mathbf{x}_c - \mathbf{N}_c \varphi r R_b cot\theta \tag{5.3}$$

However, the curvature K can be zero if surface is a plane. Then the circumsphere will be infinite large because r in equation 5.3 is infinite. In fact, if the surface is a plane, the curvature should have little impact on the circumsphere size which means r should be 1 in equation 5.3. When $K \to \infty$, the circumsphere should become the bounding sphere which can provide good tolerance to artifact and reduce unnecessary collision detection as shown in Figure 5.4.

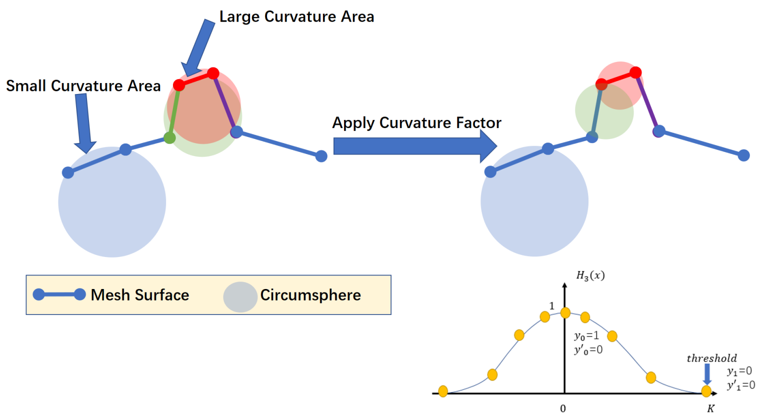

Figure 5.4: Adjust the size of circumsphere according to curvature

To model the relationship between the curvature K and its influence to circumsphere size, the cubic Hermite spline $r = H_3(K)$ is used to control the influence. A threshold h is set for K, when $K > h, H_3(K) = 0$ which means that when the shape is too sharp, the circumsphere becomes the bounding sphere to provide more protection to this area. For arbitrary interval of the spline, the cubic Hermite can be computed as:

$$H_3(x) = y_0 h_0(x) + y_1 h_1(x) + y_0' H_0(x) + y_1' H_1(x) \tag{5.4}$$

where $h_0(x), h_1(x), H_0(x), H_1(x)$ are the Hermite basis functions, y_0 is the starting point at $x = 0$ and y_1 is the ending point at $x = h$ with tangent y_0' and y_1' at $x = 0$ and $x = h$ respectively. As can be seen in Figure 5.4, for $K = 0$, the value of $H_3(K)$ is set as $(K, H_3(K)) = (0, 1), y_0' = \frac{dy}{dx}|_{K=0} = 0$. For $K > h$, $(K, H_3(K)) = (h, 0), y_0' = \frac{dy}{dx}|_{K=h} = 0$. This interpolation function can be calculated at the initial configuration and used directly afterwards. After applying the curvature factor, the centre of the circumsphere can be computed as:

$$\mathbf{C}_{cir} = \mathbf{x}_c - \mathbf{N}_c \varphi R_b cot\theta H_3(K) \tag{5.5}$$

As can be seen in Figure 5.5, after applying the curvature factor, the circumpshere in the high curvature area (red color) shrink more than the circumsphere in the low curvature area. For the high curvature area, protection is more important than good approximation.

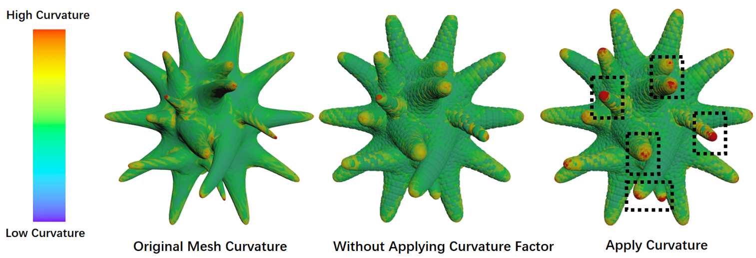

Figure 5.5: Result comparison of applying curvature factor.

5.2.3 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 the collided area to approximate the deformed surface better and provide more accurate collision response. Thus, a physical quantity which used to describe the material property is needed. The deformation can be measured by the amount of energy stored inside the deformed area. The energy accumulated in the deformed object is determined by the deformation gradient which provides a good reflection of how much the object deformed. The deformation gradient can be defined as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \tag{5.6}$$

where $\mathbf{x}$ and $\mathbf{X}$ are the coordinate of vertex in deformed state and rest state respectively. How to define the form of energy is dependent on the constitutive model used. The energy reflects the tendency of

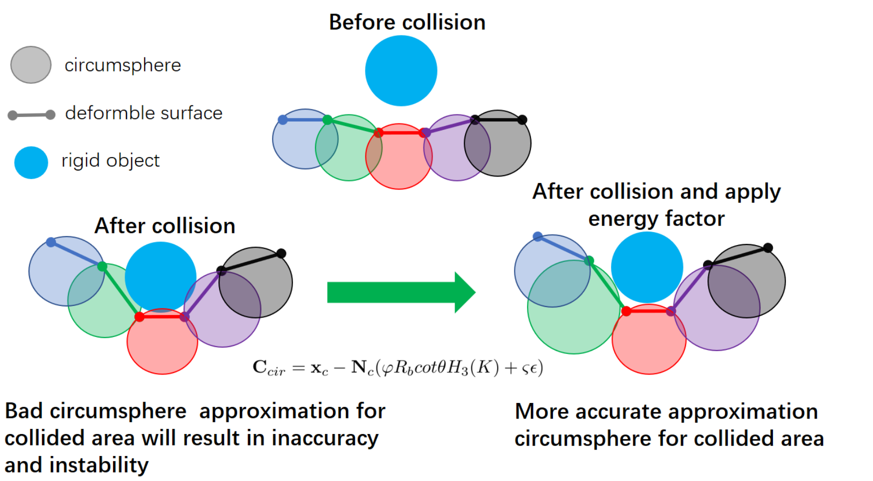

Figure 5.6: Adjust the size of circumsphere according to material property

recovering to initial state. A scalar energy factor $\epsilon = \mathbf{E}$ is used to dynamically adjust the size of circumsphere according to the material property. Theoretically, the energy can be infinite large or small. What is needed is a scalar which can reflect the change of energy rather than the accurate energy. So a user defined scalar factor ς is applied to scale the influence of the energy factor on the position of circumsphere (see Figure 5.6).

$$\mathbf{C}_{cir} = \mathbf{x}_c - \mathbf{N}_c(\varphi R_b cot\theta H_3(K) + \varsigma\epsilon) \tag{5.7}$$

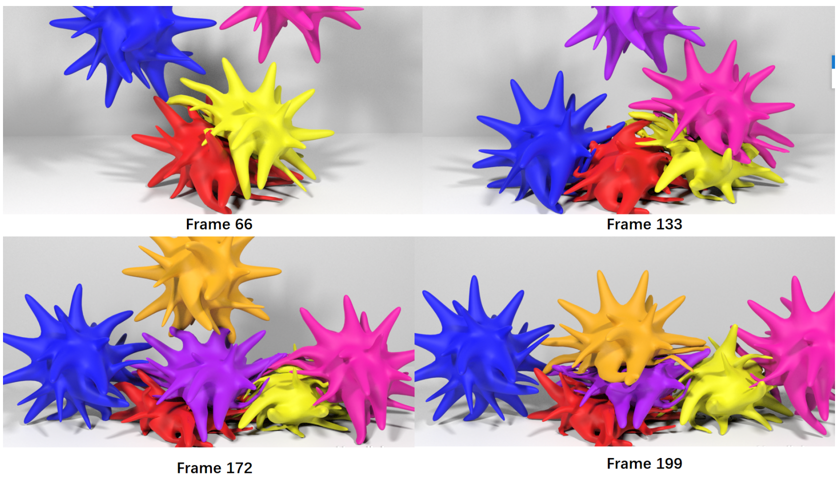

Figure 5.7: The test of the circumsphere based method in complex scene.

As can be seen in Figure 5.7, the experiment is simulated based on FEM method and the models in the scene contains 38910 points and 38989 triangles. It can be seen that even in a complex simulation environment which involves frequent sliding contact, this method can provide good approximation to the shape of the object especially for highly deformed objects, better protection to the high detailed area and generate visual plausible result. Further analysis of this method can be found in section 5.2.6.

5.2.4 Updating of Circumsphere

In order to reduce the computation cost of circumspheres further, circumsphere structure is not updated each frame. The structure is only updated when the shape change of its underlaying primitive exceeds a threshold δ compared to last frame. The change of primitive area is used to measure the rate of shape change $h_{shape} = S_{current} \setminus S_{last}$, where $S_{current}$ and S_{last} are the area of primitive in current frame and last frame respectively. If $|h_{shape} - 1| < \delta$, the centre of corresponding circumsphere is updated using the same curvature factor and energy factor in last frame. When $|h_{shape} - 1| \geq \delta$, the curvature and energy factors are updated. There is no denying that such updating method will affect accuracy, but the tradeoff is the improvement of realtime performance. Such improvement is of great significance to the realtime-oriented application (such as game, surgery simulation etc.). Actually, the inaccuracy has negligible influence on the visual plausibility. Further analysis about the update rate and its impact on accuracy, stability and efficiency is in Section 5.2.6.

5.2.5 Collision Resolution

When collision happened, the collided circumsphere pair and their underlying triangle primitives are bounced away. Firstly, the state which collision just happened is computed. Secondly, update the velocity for each circumsphere using the principle of linear impulse and momentum. Finally, along the direction of new velocity, move each circumsphere to final position. Then update the position of circumsphere's underlying triangle primitive $\mathbf{x}_i$ according to the translation of the circumsphere to the target position $\mathbf{x}'_i$. To integrate the collision resolution into the simulation framework (section 4.4.1), the inequality constraint can be set:

$$C(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \|\mathbf{x}_1 - \mathbf{x}'_1\|^2 + \|\mathbf{x}_2 - \mathbf{x}'_2\|^2 + \|\mathbf{x}_3 - \mathbf{x}'_3\|^2 \tag{5.8}$$

where $\mathbf{x}_i$,$i = 1, 2, 3$ is the vertex position of triangle that has been collided, $\mathbf{x}'_i$,$i = 1, 2, 3$ is the target position. This circumsphere based collision constraint provides a differential representation of the triangle vertices. When solving the constraint, the vertices are treated in an unified manner. Such unified constraint solution will improve the stability and convergence of the simulation. The details can be found in section 5.2.6.

5.2.6 Method Analysis

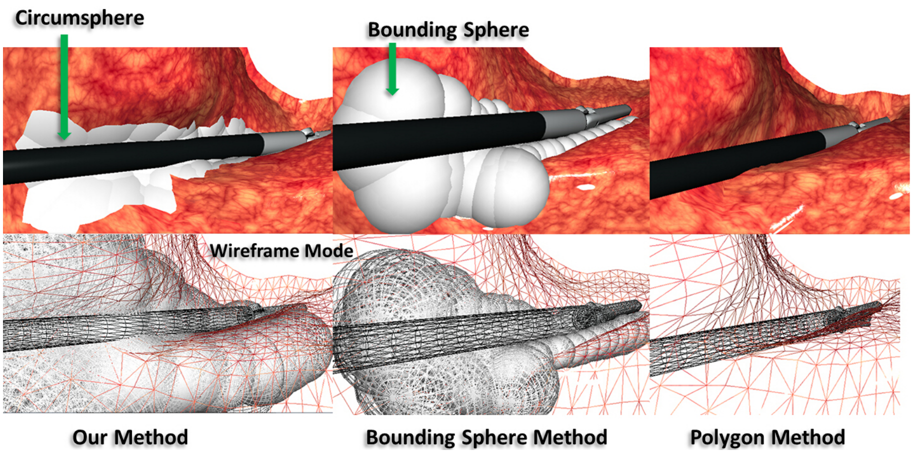

Figure 5.8: Comparison between the circumpshere, bounding sphere and polygon based method.

The polygon based method [Eri04] and bounding sphere based method [MO06] is compared with the method proposed in this thesis from accuracy, stability and efficiency perspectives. In order to compare with the ground truth, a special experiment of collision detection between a rigid cylinder and a deformable soft tissue is devised as can be seen in

Figure 5.8. Thus the exact distance between collided vertices and the surface of cylinder collider should be the cylinder's radius. This exact distance d_{exa} will be compared in the following experiments.

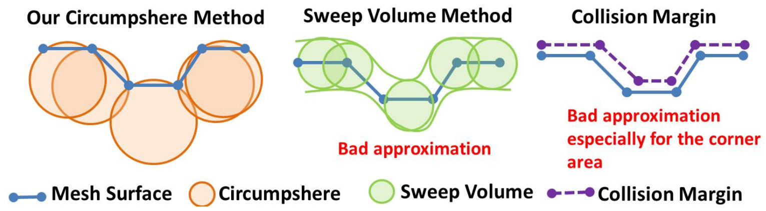

Figure 5.9: Compare proposed method with the collision margin and sweep volume

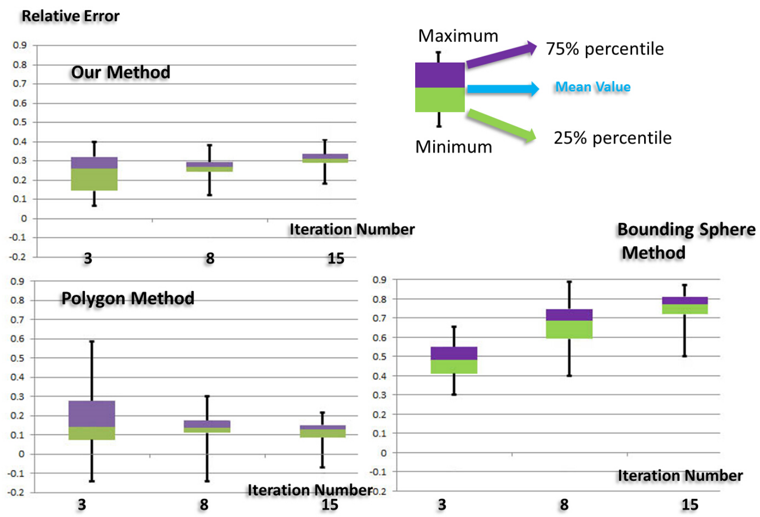

Figure 5.10: Method analysis from accuracy perspective

The average distance from collided vertices to the main axis of the cylinder is used to measure accuracy. As can be seen in Figure 5.10, the box plot graph is used to demonstrate the relative error of average distance. When iteration is low, although the polygon based method is the most accurate (because it has the smallest mean value), the accuracy is achieved at the cost of inducing visual artefacts (see Figure 5.11) which are represented by the negative relative errors. Although such artefacts can be avoided by using collision margin, the collision margin is fixed rather adaptive. It will also incur bad approximation especially for deformable object, as can be seen in Figure 5.9. Bounding Sphere method is the most inaccurate method because its poor approximation of the mesh surface. Such poor approximation is the main cause of its instability. Although the proposed algorithm is not the most accurate method, the adaptive circumsphere structure actually provided certain tolerance to numerical error without causing noticeable artefacts so that the visual plausibility of the proposed method is better than polygon based and bounding sphere based method. Figure 5.11 compares the visual plausibility of different methods.

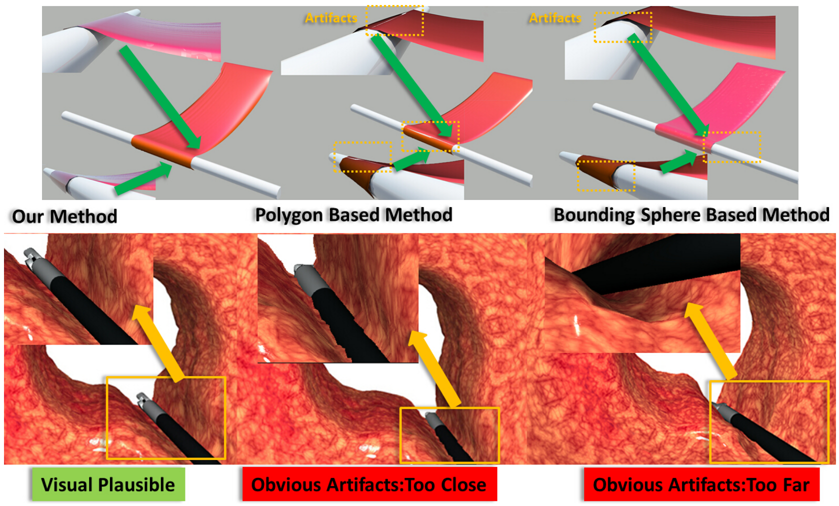

Figure 5.11: Visual plausibility comparision. The visual artefacts are obvious for polygon based method and bounding sphere method

The fluctuation of the vertex position reflects stability. Therefore the average position change is measured for all the collided vertices in each frame to evaluate the stability. As can be seen in Figure 5.12, the polygon and bounding sphere based methods are subject to dramatic fluctuation when the number of iteration is small. For the polygon based method, the high variability of polygon shape is the main cause of the instability. The high variability of polygon shape could easily increase the chance to break the already solved constraint. For the bounding sphere based method, the instability is caused by its poor approximation of mesh surface. The circumphere structure, on the contrary, can not only provides good approximation of the mesh surface, but also adds fewer unstable factors to the constraint solving process due to its uniform shape, making the proposed method the best performer especially when the number of iteration is low. From the mathematics perspective, the circumsphere actually provides a differential representation for its underlying triangle. In traditional collision detection, each vertex, edge and face of a triangle primitive need to be tested for collision. Each type of collision constraint will add unstable factors to the constraint solving process especially under low iteration. The proposed method used the differential representation of the triangle primitive which unify the type of collision test and effectively reduces the unstable factors.

Time consumed by collision detection and resolution in each frame is a good measurement of algorithm efficiency. As can be seen in Figure 5.13, the bounding sphere method is the most efficient method because it only needs to update the centre, radius and perform overlap test when collision happens. The proposed 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 one because polygon intersection test should be performed for each potentially collided pair in each frame.

As to the update threshold δ proposed in Section 5.2.4, accuracy, stability and efficiency are tested for different δ. As can be seen in Figure 5.14, the accuracy is measured by the relative error of d_{ave}, stability is measured by the variance of d_{ave}. When $\delta = 0$, circumsphere will be

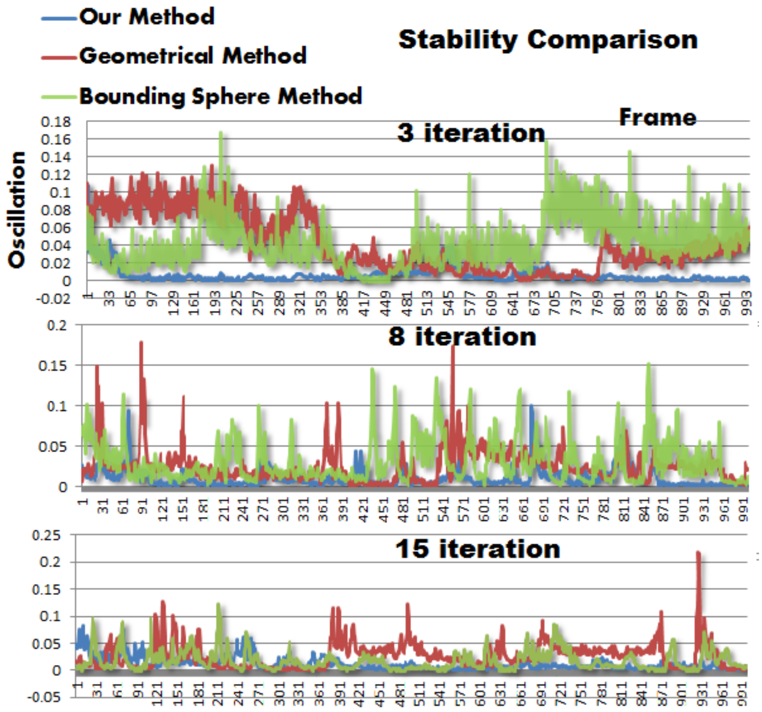

Figure 5.12: Stability comparison under different iteration number

updated in each frame which results in instability and inefficiency. The cause of instability is that the ever changing size of circumsphere constantly add unstable factors to the constraint solving process. And the cause of inefficiency is the re-calculation of circumsphere in each frame. Relative error of accuracy evolves with the increasing of δ because the approximation becomes inaccurate. The larger δ is, the less time is needed for circumsphere updating. The extreme situation is no update for circumsphere. Then the circumsphere's approximation of deformed mesh is the worst which can result in instability.

For different types of simulation purposes, different updating rate should be utilized(as can be seen in Figure 5.15). For efficiency and stability oriented scene (the first row), it involves sliding and detailed collision between various deformable objects and different collision primi-

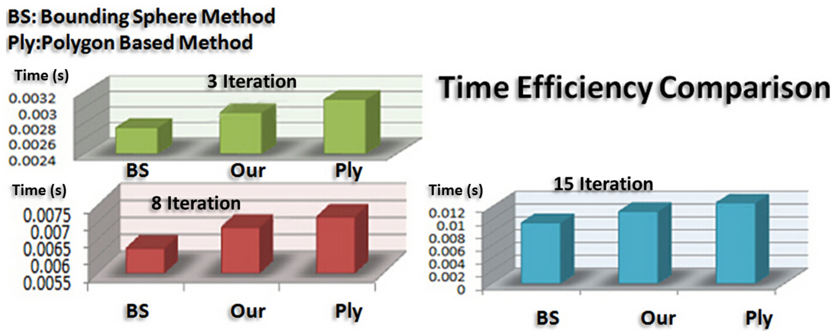

Figure 5.13: Efficiency comparison between the proposed method and others

tives. To maintain efficiency and stability, δ is set to 90%. For accuracy and stability oriented scene (second row), it is a virtual surgery system which contains detailed and sliding collision. δ is set as small as possible within the stable range (30% $\sim$ 200%). For stability, efficiency and accuracy oriented scene, it contains complex sliding collision between deformable objects. To keep stability, accuracy as well as efficiency, $\delta = 70\%$ is used.

5.3 Energized Soft Tissue Dissection

5.3.1 Edge Based Structure

As the material of soft tissue is composed of fat, fascia, lymph, nerve, and so on, some of the dissected area may still connect after dissection because different tissues react differently to high energy (Figure 5.16). Such complex incision generated by energized tools is challenging for real-time simulation.

In this section, an edge-based mesh structure is presented for dissection which is different from the polygon (triangle, tetrahedral, hexahedral etc.) based structure and can generate complex dissection pattern.

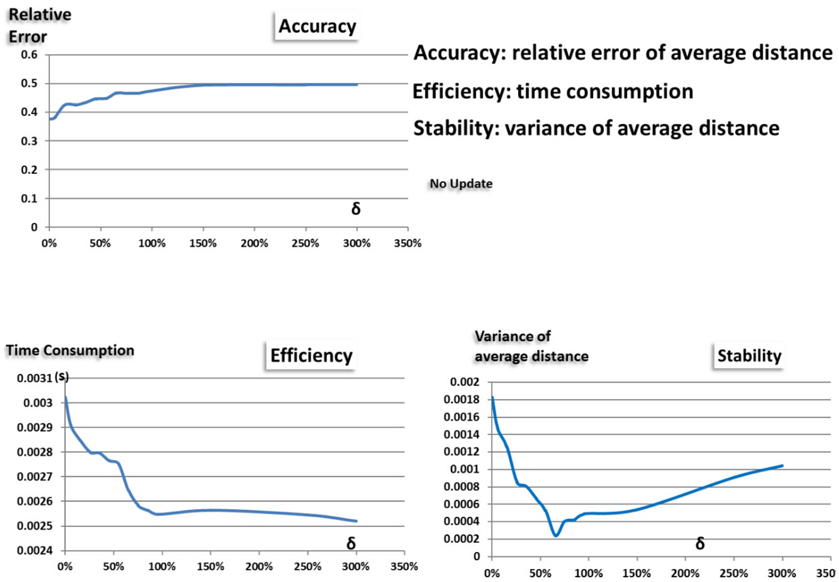

Figure 5.14: Comparison of accuracy, stability and efficiency for different δ

The edge is treated as the most basic element for the mesh. Each edge maintains the information of triangles or tetrahedrons which share it, see Figure 5.17. The proposed method can efficiently incorporate the dissected mesh into the physics computational model and produce more realistic incision pattern which generated by energized tools (see Figure 5.22).

In the initial stage, each polygon primitive keeps the counts it has been shared by the edges, the counter for the k-th polygon primitive is denoted as $share\_count[k]$. When an edge is dissected, the shared count of this edge's surrounding polygons should be subtracted by one. When rendering the mesh, there is no need to reorganize the index buffer and change the size of vertex buffer. In each frame, the rendering buffer is only fed with the polygon faces whose *share_count* remains the same as the initial stage. Further analysis is in section 5.3.4.

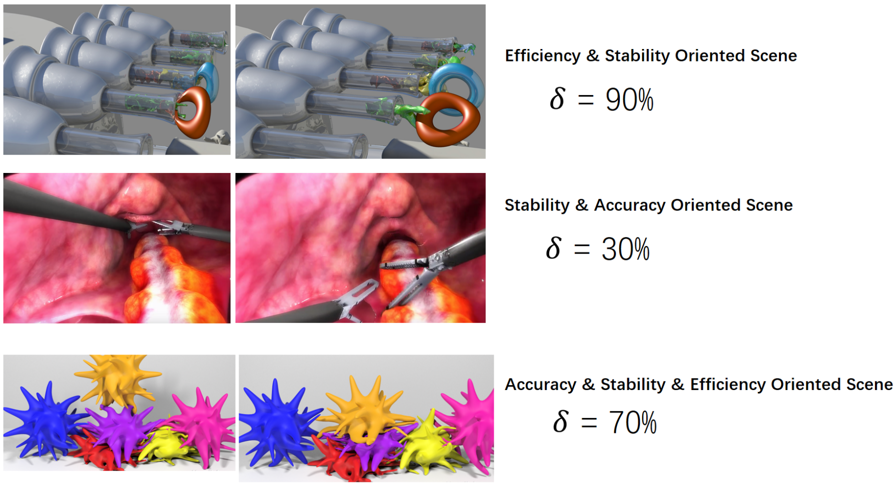

Figure 5.15: Different δ used for different scenes

Figure 5.16: The soft tissue dissection in laparoscopic surgery

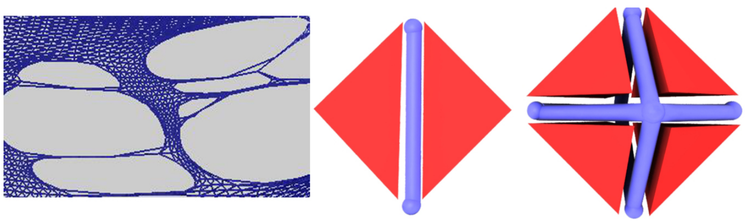

Figure 5.17: Edge based structure

5.3.2 Heat Transfer Model

Due to the high energy in the tool, most of the soft tissue will be dissected immediately when touched, so it is assumed that there is no heat transfer on the soft tissue. The heat transfer is only between soft tissue and energized tools. A local $N \times N \times N$ uniform grid coordinate based on the heat source is built. dl is taken as the length step of the grid coordinate and dt as the time step. The temperature at a grid node $\mathbf{n} = (n_x, n_y, n_z)$ is donated as $u(\mathbf{n})$. According to the parabolic heat equation:

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \tag{5.9}$$

where α is the thermal diffusivity. ∇^2 is the Laplace operator. To solve this parabolic PDE, the finite difference method can be used. At $u(\mathbf{n})$, the $\frac{\partial^2 u}{\partial x^2}$ can be solved as:

$$\frac{u(n_x+1,:,:) + u(n_x-1,:,:) - 2u(n_x,:,:)}{dl^2} \tag{5.10}$$

It is the same for the computation of $\frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial z^2}$. The temperature at the vertex of the mesh can be interpolated based on its surrounding grid node's temperature. A conductivity parameter ξ is used to control how much heat the soft tissue can absorb. Due to the proposed dissection method is based on edge, the temperature (τ_e) of an edge (e) is calculated by using the average temperature at the edge's nodes.

$$\tau_e = (u_a + u_b)\xi \Delta t/2 \tag{5.11}$$

where Δt equals to the sum of the time the edge e overlaps with the implicit shape proxy of the tool, u_a, u_b are the temperature at the edge's nodes. When τ_e excess the ignition temperature δ, the edge will be dissected. For different soft tissue, δ have different values. Following are some reference data for the ignition temperature of soft tissues [KCJG$^+$08] (Figure 5.18).

5.3.3 Dissection Area Modelling

Due to the high energy of the energized tool, the newly generated surfaces of the dissected area will shrink because the fibers and protein

Ignition Temperature (δ)	
Name	Description
peritoneum (fascia)	172°C ± 17 °C
mesentery (membrane)	96.4°C ± 4.1 °C
liver (fatty)	76°C ± 2.9 °C

Figure 5.18: Ignition temperature data for some soft tissues.

inside the tissue is burnt [KCJG+08]. According to the equation 4.6 in Chapter 4, the hyperelastic deformation can be conceived as the process that the deformable object from the $\mathbf{D}_s$ state gradually goes back to $\mathbf{D}_m$ state, which equals to the process that $\mathbf{F}$ gradually recovers to the identity matrix. If the target shape that $\mathbf{F}$ goes back to is changed, the shrink effect can be achieved. The newly exposed surface should

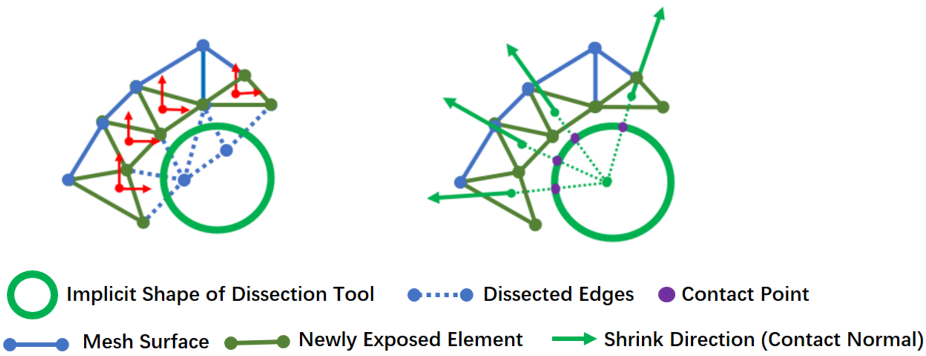

Figure 5.19: Dissected area modeling

shrink perpendicular to the contact surface. An implicit sphere is used to represent the shape of heat energy source which normally locates at the tip of the tool. To get the contact normal, the contact point ($\mathbf{x}_c$) is approximated as the intersection between the implicit shape of the energy source and the lines which determined by the center of the newly exposed element and the center of the energy source, see Figure 5.19.

Assuming the implicit shape function represents the dissection tool is $f(\mathbf{x}) \in R$, the normal ($\mathbf{N}_c$) at the contact point is $\partial f/\partial \mathbf{x}_c$. To build the local coordinate for the newly generated primitive, the direction of $\mathbf{N}_c$ is used as one axis. As to other two axes ($\mathbf{N}_a$, $\mathbf{N}_b$), they are computed by using the face normal ($\mathbf{N}_f$) of the newly generate face and $\mathbf{N}_c$. $\mathbf{N}_a = \mathbf{N}_f \times \mathbf{N}_c$, $\mathbf{N}_b = \mathbf{N}_c \times \mathbf{N}_a$.

To achieve shrinkage on specific direction, as can be seen in Figure 5.20, the deformation gradient $\mathbf{F}$ is decomposed into elastic component $\mathbf{F}_e$ and shrinkage component $\mathbf{F}_s$, $\mathbf{F} = \mathbf{F}_e\mathbf{F}_s$. The shrinkage component can be modeled as:

$$\mathbf{F}_s = \mathbf{R}^T \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \mathbf{R} \tag{5.12}$$

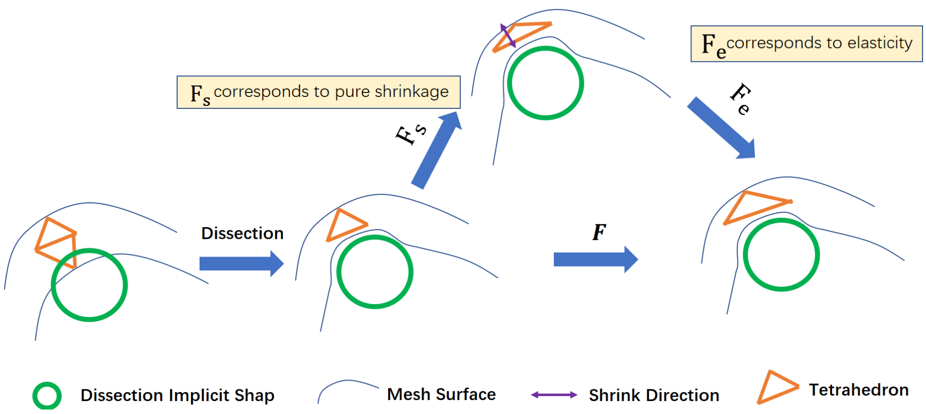

Figure 5.20: Illustration of the deformation gradient decomposition

where $\mathbf{R}$ is the rotation matrix which transforms the initial normal strain direction to align the local coordinate frame of the contact point. S_x, S_y, S_z are the shrink ratio in each direction.

The proposed edge based structure can generate more complex patterns than other polygon based dissection methods (Figure 5.21). When the *share_count* equals to zero which means no primitive elements share this edge, the edge can be rendered. The edge is treated as a elastic spring, it will break when it exceeds the breakage threshold. This complex pattern is important for simulating fascia dissection.

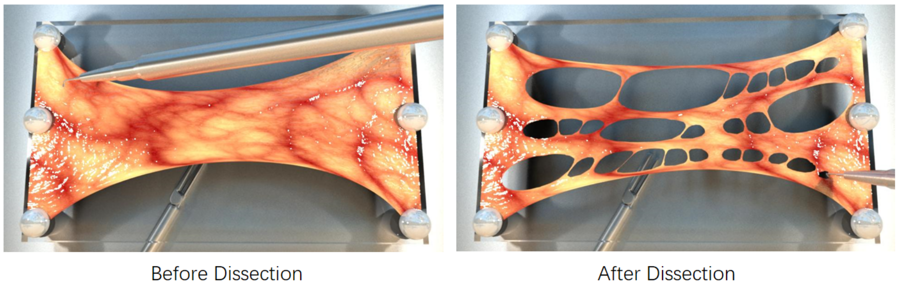

Figure 5.21: Energized tool dissection using the proposed method which can produce complex dissection pattern.

5.3.4 Method Comparison and Results

Before making comparison, the dissection methods are categorized into topology-changed and topology-unchanged. Different from the topology-unchanged methods, the topology-changed methods will create newly generated vertices. The proposed method belongs to the topology-unchanged method.

For the buffer update, the topology-changed methods need to change the vertex and index buffers size, which require frequent inserting and deleting operations.The complexity of inserting is denoted as $O(ins)$ and deleting as $O(del)$. There is no need for the proposed method to update both buffers. The original vertex and index buffers are referenced and the faces whose *share_count* equals to initial configuration are rendered. The complexity of the buffer operation for the proposed method is $O(n)$ because all the faces are needed to be iterated once for each simulation step.

For the newly generated primitive, the topology-changed methods need to perform mesh refinement ($O(re)$) and find the neighbors for the newly generated primitives which includes element deleting ($O(N\_del)$) and inserting ($O(N\_ins)$). The proposed method only needs one iteration of *share_count* to update the neighbour information for vertices and faces ($O(n)$).

For the ill shape handling, the topology-change method needs to concern whether the newly generated polygons are in ill shape, assuming the complexity as $O(ill)$. The proposed method has not change the topology so that ill shape test is not needed.

The overall complexity of the proposed method is $O(n)$ and the topology changed method is $O(ins) + O(del) + O(N\_ins) + O(N\_del) + O(re) + O(ill)$. Comparing with the topology-unchanged methods, the proposed method has the same complexity as others.

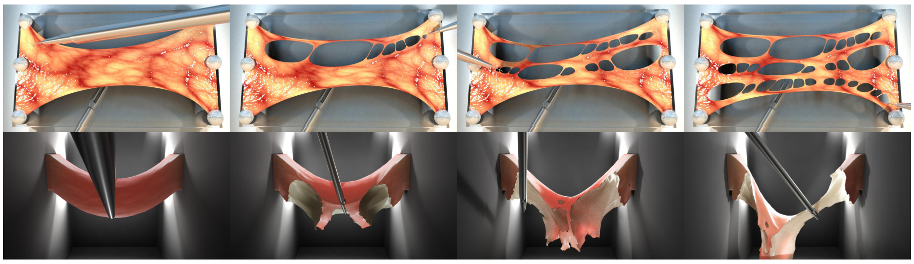

Figure 5.22: Energized tool dissectionof membrane and fat

In Figure 5.22, the fascia dissection is simulated using the following parameters: Energized tool temperature: 170°C. Ignition temperature:160 °C. Shrink ratio:$S_x = 0.1, S_y = 0.3$. Thermal conductivity parameter: 0.4. The fascia mesh is composed of $6K$ triangles, running at 103.2 fps.

For the fatty tissue dissection, the following parameters is used: Energized tool temperature: 90°C. Ignition temperature:78 °C. Shrink ratio:$S_x = 0.1, S_y = 1, S_y = 1$. Thermal conductivity parameter: 0.6.

The fatty tissue is composed of $30K$ tetrahedrons and the membrane is composed of $6K$ triangles. The average fps is 15.1 fps. For the energized tool, the thermal diffusivity used is $4.210^{-6}m^2/s$.

5.4 Geometric Based Tool Clipping

Clipping is an important operation in laparoscopic surgery. Clips are usually clamped onto the blood vessels for the purpose of hematischesis. From the perspective of simulation, clips belong to rigid body. The interaction between blood vessel and clip needs a robust solver which can handle both rigid body and soft body simulation. Adding a rigid body simulation feature into the simulation will increase the computational cost and affecting the stability of the simulation because various types of constraints will make the accuracy of simulation much worse. Contradictions between different constraints set will becomes more obvious especially when low iterations are used for realtime requirement.

Due to the weight of the clip is very small, it can attach to the vessels and follow its movement. Instead of adding rigid body simulation to the system, a geometric based method that can capture the local transformation of the vessel and make the clip follow the local transformation of the vessel has been proposed.

Assuming a vessel mesh is composed of n vertices with deformed position $\mathbf{x}_i$ and rest configuration position $\mathbf{X}_i$, $i = 0, 1, ..., n-1$. When applying the clip to the vessel, the vertices that have been clipped are identified by using the bounding sphere or box of the clipper. So the rigid transformation $\mathbf{A}$ of this clipped area can be approximated by solving the minimization:

$$\min_{A} \sum_{i=s}^{t} w_i \|(\mathbf{A}(\mathbf{X}_i - \mathbf{X}_{cm}) - (\mathbf{x}_i - \mathbf{x}_{cm}))\|_{\mathbf{F}}^2 \tag{5.13}$$

where $\mathbf{x}_i$ and $\mathbf{X}_i$, $i = s, s+1..., t$ are vertices fall inside the clipped region. $\mathbf{x}_{cm}$ and $\mathbf{X}_{cm}$ are deformed and rest state of the geometric center for the clipped vertices, w_i is the weight factor which reflects the contribution of $\mathbf{x}_i$ to the final transformation approximation. This moving least square minimization problem finds a optimial transformation from the rest state to the deformed state. However, if the resolution of the vessel mesh is low, the number of vertices falls inside the clip region could be quite small which can not well capture the transformation of the vessel.

To solve this problem, the concept of virtual auxiliary nodes is introduced. The virtual auxiliary nodes are generated in the initialization stage by adding sampling points into the tetrahedralized vessel mesh. During simulation, every virtual auxiliary node following the movement of tetrahedron that it belongs to via barycentric coordinate. Then the transformation approximation problem can be modified as the minimization of the following equation:

$$\min_{A} \sum_{i=p}^{q} \| w_i (\mathbf{A}(\mathbf{X}'_i - \mathbf{X}'_{cm}) - (\mathbf{x}'_i - \mathbf{x}'_{cm})) \|^2_{\mathbf{F}} \tag{5.14}$$

where $\mathbf{x}'_i$ and $\mathbf{X}'_i$, $i = p, p+1..., q$ are both vertices and virtual auxiliary nodes fall inside the clipped region. The solution of this minimization is:

$$\mathbf{A} = [\sum_{i=p}^{q} w_i (\mathbf{x}'_i - \mathbf{x}'_{cm})(\mathbf{X}'_i - \mathbf{X}'_{cm})^T][\sum_{i=p}^{q} w_i (\mathbf{X}'_i - \mathbf{X}'_{cm})(\mathbf{X}'_i - \mathbf{X}'_{cm})^T] \tag{5.15}$$

The rotation of the transformation $\mathbf{A}$ can be extracted using the polar decomposition as: $\mathbf{A} = \mathbf{RS}$ where $\mathbf{R}$ is the rotation matrix. Then the transformed positions of the clip's vertices ($\mathbf{x}_i^c$) can be approximated as:

$$\mathbf{x}_i^c = R(\mathbf{x}_i^c - \mathbf{x}_{cm}^c) + \mathbf{x}_{cm}^c \tag{5.16}$$

where $\mathbf{x}_{cm}^c$ represents the translation for the clip which can be computed as the geometric center translation of the clipped area $\mathbf{x}_{cm}^c = \mathbf{x}_{cm}' - \mathbf{X}_{cm}'$.

When extracting the rotation from the transformation matrix $\mathbf{A}$, polar decomposition has to be made which is a computational cost operation especially on GPU because it has some code branches which is not computational efficient on GPU. Inspired by the works proposed in [MBCM16] which presents a robust rotation extraction method for deformation simulation, the expensive polar decomposition based rotation extraction is replaced by this rotation updating strategy:

$$\mathbf{R} = \phi(\omega)\mathbf{R} \tag{5.17}$$

where $\phi(\omega)$ is known as the exponential map which is a rotation matrix with axis $\omega/|\omega|$ and angle $|\omega|$. ω can be computed as:

$$\omega = \frac{\sum_i \mathbf{r} \times \mathbf{a}_i}{\|\sum_i \mathbf{r}_i \cdot \mathbf{a}_i\| + \varepsilon} \tag{5.18}$$

where $\mathbf{r}_i$ and $\mathbf{a}_i$, $i = 1, 2, 3$ are the column vectors of $\mathbf{R}$ and $\mathbf{A}$ respectively. ε is the safety parameter ($\varepsilon = 1e^{-9}$ is used). This iterative update of $\mathbf{R}$ is simple and can be implemented in a fews lines of code without branch. This feature is quite suitable for GPU implementation.

For the clipped vessel, the shape of the vessel in the clipped area should be flattened. However, due to the fact that the geometric method is used to make the clip follow the movement of vessel, there is no dynamics relation between the clip and vessel. To generate the flatten effect of the vessel,the idea of visual mesh is applied which is proposed in [MCM16].The vessel is deformed before sending the 3D mesh into the renderer which means only the mesh is deformed for visual effect purpose. The visual mesh does not influence the physical simulation because it is only performed in the rendering stage.

To flatten the clipped region, the proposed method is based on the skeleton computed for stiffness painting (see section 4.3.5). Firstly, the clipped vertices and the region's geometric center are projected to the skeleton curve. Then the displacement for flatten effect is computed based on the distance from the projected vertices ($\mathbf{x}_i^{\mathbf{P}}$) to the projected geometric center ($\mathbf{x}_{cm}^{\mathbf{P}}$) (see Figure 5.23):

$$\mathbf{d}_i = \underbrace{\frac{\eta}{\|\mathbf{x}_i^{\mathbf{P}} - \mathbf{x}_{cm}^{\mathbf{P}}\|_F^{\gamma}}}_{factor} \underbrace{\frac{\mathbf{x}_i - \mathbf{x}_i^{\mathbf{P}}}{\|\mathbf{x}_i - \mathbf{x}_i^{\mathbf{P}}\|}}_{direction} \tag{5.19}$$

where $\mathbf{x}_i^{\mathbf{P}}$ and $\mathbf{x}_{cm}^{\mathbf{P}}$ are the projection of $\mathbf{x}_i$ and $\mathbf{x}_{cm}$ onto the skeleton curve. $\|\mathbf{x}_i^{\mathbf{P}} - \mathbf{x}_{cm}^{\mathbf{P}}\|_F^{\gamma}$ measure the distance between $\mathbf{x}_i^{\mathbf{P}}$ and $\mathbf{x}_{cm}^{\mathbf{P}}$. The above equation is divided into a factor and displacement direction. γ controls the fading speed of the factor and η influence the intensity of the factor. $(\mathbf{x}_i - \mathbf{x}_i^{\mathbf{P}})/\|\mathbf{x}_i - \mathbf{x}_i^{\mathbf{P}}\|_{\mathbf{F}}$ is the normalized direction for flatten. As can be seen in Figure 5.24, different intensity and fading factors can produce different clip effect.:

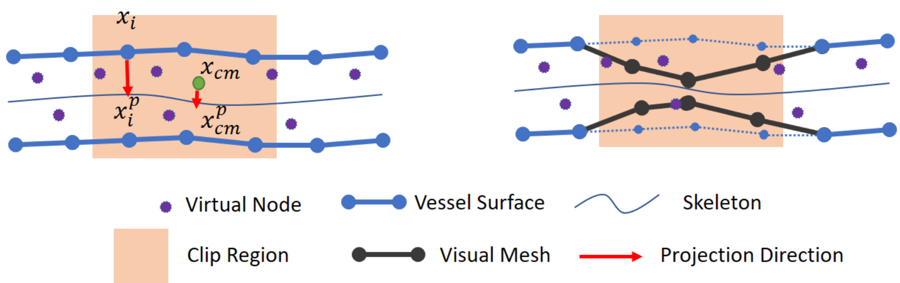

Figure 5.23: Illustration of the geometric based clip method.

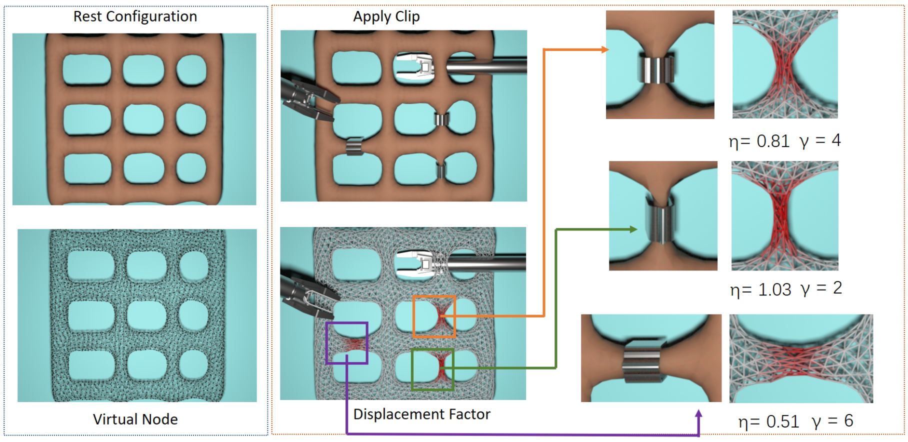

Figure 5.24: Illustration of clip effects under different intensity and fading parameters.

5.5 Summary

In this chapter, a surgery interaction system is presented for laparoscopic surgery simulation, which can provide robust tool collision (especially sliding contact), energized dissection and clipping. The tool collision is based on an implicit circumsphere based collision detection and resolution method for deformable objects which takes into consideration both local geometry features and the material properties. This method is not only computationally efficient, but also stable and comparatively accurate, outperforming the existing methods in overall performance. This implicit circumsphere method can also provide better prevention to collision tunnelling than existing methods. Besides, this method is compatible with all existing broad phase and narrow phase collision query techniques.

Based on the collision detection system, a computationally efficient energized dissection and clipping system is proposed. The energized dissection takes into consideration of the heat factors when performing the soft tissue dissection. A physically based modeling method is proposed for the dissected area based on the energized dissection. The clip system is based on a computationally efficient geometric method for clip to follow the motion of clipped object.

Chapter 6

Realistic Soft Tissue Rendering

6.1 Introduction

In laparoscopic surgery, the appearances of soft tissues are mostly composed of complex organic patterns. As the position of the camera is very close to the surrounding anatomies, it poses high requirement on the quality of the soft tissue's texture. Low quality texture asset will directly influence the realism of final rendering. In the traditional texture generation pipeline, the digital artists firstly search and collect references of specific organic tissues from video or images for the preparation of digital production, which normally takes quite a long time and the quality of existing references can not be guaranteed. Those collected references mostly can not be used directly because it contains glinting reflex and superficial wrinkling on top of the tissue's surface which caused by the lighting environment. So it is difficult to obtain a nice and clean base texture for the 3D anatomy model. Besides the based color texture, there is not any effective way to obtain the material textures such as specular, roughness. The rendering of soft tissue still highly dependent on the artist skills.

Soft tissue is different from the objects rendered in the digital entertainment applications which are most single layer. Soft tissue is a complex composite of different materials. Skin rendering is the most relevant technique to soft tissue rendering in computer graphics research but the material properties of soft tissue is not totally the same as the skin rendering. They may share some resemblances such as the subsurface scattering effect but the layers of soft tissue exhibit obvious different properties due to its special organic patterns. For surgery simulation, there is not any specially designed rendering pipelines yet for realistic soft tissue rendering. Most of the simulators render the soft tissue using simplified rendering pipeline from game industries which may easily cause unrealistic plastic appearance. In this thesis, the following contributions are made:

- An efficient procedural organic material generation pipeline has been proposed. It enables the users to efficiently generate organic look texture based on the real image.
- A multi-layer soft tissue rendering pipeline which includes multi-layer soft tissue shading, optimized lighting effect modeling and post-processing.

6.2 Procedrual Muti-Layer Soft Tissue Material Generation

6.2.1 Quadtree Based Procedural Texture Generation

Most of the internal tissues are wrapped by membrane and composed of muscles and fat. The membrane consists of multiple layers of epithelial cells overlying the loose connective tissue layer. The organic patterns of those tissue are reflected by the distribution of the fat and muscle. To get a well description of soft tissue pattern, MRI scan can provide a good reference. The component of soft tissue will display different colors in the MRI image. Analyzing the soft tissue pattern based on the MRI data has been widely used in diagnosis of soft-tissue masses such as tumor [BR09][GLL+17]. The fat will be bright white while the muscle will be darker. As can be seen in Figure 6.1, it is the CT data of splenic tumor with a gradual centripetal fill-in organic pattern and its corresponding organic look in the real image. The organic pattern will determine the appearance of the soft tissue. The realism of the rendering of soft tissue is dependent on the quality of the organic pattern.

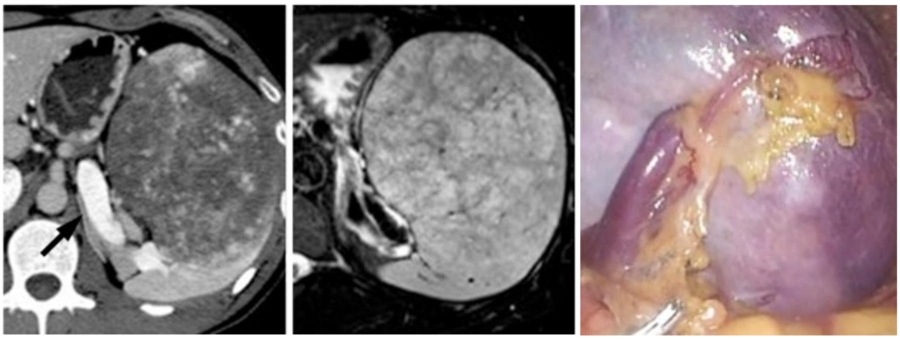

Figure 6.1: The Abdominal CT which shows an enhanced splenic tumor with a gradual centripetal fill-in pattern [KKY+14].

How the organic pattern can influence the rendering of the soft tissue is important for the soft tissue rendering. From the computer graphics perspective, the organic pattern can be used as a mask for the painting or filling for materials such as base color, roughness, specular etc. The challenge is how to find a efficient way of generating such mask.

Procedural texture generation provides a wide range of possibilities for generating organic patterns. A procedural texture is a computer generated image which is created using an algorithm to represent a natural-looking element, such as organic, wood, textile, metal, stone etc. Noise (randomness) control is the central of the procedural content generation process. An effective way of controlling noise distribution can facilitate artists creating desired effect easily.

In the procedural content generation (PCG) research, the algorithmic content generation can be classified into learning based, search based and tiling based methods [SO14]. The learning based method is dependent on the existing data to generate similar content but the existing data (organic pattern) is hard to obtain. The search based method generates contents by searching a defined space with evaluation function but the evaluation of soft tissue pattern is hard to define. So the tiling based method is chosen which generates contents from the randomization and combination of small tiles. This method is more intuitive and easy to control for artists.

The key challenge of the tiling based method is how to organize the randomization (such as rotation, opacity, size, offset etc) of tiles (or patterns) efficiently. In image segmentation, quad-tree split is a widely used technique for the representation of image. Each tree node represent a level of subdivision. The children nodes represent the uniformly subdivided regions of current node's image region, as can be seen in Figure 6.2. Each tree node can have its own pattern content. The core idea of the quadtree procedural texture generation process is that subdivide the input image, apply randomness to each level's node and then blend the result (see Figure 6.3). The blend mode can also be varied, such as add, subtract, max, min etc.

After introducing the basic idea of the quadtree based procedural texture generation pipeline, the procedure of generating organic looking

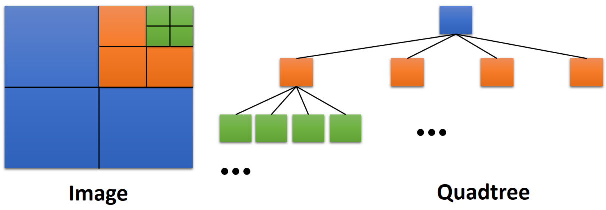

Figure 6.2: Illusion of quadtree image split

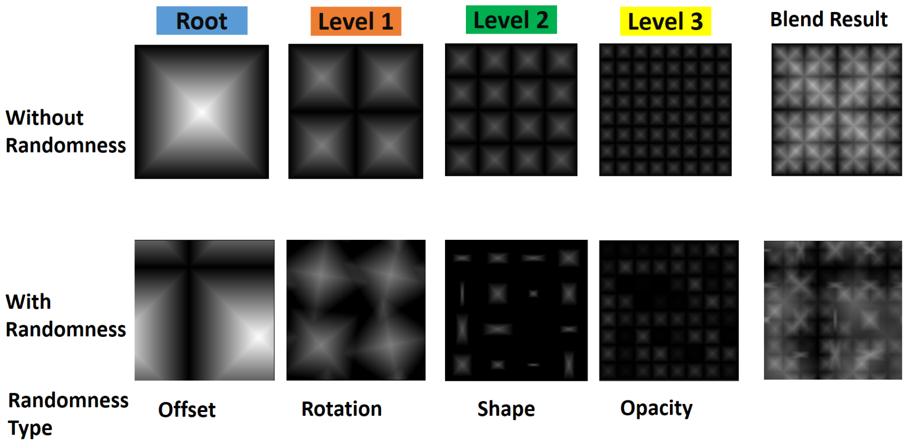

Figure 6.3: Illusion of procedural texture generation process. The first row shows each level's pattern and blend result. The second row shows each level's pattern with randomness and the blend result.

pattern will be introduced. In Figure 6.4, a 7 level quadtree is created with patterns on the last two levels. As can be shown in Figure 6.1, the centripetal fill-in pattern requires the randomness of the pattern should include the rotation, offset, size, opacity etc. For the pattern offset (grey node), the polar coordinate is used to control the rotation of the offset, which can add sparsity to the final looking. For the size of the pattern (blue node), the width and height will influence the sharpness of area. The larger they are, the smoother the result is. For the rotation of the pattern (orange node), it affects the natural looking of the connectivities of the patterns. As shown in Figure 6.4, two examples are demonstrated.

The difference between two examples is the level that apply the pattern (green node). The earlier the pattern is applied, the larger the organic pattern is.

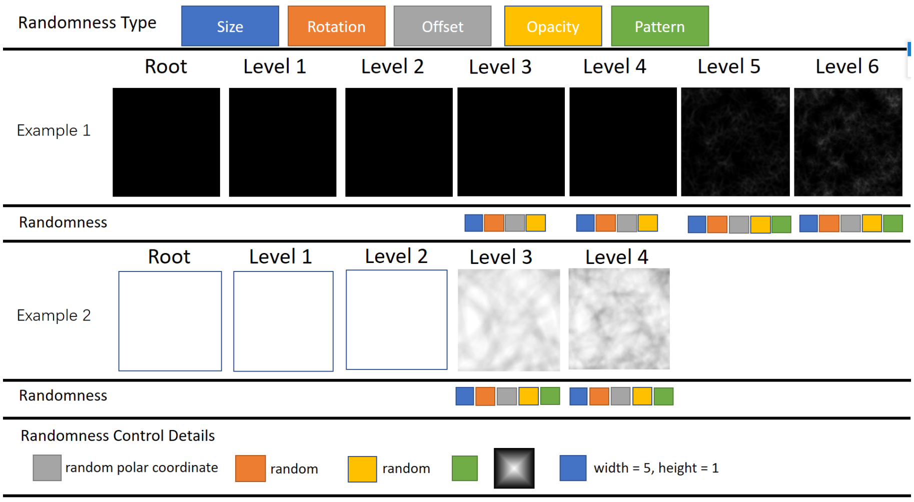

Figure 6.4: Generation process of the organic pattern mask texture

6.2.2 Material Generation Pipeline

Following is the procedure of generating the organic looking material texture asset. The rendering pipeline is based on the state-of-art rendering technique Physically Based Rendering (PBR) (see section 6.3). The basic PBR pipeline needs several material textures such as base (diffuse map), roughness (surface smoothness map), metallic, height (displacement map), normal etc.

The whole procedure is dependent on the procedural organic pattern which is generated in previous section. For diffuse map generation, this gray scale organic pattern is used as mask for color blending. Which color is used to blend will determine the basic appearance of the tissue. The color can be chosen from photo or video references by extracting the color palette and prominent color proportional map as can be seen in

Figure 6.5. The color palette is used for artists picking color for textures and the prominent proportional map gives a overview of prominent colors in the image which is useful for creating environment map or lighting map.

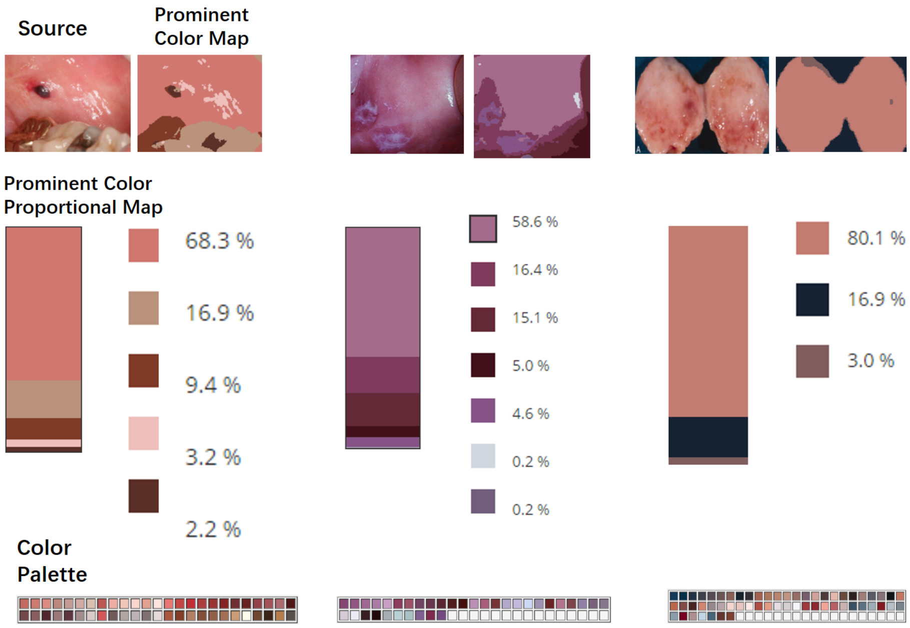

Figure 6.5: Illustration of color palette and prominent color proportional map

The color palette is obtained by iterating every pixel and measuring the similarity between two pixels. The measurement is performed in component-wise order. For example, for color $\mathbf{C}_1, \mathbf{C}_2$, the distance between the red component is $dist(\mathbf{C}_1, \mathbf{C}_2) = |\mathbf{C}_1.r - \mathbf{C}_2.r|$. If result of all the components measurements less than user specific threshold, two pixels are marked as similar. The color palette is obtained by averaging each group of similar colors. For the prominent color proportional map, it is obtained by larger threshold for measurement.

For the roughness, metallic, displacement, normal map can all be generated using the same idea. The example of the generation of organic looking texture procedure can be seen in Figure 6.6 which is only for demonstration purpose rather than specific soft tissue material. Each layer of color can be blended based on the mask of organic pattern. The mask can also be post processed in the image space (such as blur, level, normal etc) for different materials.

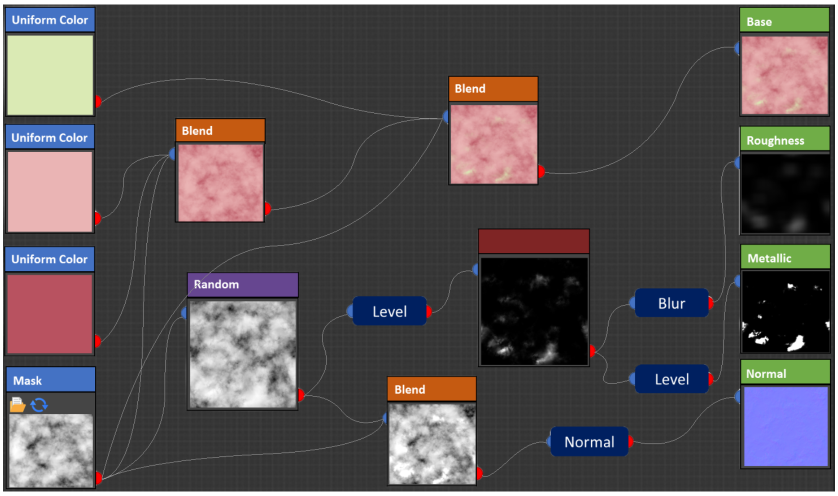

Figure 6.6: Illustration of organic texture generation process.

6.2.3 Multi-Layer Soft Tissue Rendering

The mucous membrane is mostly of endodermal origin and is continuous with the skin at various body openings such as the inside mouth, lip, eyes etc. Mouth is the first portion of the alimentary canal. The soft tissue appearance inside the mouth bears great resemblance to the internal soft tissues in laparoscopic surgery, as can be seen in Figure 6.7[1]. High quality references of oral mucous are much easier to find due to it is at body opening position which is easy for recording by high resolution device.

According to the structure of oral mucosa, the materials are divided

[1]http://www.studiodentaire.com/conditions/fr/cancers-bouche.php

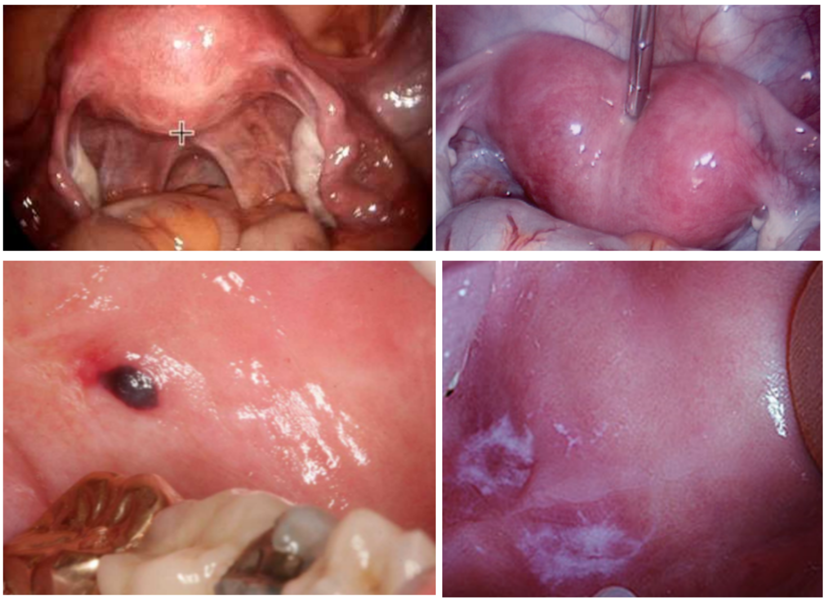

Figure 6.7: Soft tissue appearance in laparoscopic surgery (first row) and mouth (second row).

into four layers as can be seen in Figure 6.8. The first layer is the surface layer which is used to simulate the color of the epithelium. The granular layer is used to show the granular organic pattern formulated by the granular cell. The connective tissue layer formulate the connective fiber pattern of the tissue. The submucosa layer is used to represent the last material layer which usually contains the blood vessels, nerves and lymphatic vessels.

The rendering is based on the Physically Based Rendering technique (see section 6.3 for more details). As can be seen in Figure 6.9, the render pass include the direct light diffuse, subsurface scattering the direct specular, granular specular and reflection. The direct light diffuse determine the based color of the soft tissue. The subsurface scatter is used to simulate the effect that the light pass though the soft tissue. The specular (direct and granular) component can show the surface

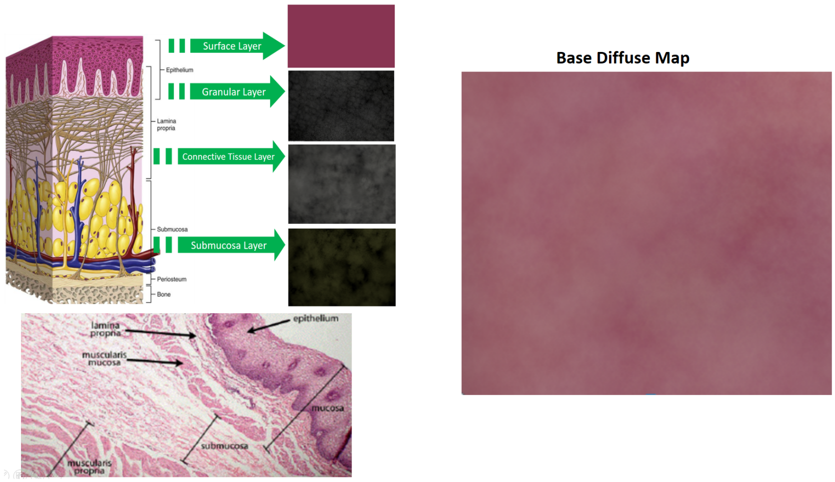

Figure 6.8: Structure of oral mucosa

roughness of the soft tissue. The reflection is used to reflect the mucus effect on top of the soft tissue.

In Figure 6.10, the proposed rendering result is compared with the real soft tissue image in mouth. Also, the soft tissue rendering in other simulators is compared. There are many causes for the unrealistic rendering effect in current surgery simulators, such as low quality reference image, incorrect roughness, single layer material and inaccurate rendering method.

6.3 Physically Based Rendering

Physics based rendering (PBR) is a popular rendering technique in game and visual effect industries. PBR physically models the flow of light in real world, making PBR rendering pipeline look correct regardless of lighting condition which is not true in non-PBR rendering pipeline. The PBR rendering is based on the theory of microfacet model which describes object surface from microscopic scale. It assumes the object's surface is composed of very small reflectors which has different physi-

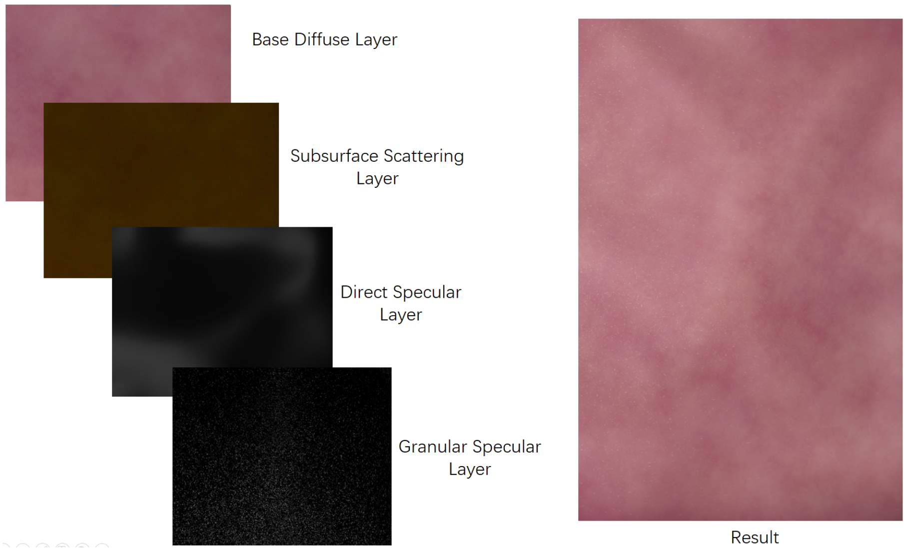

Figure 6.9: The breakdown of the render pass.

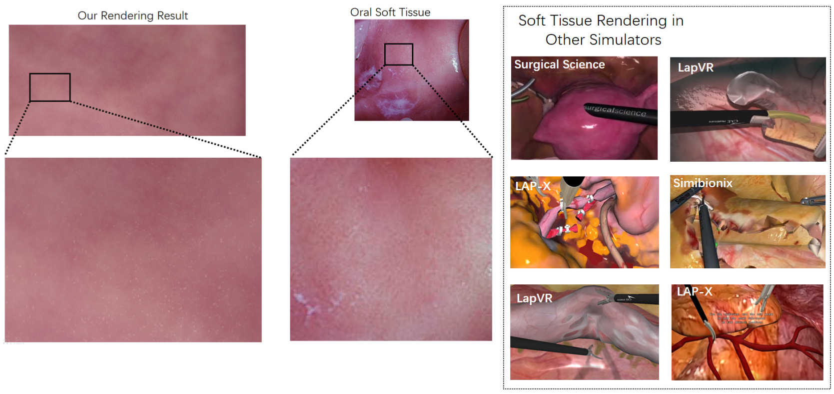

Figure 6.10: Texture and rendering quality comparison with soft tissue image in mouth and existing simulators.

cally properties such as roughness, metallic etc. The empirical model (Blinn, Phong, etc) is a fast computational model adjustable by parameters, but without considering the physics behind it. The microfacet model is inspired by real physical process.

The microfacet theory follows the energy conservation rule that the energy of outgoing light should not exceed the energy of the incoming light. When the light hit the object's surface, it separates into reflection part (specular lighting) and refraction part (diffuse lighting).

For the modeling of the reflection light, PBR is based on the reflectance equation:

$$L_0(\mathbf{p}, \omega_0) = \int_\Omega f_r(\mathbf{p}, \omega, \omega_0) L_i(\mathbf{p}, \omega_i) \mathbf{n} \cdot \omega_i d\omega_i \tag{6.1}$$

where $L_i(\mathbf{p}, \omega_i)$ is the radiance of point $\mathbf{p}$ from the incoming small solid angle ω_i (see Figure 6.11). The dot product $\mathbf{n} \cdot \omega_i$ scales the energy based on the incident angle to the microsurface around $\mathbf{p}$. $L_0(\mathbf{p}, \omega_0)$ calculates the sum of reflected radiance of a point $\mathbf{p}$ over the hemisphere area Ω and returns the amount of reflection in the direction of ω_0. f_r is the bidirectional reflective distribution function (BRDF) which weights how much the individual incoming light from ω_i contribute to the final reflect radiance based on the object's material.

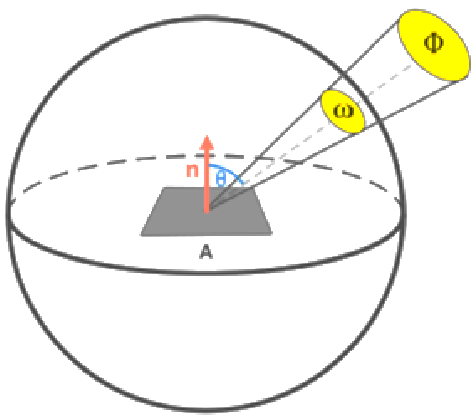

Figure 6.11: Physically based rendering shading model.

For the BRDF term $f_r(\mathbf{p}, \omega, \omega_0)$, it approximates the reflective and refractive amount on the microfacet area. Cook-Torrance is the most widely used model in PBR pipeline:

$$f_r = k_d f_{lambert} + k_s f_{cook-torrance} \tag{6.2}$$

where k_d is the ratio of diffuse light part and k_s is the specular part. $f_{lambert}$ is the lambert diffuse term and $f_{cook-torrance}$ is the specular term. The specular term $f_{cook-torrance}$ is determined by normal distribution function, geometry function and Fresnel equation:

$$f_{cook-torrance} = \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)} \tag{6.3}$$

Where D is normal distribution function describe how microfacets align with half vector. G reflect the self-shadow of the microfacets. F shows the reflection varies over the viewing angle at a surface. The rendering equation can be formulated as:

$$L_0(p, \omega_0) = \int_\Omega (k_d \frac{c}{\pi} + k_s \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)}) L_i(p, \omega_i) \mathbf{n} \cdot \omega_i d\omega_i \tag{6.4}$$

6.3.1 Subsurface Scattering Rendering

As described in equation 6.4, not all lights reflect from a surface. Some lights will penetrate into the object, scatter inside and finally go outside of the surface, which becomes visible to eyes as diffuse or subsurface scatter color.

Subsurface scattering needs to know how far the light travel through the object which normally is dependent on the path tracing. However, path tracing needs to know the thickness of the objects. In realtime rendering, the thickness of objects can be approximated by rendering the object's front and back surface each time to get the depth of the front and back of the object. However, this method will not well capture the depth when the shape is complex. Depth peeling is a strategy to solve this problem but it is a time consuming operation because of the need of multiple render passes for each layer. Instead of using realtime thickness computation method, surface thickness is approximated using the pre-computed thickness map. The thickness map is generated by casting rays in the opposite direction of the surface normal. The darker area means the thin part of the model. Although the thickness map does not reflect how long the light travel through the object, it can provide a good approximation of the object thickness. When light hit the surface, the distance that the light travel through the object can be approximated as a function of the thickness.

For the modeling of the scattering, the diffuse term in equation 6.2 is decomposed into diffuse and subsurface term.

$$f_r = k_d((1 - k_{sss})f'_{lambert} + k_{sss}f_{sub}) + k_s f_{cook-torrance} \tag{6.5}$$

where $f_{lambert}$ now becomes $f_{lambert} = (1 - k_{sss})f'_{lambert} + k_{sss}f_{sub}$, k_{sss} is the weight of the subsurface scattering and $f'_{lambert}$ is the new lambert diffuse term. Then the PBR render equation becomes:

$$L_0(p, \omega_0) = \int_\Omega (k_d(1-k_{sss})\frac{c'}{\pi}+k_d k_{sss} f_{sub}+k_s \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)})L_i(p, \omega_i)\mathbf{n}\cdot\omega_i d\omega_i \tag{6.6}$$

The computation of the subsurface term can be performed as:

$$L_0(p, \omega_0)_{sub} = \int_\Omega k_d k_{sss} f_{sub} L_i(p, \omega_i)\mathbf{n} \cdot \omega_i d\omega_i \tag{6.7}$$

where the f_{sub} is defined as:

$$f_{sub} = Ce^{-\sigma T} \tag{6.8}$$

f_{sub} is a light absorption model which depends on the thickness (T) of the model and absorption coefficient σ. The equation 6.7 now becomes:

$$L_0(p, \omega_0)_{subsurface} = \int_\Omega k_d k_{sss} Ce^{-\sigma T} L_i(p, \omega_i)\mathbf{n} \cdot \omega_i d\omega_i \tag{6.9}$$

Due to $Ce^{-\sigma T}$ will not be influenced by $d\omega$ so it can be taken out from the integration:

$$L_0(p, \omega_0)_{sub} = Ce^{-\sigma T} \int_\Omega k_d k_{sss} L_i(p, \omega_i)\mathbf{n} \cdot \omega_i d\omega_i \tag{6.10}$$

where $L_i(\mathbf{p}, \omega_i)$ is the radiance of point $\mathbf{p}$ from the incoming small solid angle ω_i. To compute this equation, the lighting model is important. The detail lighting model will be introduced in the following section.

6.3.2 Light Modeling

In laparoscopic surgery, the light is a source with area instead of a ideal point source. Different from point light, the solid angle differential $d\omega_i$ is related with the differential area of light dA. The relation between the differential $d\omega_i$ and dA can be formulated as:

$$\frac{1}{r^2} = \frac{d\omega_i}{dAcos\theta_i} \tag{6.11}$$

where r is the distance between the shading point p and the point p' on dA, θ_i is the angle between pp' and the normal of dA. Then equation 6.10 becomes:

$$L_0(p, \omega_0)_{sub} = Ce^{-\sigma T} \int_A k_d k_{sss} L_i(p, \omega_i) cos\theta_i \frac{\mathbf{n} \cdot \omega_i dA}{r^2} \tag{6.12}$$

When assuming the light source is uniform which means the radiance $L_i(\mathbf{p}, \omega_0)$ is constant (L) on A. The above equation becomes:

$$L_0(p, \omega_0)_{sub} = Ce^{-\sigma T} k_d k_{sss} L \int_A \frac{cos\theta_i \mathbf{n} \cdot \omega_i dA}{r^2} \tag{6.13}$$

The term $cos\theta_i \mathbf{n} \cdot \omega_i / r^2$ is bounded by the shape of the area light. Approximate this integration using importance sampling is widely used in game industry. The importance sampling will find most representative sample to simplify the integration. However, this method always has the risk of overestimation. Not like games which always have multiple light sources, there is only one light source in laparoscopic surgery and the accuracy of the lighting effect can greatly influence the realism. Comparing to the procedure of finding most important sample point in the importance sampling, using Monte Carlo sampling based method is feasible on modern graphic hardware especially when the sampling number is small.

Assuming there are m sampling points on the area light, then the equation 6.13 becomes:

$$L_0(p, \omega_0)_{sub} = Ce^{-\sigma T} k_d k_{sss} L \sum_{i=1}^{m} \frac{cos\theta_i \mathbf{n} \cdot \omega_i \alpha}{r^2} \tag{6.14}$$

where α is the area of the dA. When the sampling points number increase, the efficiency will be affected significantly. The sampling number is adaptively changed according to the distance between the lit point and the center of the light source. Then the final lighting equation for subsurface scattering becomes:

$$L_0(p, \omega_0)_{sub} = Ce^{-\sigma T} k_d k_{sss} L \sum_{i=1}^{m(d)} \frac{cos\theta_i \mathbf{n} \cdot \omega_i \alpha}{r^2} \tag{6.15}$$

$$m(d) = \begin{cases} a_1, h_2 < d \leq h_1 \\ a_2, h_3 < d \leq h_2 \\ \cdots, \cdots \end{cases} \tag{6.16}$$

where $m(d)$ is a piecewise function. This adaptive sampling method can ensure high accuracy for objects near the light source. For the objects that far from the light, compromise on accuracy for efficiency is made by adaptively reducing the sampling number. As can be seen in Figure 6.12 (light source is on the right side), the first row is the ground truth effect of the area light. If only use 1x sample on the area light, it becomes a point light and generates hard shading effect across the dark and bright area. When increasing the sampling rate, it can be seen that the hard boundary becomes blurry. The difference between point light and area light effect is easy to be detected near the light (right side) but it is hard to be perceived in distant area (left side). The adaptive

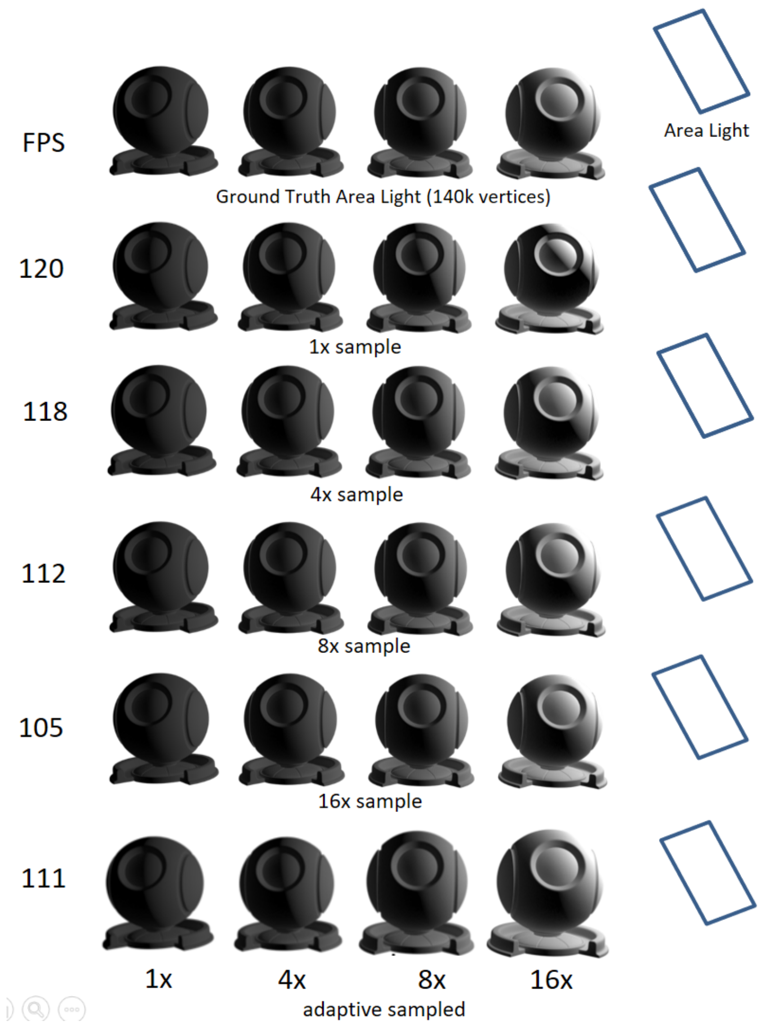

Figure 6.12: Result comparison of different sampling rate and adaptive samples.

sampling strategy can preserve the rendering quality in important area while not adding too much computational burden to processors. Besides the distance dependent $m(d)$ definition, the sampling number for the area light can also be determined by the painting. Then the $m(d)$ is defined as the group of points with the same color. The painting based sampling can produce high rendering quality for important area which is used in the following section part.

For the computation of the diffuse term, it is similar to the subsurface term. The only difference is that it does not need to compute the subsurface scattering term:

$$L_0(p, \omega_0)_{diffuse} = k_d(1 - k_{sss})L \sum_{i=1}^{m(d)} \frac{cos\theta_i \mathbf{n} \cdot \omega_i \alpha}{r^2} \tag{6.17}$$

For the computation of specular term, the same idea is applied:

$$L_0(p, \omega_0)_{spec} = \int_{\Omega} (k_s \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)}) L(p, \omega_i) \mathbf{n} \cdot \omega_i d\omega_i \tag{6.18}$$

$$= L \int_{A} (k_s \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)}) \mathbf{n} \cdot \omega_i \frac{cos\theta_i dA}{r^2} \tag{6.19}$$

$$= L \sum_{i=1}^{m(d)} (k_s \frac{DFG}{4(\omega_0 \cdot n)(\omega_i \cdot n)}) \mathbf{n} \cdot \omega_i \frac{cos\theta_i dA}{r^2} \tag{6.20}$$

Different from the diffuse and subsurface term, the specular term needs to compute D, F, G which is more computational expensive. To further improve the accuracy, only the subset of current sampling points which contribute most to the specular are sampled. To choose the subset of the sampling points, a cone around the reflection vector is defined, as can be seen in Figure 6.13.

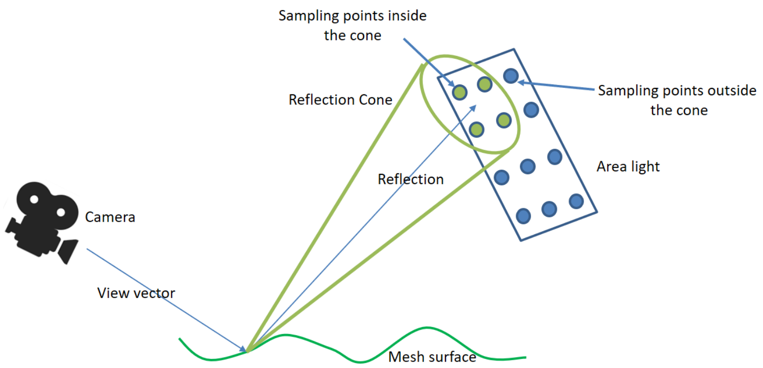

Figure 6.13: Reflection cone for choosing sampling point subset.

The intersection between the cone and the area light plane is computed. Only the points that fall into the cone region are sampled. The final equation for the specular term is:

$$L_0(p,\omega_0)_{spec} = L\sum_{i=1}^{m(d)_{cone}}(k_s\frac{DFG}{4(\omega_0\cdot n)(\omega_i\cdot n)})\mathbf{n}\cdot\omega_i\frac{cos\theta_i dA}{r^2} \tag{6.21}$$

where $m(d)_{cone}$ is the points that fall inside the intersection between reflection cone and area light region. The final PBR based rendering equation is:

$$L_0(p,\omega_0) = L_0(p,\omega_0)_{diffuse} + L_0(p,\omega_0)_{sub} + L_0(p,\omega_0)_{spec} \tag{6.22}$$

A female anatomy model locating at the rectum is used for the rendering experiment of the above PBR rendering equation. A rectangular area light is used for this experiment. The rendering passes and final rendering effect is shown in Figure 6.14[2]. Compared with the ground truth image , there is still minor visual differences but this is caused by the effect of the recording device. This visual different can be further minimized after applying post-processing which can be found in section 6.3.3. As can be seen in Figure 6.15, the user specified important sampling area is painted with blue color (8x sample), compare with the lowest sampling rate in the green color (1x sample), the hard boundary between the bright and dark area becomes blurry in the high sampling area. For the specular effect, the specular area should reflect the shape of the light. It can be seen that the higher sampling rate can generate more accurate specular (rectangular shape) than low sampling rate. In the red color area, most of it is in the dark area and no specular effect in there, lower sampling rate 4x is used to produce soft darkness area. For the green area, the model detail complexity is simple so there is not too much bright and dark variation. 1x sampling is used in this area. The lowest sampling effect can be achieved at around 117 fps while the adaptive sampling result is achieved at 110 fps for a mesh with 20k vertices which is acceptable for realtime application.

[2]https://www.youtube.com/watch?v=HOOCvFMMCqg

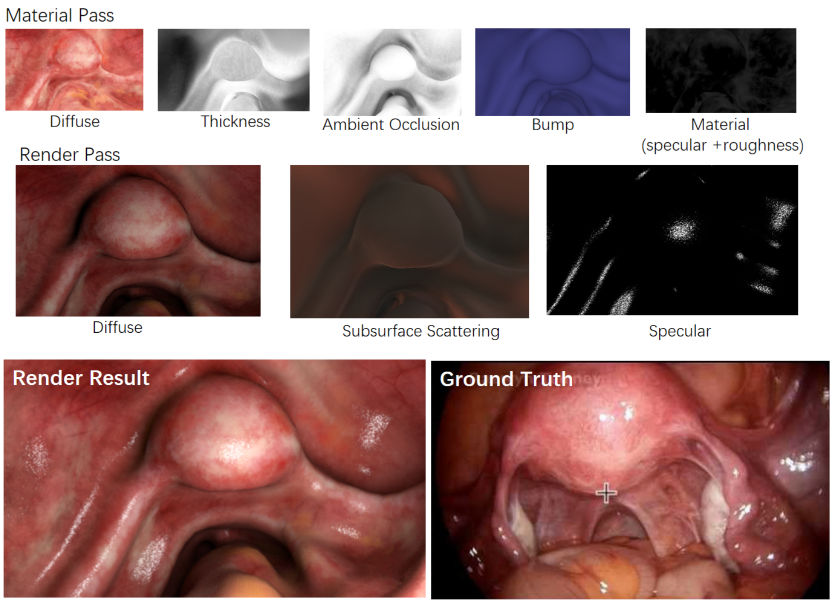

Figure 6.14: Illustration of material pass, rendering pass, final render result (before compositing) and comparison with ground truth.

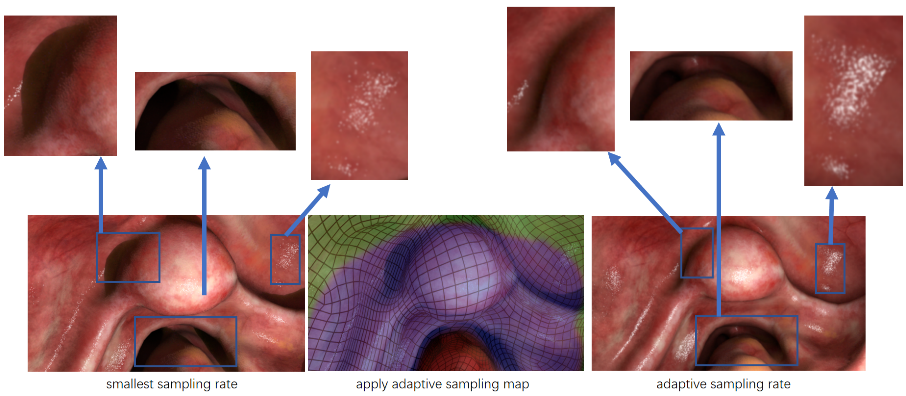

Figure 6.15: Comparison between the lowest sampling rate and painting based adaptive sampling.

6.3.3 Post Processing Pipeline

In laparoscopic surgery, the surgeons perform the operation based on the video recorded by the camera. The resolution of the device and type of monitor will both influence the present of the anatomy. Normally, the color, noise, blur and sharpness etc. may vary a lot among different recording and display devices. From the Figure 6.14, it is easy to find that the rendering result are too clear and clean, which is different from the camera recording effect of real world objects.

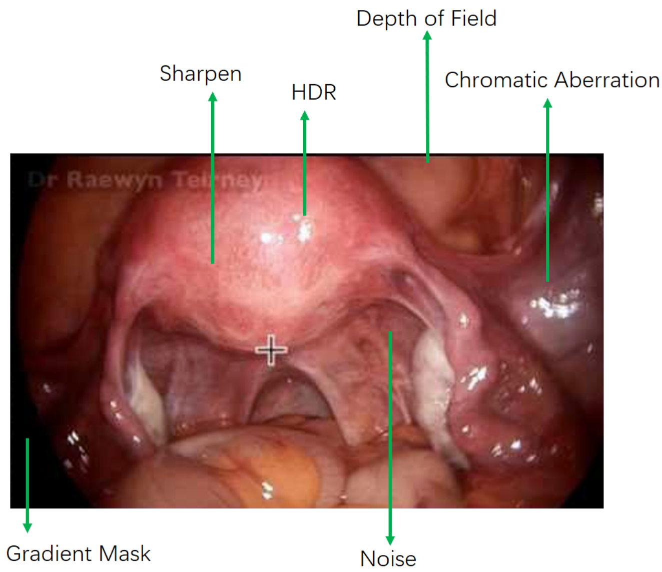

Figure 6.16: The post effect analysis of the laparoscopic surgery video.

In this thesis, a set of widely used post processing techniques are integrated into the rendering pipeline, which can achieve various effects that caused by the recording device. From the general view, the effect of the recorded video are corresponding to various post-processing effects, as can be seen in Figure 6.16. Detail analysis about those effects will be in the following part of this section.

Chromatic aberration is one of the most important effect in the post processing pipeline. In the real-world photography, when a lens fails to render all color channels at the same convergence point, the chromatic aberration happens. As can be seen in Figure 6.17[3], the chromatic aberration effect is very common in the video of laparosocpic

[3] https://reshade.me/forum/shader-presentation

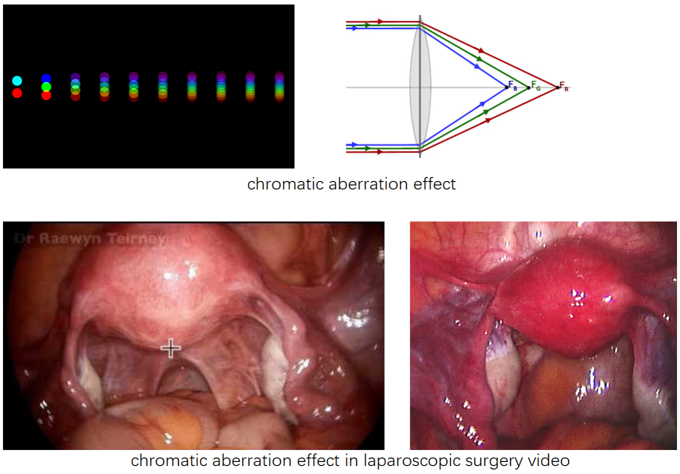

Figure 6.17: The chromatic aberration effect in laparoscopic surgery video.

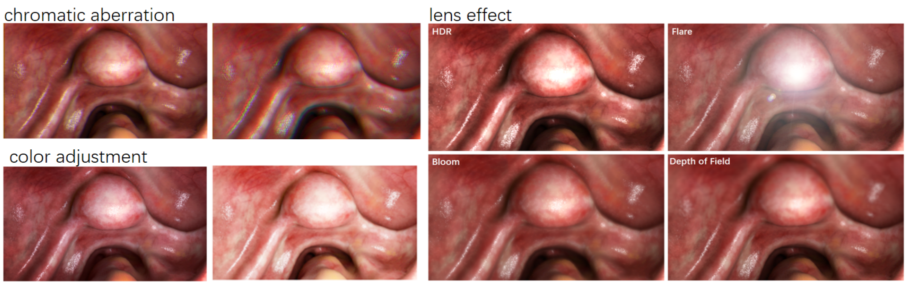

Figure 6.18: Illustration of post processing effect which include chromatic abberation, color adjustment and lens effect.

surgery especially the low quality recording device. However, the CG lighting can not generate this effect so this can be achieved in the post-processing stage. As can be seen in Figure 6.18, this effect can be achieved by offsetting the pixel RGB channels.

Besides the chromatic aberration effect, the color tuning is also important to the matching the rendering result to the color theme in the real surgery video. Tone mapping, filmic adjustment, curve editing are used (see Figure 6.18).

The lens effect are also important to the photorealism of the ren-

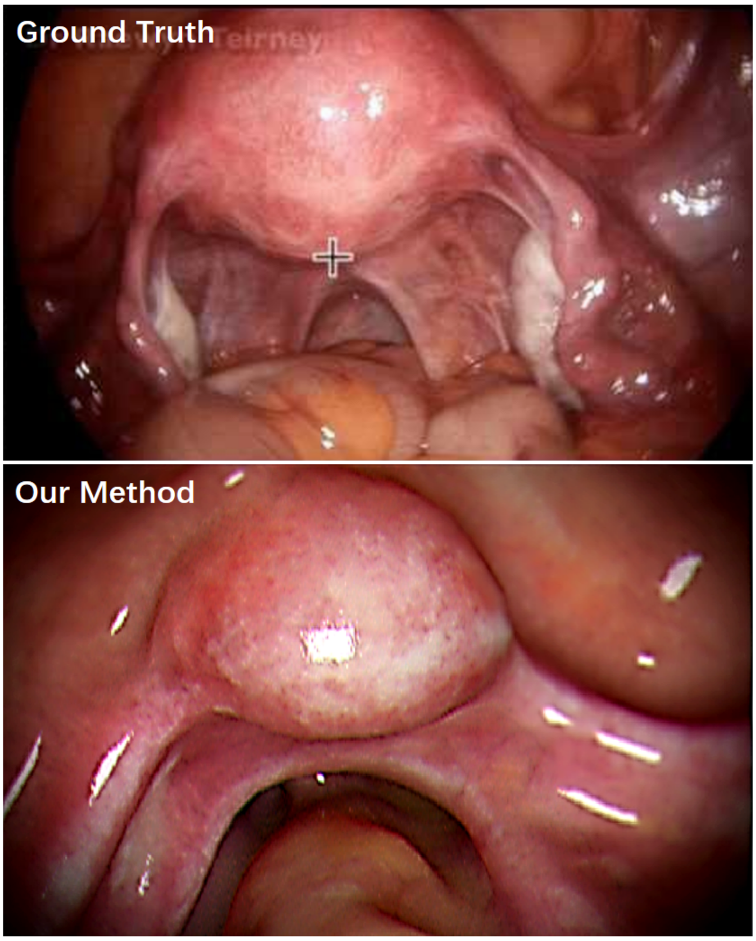

Figure 6.19: The comparison of the final processing effect with the ground truth.

dering result. Due to the strong lighting and high reflective soft tissue membrane, HDR and flare effect can often be seen from laparoscopic surgery video. The depth of field is another important lens effect to simulate the effective focus range. Noise and blur are often associate with the above effects. Figure 6.18 show applying those effects to the rendered image.

The gradient mask is used to simulate the effect of the view from laparoscopic camera. A radial gradient at the center of the image can

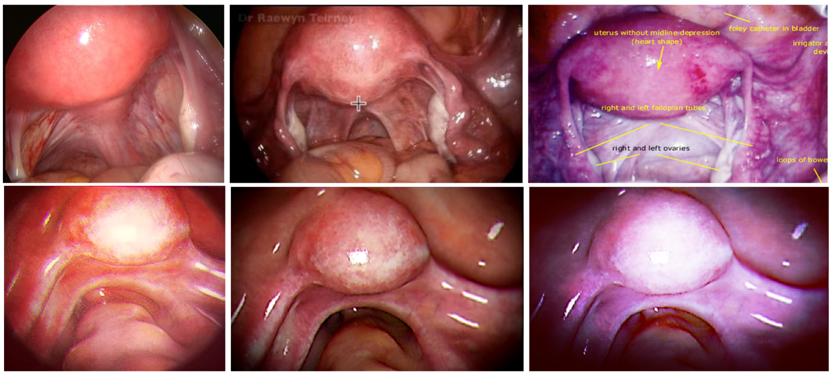

Figure 6.20: Illustration of different post processing styles (second row) and comparison with ground truth (first row).

achieve this effect. By using the chromatic aberration, sharpen, noise, HDR, depth of field and gradient mask, the final rendering effect and its comparison is shown in Figure 6.19. The combination of the above post processing techniques can achieve various effects, as can be seen in Figure 6.20[4]. How to choose the techniques is dependent on the required effect.

6.4 Summary

In this chapter, a realistic soft tissue rendering method based on procedural organic material generation and multi-layer soft tissue rendering pipeline is proposed. The procedural organic material pipeline is based on a quad tree based procedural system which can generate user controlled organic pattern materials for physically based rendering pipeline. In this research, the soft tissue rendering is dependent on the biomechanics based multi-layer structure. This multi-layer structure has been integrated into a physically based rendering pipeline to produce realistic soft tissue rendering.

[4] http://www.mondoginecologico.it/chirurgia/chirurgia-endoscopica/

Chapter 7

Laparoscopic Surgery Simulator Development

7.1 Introduction

Developing a laparoscopic surgery simulator is the interdiscipline of physics simulation, computer graphics and software engineering. It requires not only the key technologies proposed in this thesis but also the engineering pipeline which can efficiently integrate those technologies as a chain of processing elements and automatically control the data flow through the pipeline. To evaluate the techniques proposed in this thesis, this chapter will illustrate how to integrate those techniques into a pipeline for developing realtime laparoscopic surgery simulator. The detailed performance analysis of the system in each component has been made in this chapter.

7.2 System Overview

In general, the system can be divided into three components: geometry processing, simulation and rendering. In the first stage, the anatomy

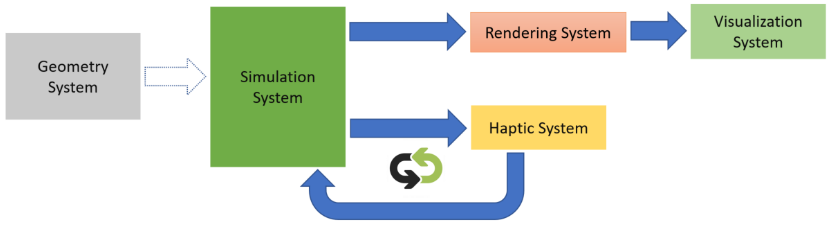

Figure 7.1: The general system pipeline and relationship between each subsystem.

data and surgical tool models are firstly sent to the geometry processing unit. The geometry processing unit is an auxiliary tool for optimizing mesh (chapter 3) and painting necessary information for simulation (chapter 4.3.5) but it is not directly involved in the realtime simulation. If the 3D model is simulation ready, it can skip the mesh optimization process and enter the painting process.

After the geometry processing system, the geometry data will be sent to GPU for simulation. The simulation includes the computation of soft tissue deformation (chapter 4) and surgical tool interaction (chapter 5). The computation result will be sent back for the rendering and haptic feedback.

For the rendering system, it takes the output of simulation system and packs the data for rendering. This rendering pipeline includes a physically based deferred shading system which consists of multi-layer soft tissue rendering and post processing (chapter 6). The rendering result can be sent to different types of display devices. For the object which needs haptic feedback, the geometry data should be sent to the haptic rendering loop to let the device capture the shape of the object.

The simulation and rendering components are the core of the pipeline which will be introduced in details in the following part.

7.2.1 GPU Based Simulation System

Physics simulation is the most computationally expensive part in the pipeline. To achieve better realtime performance, the simulation system is implemented based GPU parallel computation.

The data transfer between CPU memory and GPU memory is an expensive operation. As can be seen from Figure 7.2, normally the data is created on the pageable memory. Before sending to GPU, it needs to be firstly transferred to pin memory then can be sent to device memory. To reduce the transfer cost, the pinned memory in the physical memory (RAM) is used which will not be swapped out but has limited capacity. The copy from pinned memory is faster than pageable memory due to the DMA (Direct Memory Access) which can direct fetch this memory without the help of CPU. The data need to be transferred between GPU and CPU in each frame include the 3D model position, contact information (type, time) and dissection information for each vertex and primitive. On the GPU cores, the main computation power is used on constraint solving for deformation, collision detection, dissection and clip. The deformation constraint can simulate wide range of soft tissue behaviours based on continuum mechanics (see chapter 4). The collision, dissection and clip constraints take the haptic tool information as input and perform circumsphere based collision, dissection and clip operations (5).

7.2.2 Deferred Shading Based Rendering Pipeline

The rendering is based on a deferred shading pipeline. Deferred shading is a screen space rendering technique which means the shading is not performed in the first render pass. Instead the shading is deferred until the second pass. In the first pass, the data for shading computation is collected, such as depth, normal, material id etc. This data collection

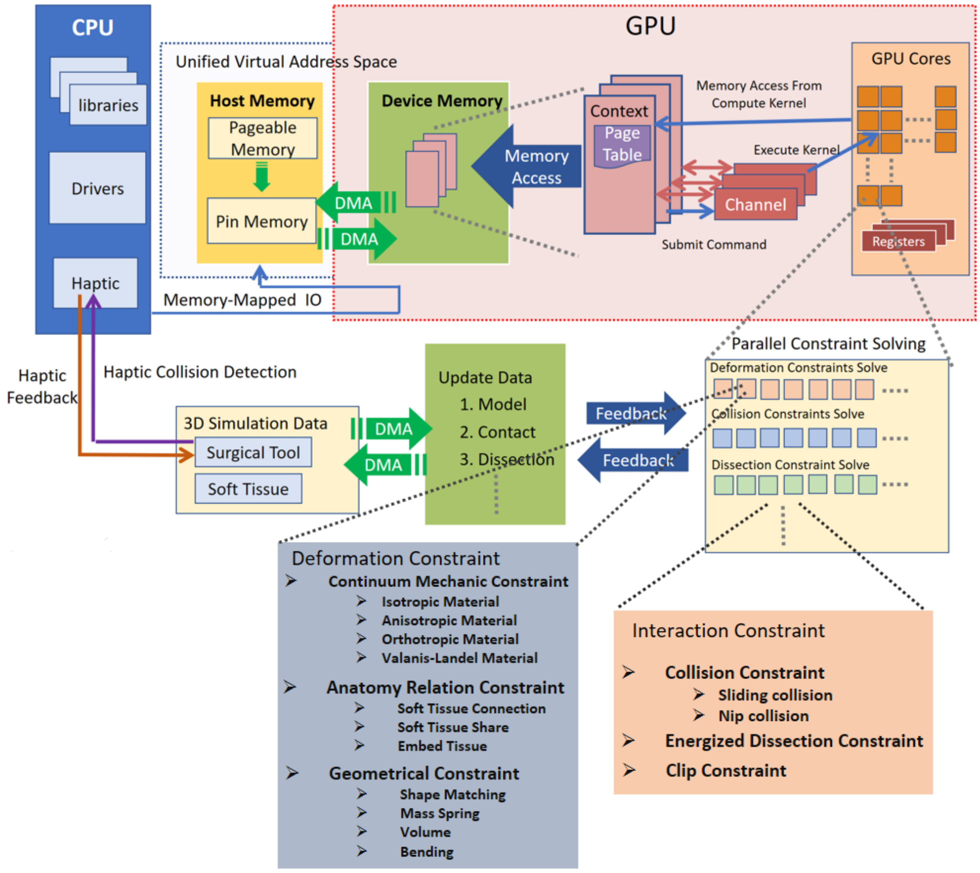

Figure 7.2: GPU based simulation pipeline.

is also called the geometry buffer (G-Buffer) filling. In the second pass, actual shading is performed in the screen space based on the geometry information in the G Buffer. The reason to use deferred shading is that it decouples scene geometry from lighting. It provides an ease of using computationally efficient screen space rendering techniques and convenient for managing complex shading effect.

The design of the rendering pipeline is shown in Figure 7.3. Baking necessary maps for rendering is the first stage of the rendering pipeline. Baking maps means transferring mesh and material based information into texture map. Those maps are then ready for sampling in the later

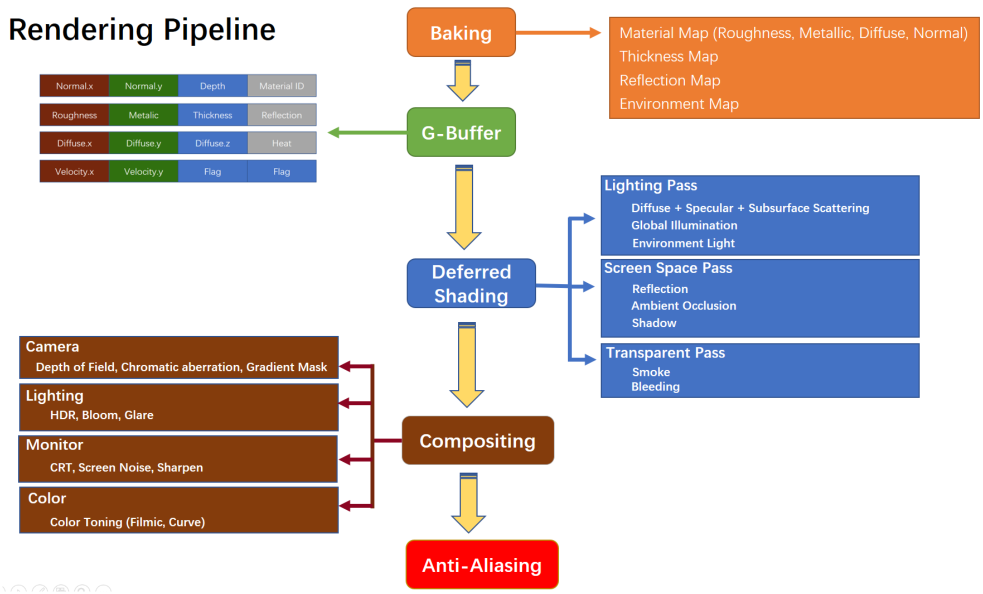

Figure 7.3: Rendering Pipeline Overview.

shading stage to provide material and auxiliary information.

In the G-Buffer stage, besides the traditional shading information, specific information is added for generating special rendering effect for surgery simulation. As can be seen in Figure 7.4, the heat, velocity and two flag channels are added into G-Buffer. The heat will determine the color of dissected tissue. Two flag channels can be used for advanced effect such as smoke and bleeding effect.

RGBA16	Normal.x	Normal.y	Depth	Material ID
RGBA8	Roughness	Metalic	Thickness	Refection
RGBA8	Diffuse.x	Diffuse.y	Diffuse.z	Heat
RGBA8	Velocity.x	Velocity.y	Flag	Flag

Figure 7.4: G-Buffer overview.

In the deferred shading stage, the main shading operations are performed. The lighting pass is firstly performed for diffuse, specular and subsurface scattering. The global illumination and environment light are optional according to the requirement. Instead of using GPU based global illumination, the computationally efficient polynomial based representation is used for irradiance environment maps [RH01]. However, due to the limited and close environment in laparoscopic surgery and single light source, global illumination and irradiance environment maps are not necessary for some cases such as weak light intensity environment. For reflective materials, the screen space based reflection technique [HFK16] is used. However, due to this technique is based on light tracing, the reflection importance map is used to further improve the efficiency of reflection computation. The importance reflection map is grayscale map which specify the reflective region. The screen space effect pass is responsible for the ambient occlusion and shadow in the scene. We used the classical SSAO and shadow map for each effect.

In the compositing stage, the techniques are classified into camera, lighting, monitor and color based effects. More details can be found in chapter 6.3.3.

7.3 Development of Laparoscopic Surgery Simulator

In this research, a laparoscopic surgery simulator prototype is developed which can provide basic laparoscopic skill training. To improve the effectiveness and efficiency in the training, the design and structure of the simulator should be similar to the real clinical platform so that it can provide the medical practitioners another alternative to sharpen their surgical skills.

There are mainly two different formats of laparoscopic surgery. One is robotic based laparoscopic surgery. It will make use of the technology such as da Vinci system which is based on a platform located away from the patient and controlled by doctors sitting in front of a true minimally invasive 3-D view of the surgical field. The other is the surgeon based surgery. The surgeon will directly manipulate the surgical tool and see the enlarged surgical elements on a TV monitor during the surgery. In this section, the simulator is mainly used for the surgeon based surgery training which is still the common way of performing laparoscopic surgery in clinic theater.

According to the above facts, the design of the simulator includes two parts: the hardware design and software design. The hardware design focuses on the building of the training platform (surgeon based surgery training) and the software design concentrates on the development of the virtual training environment.

7.3.1 System Configuration

In laparoscopic surgery, instruments (such as graspers, scissors, clip applier) will be inserted into the body's trocars and the surgeons will hold the instruments to perform the surgery. To simulate this environment, a surgery box is designed, on which holes are provided to play the role of trocars. The real surgery tools will be inserted into the holes and connect to a pair of haptic devices which are responsible for providing force feedback. For the haptic device, Omni Phantom [1] is chosen because it can meet the requirement of manipulation degree of freedom in surgery, provide continuous force feedback, support flexible customization and have been validated in various successful simulators [ECNN$^+$16]. The real laparoscopic surgery tools are connected to the haptic device via an interface in the surgery box. The virtual reality controller can also be attached to the surgery tools which can synchronize the tool transfor-

[1] http://www.geomagic.com/.

mation to the VR headset to provide more immersive user experiences.

Figure 7.5: System Hardware.

For the display device, a TV monitor is often used in clinic theater to visualize the surgical operations inside the patient abdomen. The options of the monitor can be 2D and 3D [WRK$^+$14]. Using head-mounted 3D display to augment laparoscopic skills especially to enhance the sense of depth for surgeon is the recent trend of laparoscopic visualization [ACW$^+$15]. In the development of this simulator, both 2D, 3D [2] and head-mounted 3D monitors [3] are used for different training purposes.

For the software, the system used the in-house engine which is developed based on the research of this thesis. It includes the geometry processing, physics simulation, rendering and visualization. The geometry processing mainly includes the research proposed in chapter 3 which is responsible for the simulation ready model generation. As the core of the system, the physics simulation contains the research in chapter 4

[2]https://www.asus.com/us/Monitors/VG278H/.
[3]https://www.vive.com/us/

which includes continuum mechanics based deformation, dissection and collision simulation. The rendering mainly includes the multi-layer soft tissue rendering research in chapter 6.

The system is based on C++ which can provide high level abstraction and low level hardware (haptic, VR headset etc.) access easily. To make the system compatible on multi-platform and easily communicate with used hardware, OpenGL is used for the graphics part of the system. For the haptic device programming, OpenHaptics is used which is a C++ and OpenGL based haptic library provided by the hardware vendor of Omni Phantom. For the VR headset, 3rd party API OpenVR are used which allows the access to VR devices from different vendors without knowing the hardware they are targeting at. For the GPU based optimization, CUDA is used. The system is running on a workstation with Microsoft Windows 7 system, Intel XEON E5-1650 CPU, 32GB RAM and Nvidia GTX960 graphic card.

7.3.2 System Implementation

The whole system implementation for the rectum cancer surgery simulator can be seen in Figure 7.6. Each node represents a specific function.

The Soft Tissue node takes Model node as input which is responsible for the loading of 3D model data. The Soft Tissue node can represent various types soft tissue (fat, membrane and embedded tissue). The property of the tissue can be specified by the external curve file or traditional FEM based method (see chapter 4.3). The response to the collision can also be specified in the Soft Tissue node. The collision update rate and circumsphere factors (φ, ς) are corresponding to the parameters in section 5.2.4 which are responsible for balancing the stability, accuracy and efficiency of the collision.

The Haptic Device node is responsible for the loading of the proxy

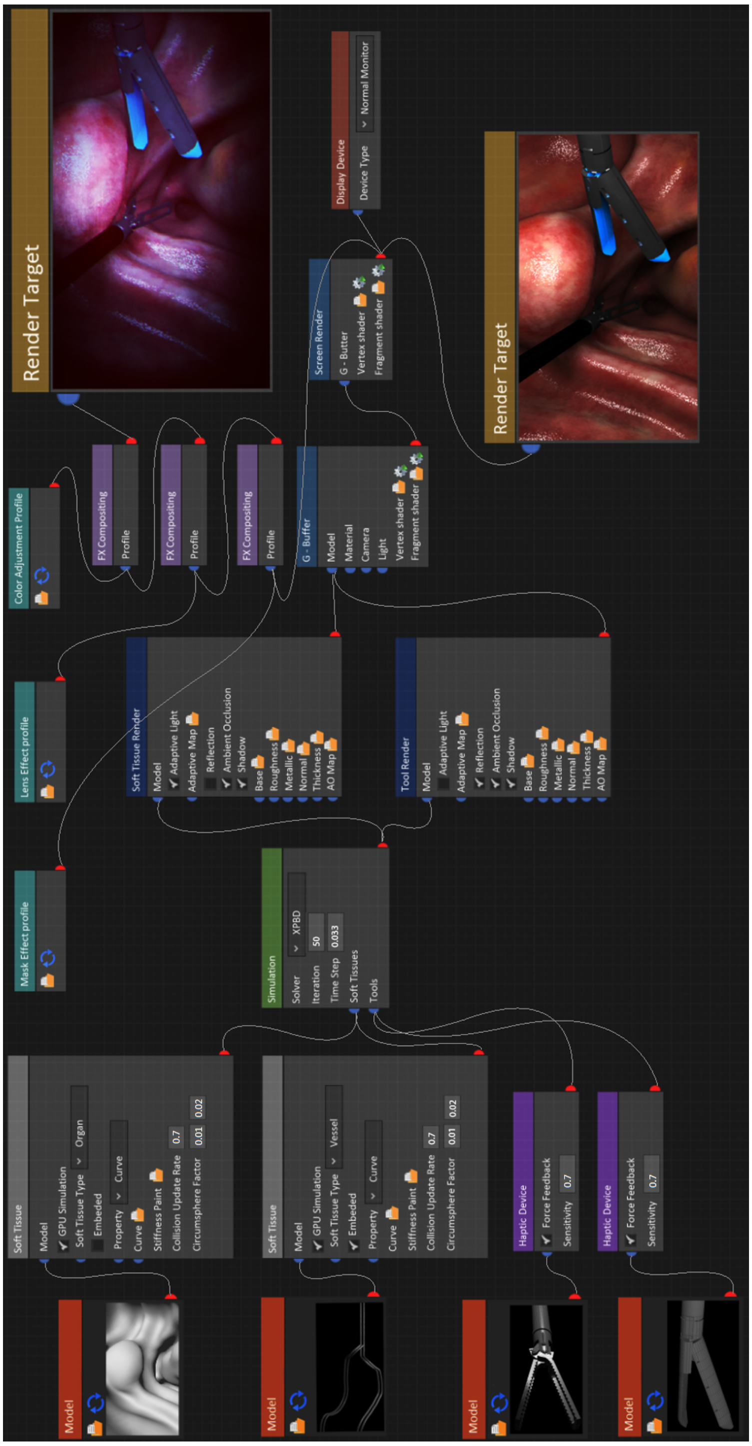

Figure 7.6: The work flow of the rectum cancer surgery system.

model for virtual surgery tools. The sensitivity and force feedback parameters can be adjusted in this module.

The Simulation Node is the core of the system. It takes the Soft Tissue node and Haptic Device node as the input. The Simulation Node will take the constraints that provided by the Soft Tissue node and Haptic Device Node. Inside the Simulation node, it will feed the input data into the specified solver (XPBD is used). The simulation result will be fed to the corresponding render node.

The Soft Tissue Render node and Tool Render node are responsible for the rendering of soft tissue and surgical tool. The lighting mode and materials can be specified in this node as the form of external file. The adaptive lighting only used on Soft Tissue Render node but not used for Tool Render node due to the darkness on the surgery tool is often not that obvious so the smoothness between dark and bright area is not necessary.

G-Buffer node takes the Soft Tissue Render node, Tool Render node, Shaders, Camera Node and Light Node as the inputs. It fills the G-Buffer using the render data in the way described in section 7.2.2. Follow the rendering pipeline, the Screen Render node will take the G-Buffer node information and perform the final render.

The FX Compositing node is responsible for the post processing. Each FX Compositing Node takes an Effect Profile as input. The Effect Profile includes Color Adjustment Profile, Lens Effect Profile and Mask Effect Profile as shown in section 6.3.3. In the post processing pipeline, the chromatic aberration and tune mapping are used for creating the basic color theme, the depth of field, screen blur is used for lens effect and the gradient mask are applied.

The above is the system implementation of the rectum cancer surgery application. For the development of training module, it is made up of deliberate and repetitive hands-on training of basic and procedural laparoscopic skills on simulators. For each type of skill training task, a specific simulation modality is used. The training modules include peg transfer, pattern cutting, ligating, eye hand coordination, clip and dissection, as can be seen in Figure 7.8.

The training module does not have high require on the realism of soft tissue. A simplified pipeline of rectum cancer surgery is used. Features such as subsurface scattering in the lighting pass, the whole compositing pass are turned off. Due to the fact that it is a simplified version of the simulator system, details will not be covered here.

7.3.3 Performance Analysis Implementation

The Figure 7.7 records the procedure of operating the simulator. It includes navigation, tissue split (sliding contact), vessel clip and dissection. The detailed performance is sampled during each type of operation.

It can be seen that during navigation stage the cost of the system mainly spend on deformation simulation and haptic device. To capture the continuous force feedback, the geometry needs to be rendered to haptic device to make it aware the topology of the mesh at the rate of 1000 Hz. The deformation simulation in the navigation stage will not cost too much computational power because no large deformation happens so the expensive operations such as SVD is not involved. The cost on collision detection mainly spends on the broad phase detection so cost is comparatively small because no contact happens yet in this stage.

In the soft tissue split stage, sliding contact is the most frequently performed operation. The narrow phase collision detection and resolution consumes larger computational power than the navigation stage

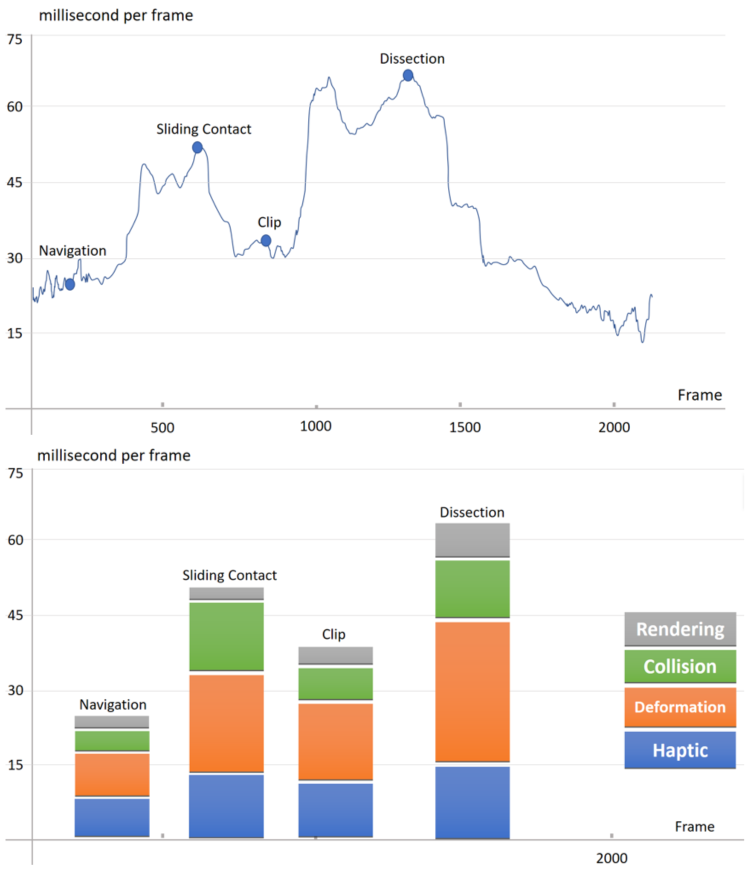

Figure 7.7: Overall performance analysis.

because the circumspheres need to be adjusted and updated to provide good mesh approximation and protection to high detailed area. The deformation cost in this stage increases due to the large deformation simulation. Meanwhile, frequent sliding contact causes the increase of cost on haptic collision processing.

In the clip stage, the focus of the operation is the interaction between the tip of the surgery tool and the vessel. Compared to the previous stage, the frequency of sliding contact reduced a lot so the cost for the collision and haptic decreased. The cost for deformation also decreased due to the operation only limited to the vessel and its connected tissue.

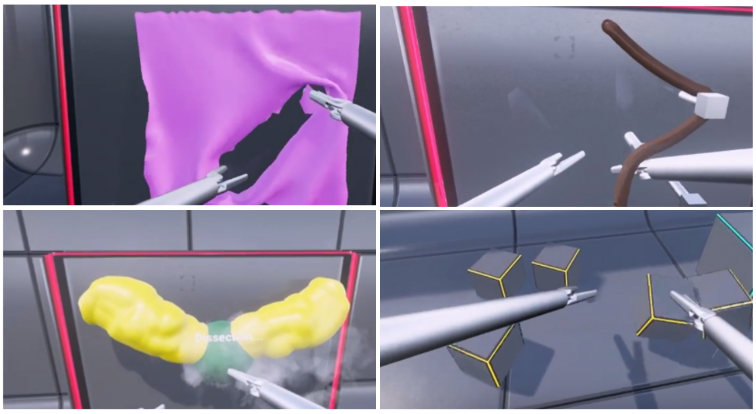

Figure 7.8: The modules for laparoscopic skill training.

The rendering cost in this stage increased a little due to the processing of visual mesh.

In the dissection stage, the energized dissection is the main operation in this stage. The cost for the deformation simulation increased significantly due to the heat transfer based dissection and large deformation simulation. The collision and haptic remains the same level as that in the previous stage because the interaction in this stage is still limited between the tip of the surgery tool and soft tissue. The increase of the rendering cost is caused by the smoke, newly generated incision surface rendering.

7.4 Summary

In this chapter, a development pipeline based on the research of this thesis is proposed. The development pipeline is composed of the geometric process, simulation, rendering and post process. The simulation and rendering are the core of the whole pipeline. Based on this pipeline, a laparoscopic skill training system and rectum cancer surgery application are developed and the detailed system performance analysis is

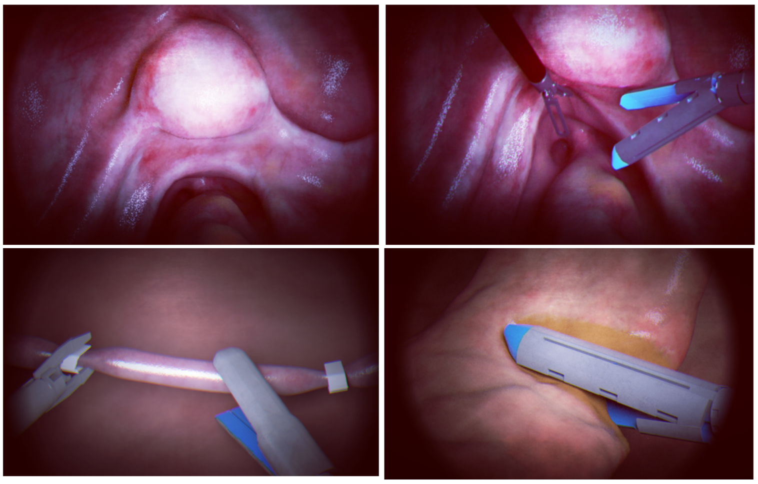

Figure 7.9: Illustration of the simulator system for analysing.

made in this chapter.

Chapter 8

Conclusion and Future Works

8.1 Summary and Conclusion

This thesis focused on improving the visual realism of laparoscopic surgery simulation from the perspective of computer graphics. It begins from an overview of the existing laparoscopic surgery simulator in Chapter 2. Based on the analysis of the factors that influence the visual realism from the perspective of computer graphics (**OBJ1**), state-of-art works in anatomy modeling, soft tissue deformation simulation, surgery tool interaction and soft tissue rendering have been reviewed (**OBJ2**).

For anatomy modeling, chapter 3 solves the problem of how to design an efficient method for the modeling of complex anatomy structure (**Q1**). To design an efficient geometric method which can take advantage of the already existing 3D anatomic models on the market and converting it to simulation ready model (**OBJ3**), this thesis proposed a voxelization and remesh based pipeline which can keep the shape of original 3D surface model but eliminate the ill shaped and degenerate polygons without influencing existing artistic pipeline (**C1**) and a surface mesh parameterization transfer algorithm which can transfer the original surface parameterization (UV mapping) to the simulation ready model without distortion in the parameterization space (**C2**).

For the soft tissue simulation, chapter 4 worked on how to improve the realism of soft tissue simulation and make it more easily editable to achieve user desired effect (**Q2**). To design a physically based soft tissue modeling method which can reflect the muti-layer nature of soft tissue and provide intuitive physics based user control to soft tissue property (**OBJ4**), this thesis proposed a nonlinear multi-layer soft tissue simulation method. It includes a multi-layer soft tissue modeling method which generalize the soft tissue into fascia, fatty and embedded tissues (**C3**), a hyperelastic based curve editing system which enables the user to control the physical behaviour of soft tissue using intuitive physically based curve system (**C4**) and a skeleton based stiffness painting system for generating heterogeneous soft tissue (**C5**).

For the surgery tool interaction, chapter 5 targeted at how to improve the interaction between surgery tools and soft tissues from the perspective of accuracy, stability and efficiency (**Q3**). To develop a robust surgical tool interaction system which can provide stable sliding contact, energized dissection and clipping operations (**OBJ5**), this thesis proposed 1) an adaptive circumsphere based collision detection and resolution method which can reduces collision tunnelling effectively (**C6**) and provide stable and robust collision response (**C7**), 2) a computationally efficient dissection model based on heat transfer which provides physically based modeling for the dissected area (**C8**), 3) an efficient geometric based soft tissue clip method which can keep the clip attaching on the vessel and producing visual plausible clip result without introducing the rigid body simulation (**C9**).

For the soft tissue rendering, chapter 6 aimed at how to achieve realistic rendering of soft tissue (**Q4**). To design an efficient pipeline for realistic multi-layer soft tissue rendering which includes material asset generation, rendering and post-processing (**OBJ6**), a procedural organic material generation pipeline which enables the users to efficiently generate organic look texture based on real images (**C10**) and a multi-layer soft tissue rendering rendering pipeline which includes multi-layer soft tissue shading, optimized lighting effect modeling and post-processing (**C11**) have been proposed.

To test the feasibility of the proposed techniques, chapter 7 focused on the development of a laparoscopic surgery simulator based on proposed techniques in this thesis. It includes the hardware setup, GPU based simulation pipeline, deferred shading based rendering pipeline. The detailed performance analysis has also been made in this chapter.

In conclusion, the visual realism of laparoscopic surgery simulation is determined by various issues (modeling, simulation and rendering). Those issues will influence the performance of laparoscopic surgery simulation from the perspective of efficiency, stability and accuracy. In this thesis, a set of tailored techniques have been proposed from the perspective of computer graphics to solve those challenges. Due to the lack of standard computer graphic production pipeline for laparoscopic surgery simulation, a feasible system pipeline for the development of simulator has also been proposed based on those tailored techniques which can benefit both the researchers and practitioners from industries.

8.2 Future Works

From the anatomy modeling perspective, this paper only focused on generating simulation ready model based on existing 3D anatomies. Due to the shape variation of the anatomy model is very limited, the training platform can only provide monotonous training task under limited anatomical conditions. For the users who used the simulator for several times, such tasks will quickly become boring and the training becomes less useful. For a high fidelity surgery simulator, providing variational anatomy shapes according to the patient specific data is the future of surgery simulator [HCGD17], which can make the training progress more practical and significant. Creating 3D anatomies for each patient is not practical. Due to the fact that the shape variations between human anatomies is not large, generating patient specific models based on a high quality template model is promising. The template model can be generated from the modeling pipeline in this research. The challenge is how to design an effective surface registration algorithm that can establish surface correspondences between patient specific data and template model, then deform the template model into the desired target shape. Ali recently et al. [DLG+13]) proposed a promising anatomy transfer pipeline which can transfer a reference anatomical model from an input template to an arbitrary target character. The surface registration in this pipeline is based on the classic Iterative Closest Point method [ZF03] [BM92] which is not computationally efficient and accurate enough especially under extreme conditions. Recently, learning based registration methods is popular. Huang et al. [HKC+17] proposed a convolutional networks based shape descriptor learning method from part correspondence which can be quite useful for the registration of detail feature of anatomy model.

From the simulation perspective, the whole simulation pipeline is based on polygon mesh representation. The main drawback of polygon based representation is that the accuracy of the simulation is dependent on the resolution of the mesh. To achieve realtime performance, low resolution mesh has to be used which is the source of artifacts such as zigzag incision area when performing dissection. The polygon representation constrains the topology relation of the mesh vertices which makes the topology changing operations in simulation very complex because the simulation constraints and rendering surface are determined by the topology relationship. Use more simple mesh representation for the soft tissues which will be dissected or split frequently during simulation will benefit the simulation efficiency and accuracy significantly. Meshless representation is a good candidate for this purpose which has been widely used in fluid simulation [MM13]. How to design the kernels for particles will determine the physical property of the simulated object. For meshless based deformation simulation, there are many solid works [BJ14] [GBB09] [MKN+04] on designing the kernel types to simulate various physical properties. Recently, a shape matching based large elasto-plastic deformation technique is proposed et al.[CMM16] which can adaptively change the kernel size to adapt to the needs of shape change. This technique is quite promising to create soft tissue that can be arbitrary dissected without introducing zigzag boundary artifact in the polygon based method. However, the challenging part of using meshless mesh representation is how to render surface especially the surface with parameterization information (such as texture UV coordinate) on it. In [CMM16], a novel surface tracking method has been proposed which can combine the procedural material generation techniques in this research to project the multi-layer organic patterns onto the tracked surface of the meshless object. How to design the optimized projection plane or projection shape to ensure the smooth distribution of the pattern on the meshless object is an interesting topic for further research.

From the user experience view, how much realism is enough for surgery simulation is an unsolved problem yet. Finding the valid point of balance between efficiency, accuracy and stability is significant for the development of surgery simulator, which needs support from clinic theater. The haptic feedback of current surgery simulator still depends on physical behaviour of virtual object. When the simulation of virtual object is not accurate, the accuracy of haptic feedback will be influenced greatly. However, this thesis did not cover how to improve the haptic fidelity based on limited visual realism which is a quite important issue especially when the simulation hardware can not provide high realism simulation due to the lower update rate caused by heavy computational tasks but want to achieve believable user experience. Recently, an adaptive prediction method for smooth haptic rendering is proposed et al. [HS16] which can be very helpful for generating smooth user experience for complex and computationally intensive scenarios. Following this path and designing a clinic validated computer graphics realism standard has great impact on the development of laparoscopic surgery simulator because it can offer more definite and clear goals for the development which can improve the efficiency of development significantly.

Chapter 9

Appendix

9.1 Global Based Simulation Method Analysis

The traditional way of solving this large linear system is using Newton's method [BW98] but the high cost limits its utilities in realtime applications because it requires solving a linear system which changes after each iteration ($\nabla \mathbf{f}$ is changing in each iteration). Instead of solving the implicit integration directly, a variational form of implicit integration is described in [MTGG11] which formulate the problem as a non-convex optimization problem. It operates directly on elastic potentials rather than forces:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \underbrace{\frac{1}{2h^2}(\mathbf{x}-\tilde{\mathbf{x}})^{\mathbf{T}}\mathbf{M}(\mathbf{x}-\tilde{\mathbf{x}})}_{first} + \underbrace{E(\mathbf{x})}_{second} \tag{9.1}$$

The first term of the equation 9.1 reflect the trend of attracting $\mathbf{x}$ to predicited state $\tilde{\mathbf{x}}$ according to Newton's law. The second term will penalize the elastic deformation. The energy term is dependent on the constraint type. The violation of constraint will cause the growing of the energy term. The elastic deformation can be viewed as the process of penalizing the growing energy and restoring the deformed state to the rest state. The critical points of the above non-convex minimization problem ($\nabla \mathbf{f} = 0$) is actually the following equation (which can be derived from equation 4.33 and 4.34).

$$\mathbf{M}(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}) = h^2 \mathbf{f}(\mathbf{x}^{n+1}) \tag{9.2}$$

where the $\tilde{\mathbf{x}} = 2\mathbf{x}^n - \mathbf{x}^{n-1} = \mathbf{x}^n + \mathbf{v}^n h$ represents the inertial of the object.

Instead of using Newton's method, projective dynamics [BML+14] becomes popular in recent years. This method will produce a global linear system which is not dependent on the runtime state (can be prefactorized).

$$(\frac{\mathbf{M}}{h^2} + \mathbf{L})\mathbf{x} = \frac{\mathbf{M}}{h^2}\tilde{\mathbf{x}} + \sum_i \omega_i \mathbf{S}_i^T \mathbf{G}_i^T \mathbf{p}_i \tag{9.3}$$

where $\mathbf{S}_i$ is the selection matrix which select the points that involved in the ith constraint manifold. $\mathbf{p}_i$ is the auxiliary variable which store the projection result of $\mathbf{S}_i\mathbf{x}$ onto the ith manifold, $\mathbf{G}_i$ is the discrete differential operator. Projective dynamics is based on an alternating (local $\setminus$ global) optimization. The local step will fix $\mathbf{x}$ and solve $\mathbf{p}_i$ for each constraint manifold then store the projection of $\mathbf{S}_i\mathbf{x}$ onto the constraint manifold Ω_i to $\mathbf{p}_i$. The global solve step will treat $\mathbf{x}$ as variable and fix $\mathbf{p}_i$. Projective dynamics can be solved very fast per iteration. The whole minimization process in projective dynamics will find a balance between inertia motion and elastic behavior. However, the main drawback of projective dynamics is the formulation of the energy term, which is a quadratic distance measures between the current state and projected result. That means the energy is limited to the quadratic form so the classical constitutive models from continuum mechanics , such as St. Venant-Kirchoff, Neo-Hookean etc., not be used. Also the global solve step of projective dynamics requires solving a linear system which is normally dependent on a direct solver. However, the direct solver can not be easily accelerated by parallel computation according to the previous research [Wan15]. When topology change happens, the direct solver becomes inefficient because the linear system can not be prefactorized.

Liu et. al. [LBK17] interpreted the projective dynamics as a quasi-Newton method. In the Newton's method, the Hessian of the $f(\mathbf{x})$ (in equation 9.1) needs to be computed as the descent direction $-\nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x})$. Instead of computing the problematic Hessian (requires definiteness), Liu used $\frac{\mathbf{M}}{h^2} + \mathbf{L}$ as the approximation of Hessian in the Newton's method. Now the descent direction in Newton's method becomes:

$$(\frac{\mathbf{M}}{h^2} + \mathbf{L})^{-1}\nabla f(\mathbf{x}) = \mathbf{x} - \underbrace{(\frac{\mathbf{M}}{h^2} + \mathbf{L})^{-1}(\frac{\mathbf{M}}{h^2}\tilde{\mathbf{x}} + \sum_i \omega_i \mathbf{S}_i^T \mathbf{G}_i^T \mathbf{p}_i)}_{\mathbf{x}^*} \qquad (9.4)$$

We can see that $\mathbf{x}^*$ is actually the result of one step projective dynamics solve. $\mathbf{d} = \mathbf{x} - \mathbf{x}^*$ can be interpreted as the descent direction. The definiteness of the Hessian can be guaranteed. The Projective Dynamics then can treated as a quasi-Newton method, computing the next iteration as a linear search procedure $\mathbf{x} + \alpha\mathbf{d}$. In Projective Dynamics, $\alpha = 1$ which can guarantee the decrease of the objective function $f(\mathbf{x})$ in equation 9.1 using the Projective Dynamics material. Projective Dynamics as a quasi-Newton method enables the integration of more generalized hyperelastic materials that satisfying the Valanis-Landel assumption but still limited to hyperlastic material. The parallel implementation and optimization for this method still require further research. However, the performance of this method will still suffer topology change which is the common problems for global solving based methods.

9.2 Projective Dynamics and PBD Relation

The first step of PBD is the position prediction, which can be perceived as the first term of equation 9.1. For the energy term in equation 9.1, if the energy is optimized using a Gauss-Seidel or Jacobian style which means treating $E_i(\mathbf{S}_i\mathbf{x}, \mathbf{p})$ as constraint and minimizing each $E_i(\mathbf{S}_i\mathbf{x}, \mathbf{p})$ squentially (equation 9.5), the projective dynamics will becomes PBD.

$$\frac{1}{2}\sum \|\mathbf{G}_i\mathbf{S}_i\mathbf{x} - \mathbf{p}_i\|^2_{\mathbf{F}} = \frac{1}{2}\sum \|\mathbf{x}_{\omega_i} - \mathbf{p}_i\|^2_{\mathbf{F}} = C_i(\mathbf{p}_i) \tag{9.5}$$

where $\mathbf{x}_{\omega_i}$ is the predicted positions of points involved in the i the constraint. Linearizing $C(\mathbf{p}_i)$:

$$C(\mathbf{p}_i) = C(\mathbf{x}_{\omega_i} + \mathbf{p}_i - \mathbf{x}_{\omega_i}) = C(\mathbf{x}_{\omega_i}) + \nabla C^{\mathbf{T}}\Delta\mathbf{x} \tag{9.6}$$

where $\Delta\mathbf{x} = \mathbf{p}_i - \mathbf{x}_{\omega_i}$. The energy minimization problem can be formulated as the Lagrangian

$$\min_{\Delta\mathbf{x},\lambda} f_i(\Delta\mathbf{x}, \lambda) = \frac{1}{2}\sum \|\mathbf{M}_i^{-\frac{1}{2}}\Delta\mathbf{x}\|^2_{\mathbf{F}} + \lambda_i(C(\mathbf{x}_{\omega_i}) + \nabla C^{\mathbf{T}}\Delta\mathbf{x}) \tag{9.7}$$

$$\frac{\partial f_i}{\partial \Delta\mathbf{x}} = \Delta\mathbf{x} + \lambda\nabla C^{\mathbf{T}}\Delta\mathbf{x} = 0 \tag{9.8}$$

$$\frac{\partial f_i}{\partial \lambda_i} = C(\mathbf{x}_{\omega_i}) + \nabla C^{\mathbf{T}}\Delta\mathbf{x} = 0 \tag{9.9}$$

Where $\mathbf{M}_i$ is the mass matrix of the ith constraint. Then get the result which is exactly the same as the PBD update rule (see equation 9.10).

$$\Delta\mathbf{x} = -\frac{C(\mathbf{x}_{\omega_i})}{\|\nabla C\|_{\mathbf{F}^2}}\nabla C(\mathbf{x}_{\omega_i}) \tag{9.10}$$

Bibliography

[ABH+07] Rajesh Aggarwal, Camilo Boza, Julian Hance, Julian Leong, Antonio Lacy, and Ara Darzi. Skills acquisition for laparoscopic gastric bypass in the training laboratory—an innovative approach. *Obesity surgery*, 17(1):19–27, 2007.

[ACW+15] A Ashraf, D Collins, M Whelan, R O'Sullivan, and P Balfe. Three-dimensional (3d) simulation versus two-dimensional (2d) enhances surgical skills acquisition in standardised laparoscopic tasks: a before and after study. *International Journal of Surgery*, 14:12–16, 2015.

[AFA+16] Marco Ettore Allaix, Edgar JB Furnée, Alberto Arezzo, Massimiliano Mistrangelo, and Mario Morino. Energy sources for laparoscopic colorectal surgery: is one better than the others? *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 26(4):264–269, 2016.

[Ale02] Marc Alexa. Recent advances in mesh morphing. In *Computer graphics forum*, volume 21, pages 173–198. Wiley Online Library, 2002.

[AR08] Sylvester Arnab and Vinesh Raja. A deformable surface model with volume preserving springs. *Articulated*

Motion and Deformable Objects, pages 259–268, 2008.

[ATC+08] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM Transactions on Graphics (TOG)*, 27(3):44, 2008.

[ATO15] Iván Alduán, Angel Tena, and Miguel A. Otaduy. Efficient and robust position-based fluids for VFX. In *XXV Spanish Computer Graphics Conference, CEIG 2015, Benicàssim (Castellón), Spain, July 1-3, 2015*, 2015.

[BA07] Margaret Bamforth and Manoj Agarwal. Direct observation of procedural skills (dops). *Workplace-based Assessments in Psychiatry*, page 56, 2007.

[BC00] David Bourguignon and Marie-Paule Cani. Controlling anisotropy in mass-spring systems. In *Computer animation and simulation*, volume 2000, pages 113–123. Springer, 2000.

[BCCB06] H Bechir, Luc Chevalier, Mohend Chaouche, and K Boufala. Hyperelastic constitutive model for rubber-like materials based on the first seth strain measures invariant. *European Journal of Mechanics-A/Solids*, 25(1):110–124, 2006.

[BET14] Jan Bender, Kenny Erleben, and Jeff Trinkle. Interactive simulation of rigid body dynamics in computer graphics. *Comput. Graph. Forum*, 33(1):246–270, 2014.

[BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *ACM Transactions on Graphics (ToG)*, volume 21, pages 594–603. ACM, 2002.

[BG07] Alexander G. Belyaev and Michael Garland, editors. *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, 2007*, volume 257 of *ACM International Conference Proceeding Series*. Eurographics Association, 2007.

[BGTG04] Daniel Bielser, Pascal Glardon, Matthias Teschner, and Markus Gross. A state machine for real-time cutting of tetrahedral meshes. *Graphical Models*, 66(6):398–417, 2004.

[BHW94] David E Breen, Donald H House, and Michael J Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372. ACM, 1994.

[BJ14] Adam W Bargteil and Ben Jones. Strain limiting for clustered shape matching. In *Proceedings of the Seventh International Conference on Motion in Games*, pages 177–179. ACM, 2014.

[BKCW14] Jan Bender, Dan Koschier, Patrick Charrier, and Daniel Weber. Position-based simulation of continuous materials. *Computers & Graphics*, 44:1–10, 2014.

[BLP+13] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Cláudio T. Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Comput. Graph. Forum*, 32(6):51–76, 2013.

[BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[BM94] B Böhm and JW Milsom. Animal models as educational tools in laparoscopic colorectal surgery. *Surgical endoscopy*, 8(6):707–713, 1994.

[BMG99] Daniel Bielser, Volker A Maiwald, and Markus H Gross. Interactive cuts through 3-dimensional soft tissue. In *Computer Graphics Forum*, volume 18, pages 31–38. Wiley Online Library, 1999.

[BML+14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4):154:1–154:11, July 2014.

[BMM17] Jan Bender, Matthias Müller, and Miles Macklin. Position-based simulation methods in computer graphics. In *EUROGRAPHICS 2017 Tutorials*. Eurographics Association, 2017.

[BO02] Gareth Bradshaw and Carol O'Sullivan. Sphere-tree construction using dynamic medial axis approximation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 33–40. ACM, 2002.

[BR09] Ravikanth Balaji and Krishnankutty N Ramachandran. Magnetic resonance imaging of a benign phyllodes tumor of the breast. *Breast Care*, 4(3):189–191, 2009.

[BSBSO] Jürgen A Baier-Saip, Pablo A Baier, Klaus Schilling, and Jauvane C Oliveira. Approximate artery elasticity using linear springs. *Journal of Medical and Biological Engineering*, pages 1–13.

[BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1998.

[BWHT] Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, and Greg Turk.

[BWHT07] Adam W Bargteil, Chris Wojtan, Jessica K Hodgins, and Greg Turk. A finite element method for animating large viscoplastic flow. In *ACM transactions on graphics (TOG)*, volume 26, page 16. ACM, 2007.

[BWK03] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 862–870. ACM, 2003.

[CAR+09] Nuttapong Chentanez, Ron Alterovitz, Daniel Ritchie, Lita Cho, Kris K Hauser, Ken Goldberg, Jonathan R Shewchuk, and James F O'brien. Interactive simulation of surgical needle insertion and steering. In *ACM Trans. on Graphics*. Citeseer, 2009.

[CCI08] Byungjoon Chang, Deukhyun Cha, and Insung Ihm. Computing local signed distance fields for large polygonal models. In *Computer Graphics Forum*, volume 27, pages 799–806. Wiley Online Library, 2008.

[CDA00] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*, 16(8):437–452, 2000.

[CDS+06] Adrian James Chung, Fani Deligianni, Pallav Shah, Athol Wells, and Guang-Zhong Yang. Patient-specific bronchoscopy visualization through brdf estimation and disocclusion correction. *IEEE transactions on medical imaging*, 25(4):503–513, 2006.

[CIRM06] Malcolm Cox, David M Irby, Richard K Reznick, and Helen MacRae. Teaching surgical skills—changes in the wind. *New England Journal of Medicine*, 355(25):2664–2669, 2006.

[CMM16] Nuttapong Chentanez, Matthias Müller, and Miles Macklin. Real-time simulation of large elasto-plastic deformation with shape matching. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 159–167. Eurographics Association, 2016.

[DBL01] *3rd International Conference on 3D Digital Imaging and Modeling (3DIM 2001), 28 May - 1 June 2001, Quebec City, Canada*. IEEE Computer Society, 2001.

[DCB14] Crispin Deul, Patrick Charrier, and Jan Bender. Position-based rigid body dynamics. *Computer Animation and Virtual Worlds*, 27(2):103–112, 2014.

[DDM01] Ara Darzi, Vivek Datta, and Sean Mackay. The challenge of objective assessment of surgical skill. *The American Journal of Surgery*, 181(6):484–486, 2001.

[Del98] Herve Delingette. Toward realistic soft-tissue modeling in medical simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.

[DLG+13] Ali-Hamadi Dicko, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, Francois Faure, Olivier Palombi, and Marie-Paule Cani. Anatomy transfer. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, 32(6), 2013.

[DO00] John Dingliana and Carol O'Sullivan. Graceful degradation of collision handling in physically based animation. In *Computer Graphics Forum*, volume 19, pages 239–248. Wiley Online Library, 2000.

[ECNN+16] David Escobar-Castillejos, Julieta Noguez, Luis Neri, Alejandra Magana, and Bedrich Benes. A review of simulators with haptic devices for medical training. *Journal of medical systems*, 40(4):104, 2016.

[EL07] M. Eitz and Gu Lixu. Hierarchical spatial hashing for real-time collision detection. In *Shape Modeling and Applications, 2007. SMI '07. IEEE International Conference on*, pages 61–70, June 2007.

[ELDY04] Mohamed A ElHelw, Benny P Lo, Ara Darzi, and Guang-Zhong Yang. Real-time photo-realistic rendering for surgical simulations with graphics hardware. In

Medical Imaging and Augmented Reality, pages 346–352. Springer, 2004.

[ENC+08] Mohamed Elhelw, Marios Nicolaou, Adrian Chung, Guang-Zhong Yang, and M Stella Atkins. A gaze-based study for investigating the perception of visual realism in simulated scenes. *ACM Transactions on Applied Perception (TAP)*, 5(1):3, 2008.

[Eri04] Christer Ericson. Real-time collision detection (the morgan kaufmann series in interactive 3-d technology) (the morgan kaufmann series in interactive 3d technology). San Francisco, CA, USA, 2004. Morgan Kaufmann Publishers Inc.

[EWS96] Bernhard Eberhardt, Andreas Weber, and Wolfgang Strasser. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16(5):52–59, 1996.

[FP15] Marco Fratarcangeli and Fabio Pellacini. Scalable partitioning for parallel position based dynamics. In *Computer Graphics Forum*, volume 34, pages 405–413. Wiley Online Library, 2015.

[Fun13] Yuan-cheng Fung. *Biomechanics: mechanical properties of living tissues.* Springer Science & Business Media, 2013.

[GBB09] Dan Gerszewski, Haimasree Bhattacharya, and Adam W Bargteil. A point-based method for animating elastoplastic solids. In *Proceedings of the 2009 ACM SIG-*

GRAPH/Eurographics Symposium on Computer Animation, pages 133–138. ACM, 2009.

[GF08] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3d mesh analysis. In *ACM transactions on graphics (TOG)*, volume 27, page 145. ACM, 2008.

[GHF+07] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (TOG)*, 26(3):49, 2007.

[GLL+17] Leonhard Gruber, Alexander Loizides, Anna K Luger, Bernhard Glodny, Patrizia Moser, Benjamin Henninger, and Hannes Gruber. Soft-tissue tumor contrast enhancement patterns: diagnostic value and comparison between ultrasound and mri. *American Journal of Roentgenology*, 208(2):393–401, 2017.

[GLM96] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180. ACM, 1996.

[GOH06] T Christian Gasser, Ray W Ogden, and Gerhard A Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of the royal society interface*, 3(6):15–35, 2006.

[GP09] Francisco González and Gustavo Patow. Continuity mapping for multi-chart textures. *ACM Transactions on Graphics (TOG)*, 28(5):109, 2009.

[GPM06] ZY Guo, XQ Peng, and B Moran. A composites-based hyperelastic constitutive model for soft tissue with application to the human annulus fibrosus. *Journal of the Mechanics and Physics of Solids*, 54(9):1952–1971, 2006.

[Ham07] Warren I Hammer. *Functional soft-tissue examination and treatment by manual methods.* Jones & Bartlett Learning, 2007.

[HCGD17] Peter R Hoskins, Noel Conlisk, Arjan J Geers, and Barry J Doyle. Patient specific modelling. In *Cardiovascular Biomechanics*, pages 207–230. Springer, 2017.

[HFK16] Lukas Hermanns, Tobias Franke, and Arjan Kuijper. Screen space cone tracing for glossy reflections. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 308–318. Springer, 2016.

[HKC+17] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. Learning local shape descriptors with view-based convolutional networks. *arXiv preprint arXiv:1706.04496*, 2017.

[HNB+06] Ben Houston, Michael B Nielsen, Christopher Batty, Ola Nilsson, and Ken Museth. Hierarchical rle level set: A compact and versatile deformable surface representation. *ACM Transactions on Graphics (TOG)*, 25(1):151–175, 2006.

[Hof10] Naty Hoffman. Background: Physically-based shading. In *SIGGRAPH '10: ACM SIGGRAPH 2010 Courses.* ACM, 2010.

[HPPMI10] Felix C Huang, Carla M Pugh, James L Patton, and Ferdinando A Mussa-Ivaldi. Learning kinematic mappings in laparoscopic surgery. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2097–2102. IEEE, 2010.

[HS16] Xiyuan Hou and Olga Sourina. Real-time adaptive prediction method for smooth haptic rendering. *arXiv preprint arXiv:1603.06674*, 2016.

[Hub96] Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15:179–210, 1996.

[Hum98] JD Humphrey. Computer methods in membrane biomechanics. *COMPUTER METHODS IN BIOMECHANICS AND BIO MEDICAL ENGINEERING*, 1(3):171–210, 1998.

[Hum03] Jay D Humphrey. Continuum biomechanics of soft biological tissues. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 459, pages 3–46. The Royal Society, 2003.

[ITF04] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 131–140. Eurographics Association, 2004.

[JBPS11] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78:1–78:8, 2011.

[JLSW02] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *ACM transactions on graphics (TOG)*, volume 21, pages 339–346. ACM, 2002.

[JP04] Doug L. James and Dinesh K. Pai. Bd-tree: Output-sensitive collision detection for reduced deformable models. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 393–398, New York, NY, USA, 2004. ACM.

[JQL+17] Tao Jiang, Kun Qian, Shuang Liu, Jing Wang, Xiaosong Yang, and Jianjun Zhang. Consistent as-similar-as-possible non-isometric surface registration. *The Visual Computer*, 33(6-8):891–901, 2017.

[KCJG+08] Fernando J Kim, MF Chammas Jr, Gewehr, E, and Morihisa. Temperature safety profile of laparoscopic devices: Harmonic ace (ace), ligasure v (lv), and plasma trisector (pt). *Surgical endoscopy*, 22(6):1464–1469, 2008.

[KCM12] Tae-Yong Kim, Nuttapong Chentanez, and Matthias Müller-Fischer. 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*, pages 305–310, 2012.

[KFN06] Katherine J Kuchenbecker, Jonathan Fiene, and Günter Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE transactions on visualization and computer graphics*, 12(2):219–230, 2006.

[KHM+98] James T Klosowski, Martin Held, Joseph SB Mitchell, Henry Sowizral, and Karel Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *Visualization and Computer Graphics, IEEE Transactions on*, 4(1):21–36, 1998.

[KKY+14] Tatsuhiko Kakisaka, Toshiya Kamiyama, Hideki Yokoo, Tatsuya Orimo, Kenji Wakayama, Yosuke Tsuruga, Hirofumi Kamachi, Taisuke Harada, Fumi Kato, Yosuke Yamada, et al. Hand-assisted laparoscopic splenectomy for sclerosing angiomatoid nodular transformation of the spleen complicated by chronic disseminated intravascular coagulation: A case report. *Asian journal of endoscopic surgery*, 7(3):275–278, 2014.

[LAM01] Thomas Larsson and Tomas Akenine-Möller. Collision detection for continuously deforming bodies. In *Eurographics Conference, 2001*, pages 325–333, 2001.

[LBK17] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph.*, 36(3):23:1–23:16, 2017.

[LBOK13] Tiantian Liu, Adam W Bargteil, James F O'Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)*, 32(6):214, 2013.

[LGK+12] Dongwoon Lee, Michael Glueck, Azam Khan, Eugene Fiume, and Ken Jackson. Modeling and simulation of skeletal muscle for computer graphics: A survey. *Foun-*

dations and Trends in Computer Graphics and Vision, 7(4):229–276, 2012.

[LH06] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics (TOG)*, 25(3):541–548, 2006.

[Li16] Wenguang Li. Damage models for soft tissues: a survey. *Journal of medical and biological engineering*, 36(3):285–307, 2016.

[LJD07] Yi-Je Lim, Wei Jin, and Suvranu De. 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, 2007.

[LSG+09] Christian R Larsen, Jette L Soerensen, Teodor P Grantcharov, Torur Dalsgaard, Lars Schouenborg, Christian Ottosen, Torben V Schroeder, and Bent S Ottesen. Effect of virtual reality training on laparoscopic surgery: randomised controlled trial. *BMJ*, 338, 2009.

[LXB16] Yijing Li, Hongyi Xu, and Jernej Barbič. Enriching triangle mesh animations with physically based simulation. *IEEE Transactions on Visualization and Computer Graphics*, 2016. accepted for final publication.

[Mar07] Hoehn Katja Marieb, Elaine Nicpon. *Human anatomy and physiology*. Pearson Education, 2007.

[MBCM16] Matthias Müller, Jan Bender, Nuttapong Chentanez, and Miles Macklin. A robust method to extract the rotational part of deformations. In *Proceedings of the*

9th International Conference on Motion in Games, pages 55–60. ACM, 2016.

[MBF04] Neil Molino, Zhaosheng Bao, and Ron Fedkiw. A virtual node algorithm for changing mesh topology during simulation. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 385–392. ACM, 2004.

[MC10] Matthias Müller and Nuttapong Chentanez. Wrinkle meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 85–92, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[MC11] Matthias Müller and Nuttapong Chentanez. Adding physics to animated characters with oriented particles. pages 83–91, 2011.

[MCK13] Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Transactions on Graphics (TOG)*, 32(4):115, 2013.

[MCKM14] Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. Strain based dynamics. In *The Eurographics / ACM SIGGRAPH Symposium on Computer Animation, SCA 2014, Copenhagen, Denmark, 2014.*, pages 149–157, 2014.

[MCM16] Matthias Müller, Nuttapong Chentanez, and Miles Macklin. Simulating visual geometry. In *Proceedings of the 9th International Conference on Motion in Games*, pages 31–38. ACM, 2016.

[MCMJ17] Matthias Müller, Nuttapong Chentanez, Miles Macklin, and Stefan Jeschke. Long range constraints for rigid body simulations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 14. ACM, 2017.

[MDM+02] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM, 2002.

[MH07] Elaine Nicpon Marieb and Katja Hoehn. *Human anatomy & physiology*. Pearson Education, 2007.

[MHHR07a] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *J. Visual Communication and Image Representation*, 18(2):109–118, 2007.

[MHHR07b] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, April 2007.

[MHL+09] Anderson Maciel, Tansel Halic, Zhonghua Lu, Luciana P Nedel, and Suvranu De. Using the physx engine for physics-based virtual surgery with force feedback. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(3):341–353, 2009.

[MHTG05a] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM transactions on graphics (TOG)*, 24(3):471–478, 2005.

[MHTG05b] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus H. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.

[Mis13] RK Mishra. *Textbook of practical laparoscopic surgery.* JP Medical Ltd, 2013.

[MKE03] Johannes Mezger, Stefan Kimmerle, and Olaf Etzmuß. Hierarchical techniques in collision detection for cloth animation. 2003.

[MKN+04] Matthias Müller, Richard Keiser, Andrew Nealen, Mark Pauly, Markus Gross, and Marc Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151. Eurographics Association, 2004.

[MLM+05] Ullrich Meier, Oscar López, Carlos Monserrat, Mari C Juan, and M Alcaniz. Real-time deformable models for surgery simulation: a survey. *Computer methods and programs in biomedicine*, 77(3):183–197, 2005.

[MM13] Miles Macklin and Matthias Müller. Position based fluids. *ACM Transactions on Graphics (TOG)*, 32(4):104, 2013.

[MMC16] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54. ACM, 2016.

[MMCK14] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):153, 2014.

[MO06] Cesar Mendoza and Carol O'Sullivan. Interruptible collision detection for deformable objects. *Computers & Graphics*, 30(3):432–438, 2006.

[MRT+10] Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer methods and programs in biomedicine*, 98(3):278–284, 2010.

[MSJT08] Matthias Müller, Jos Stam, Doug James, and Nils Thürey. Real time physics: class notes. pages 1–90, 2008.

[MTGG11] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. Example-based elastic materials. In *ACM Transactions on Graphics (TOG)*, volume 30, page 72. ACM, 2011.

[MtV05] Nadia Magnenat-thalmann and Pascal Volino. From early draping to haute couture models: 20 years of research. *The Visual Computer*, 21:506–519, 2005.

[Mül08] Matthias Müller. Hierarchical position based dynamics. In *Proceedings of the Fifth Workshop on Virtual Reality*

Interactions and Physical Simulations, VRIPHYS 2008, Grenoble, France, 2008., pages 1–10, 2008.

[Mus13] Ken Museth. Vdb: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics (TOG)*, 32(3):27, 2013.

[MWW07] Stefan Marks, John Windsor, and Burkhard Wünsche. Evaluation of game engines for simulated surgical training. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 273–280. ACM, 2007.

[NCP+06] Arturo N Natali, Emanuele L Carniel, Piero G Pavan, Paolo Dario, and Ivano Izzo. Hyperelastic models for the analysis of soft tissue mechanics: definition of constitutive parameters. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 188–191. IEEE, 2006.

[NMCW17] ALP Nunes, A Maciel, LT Cavazzola, and M Walter. A laparoscopy-based method for brdf estimation from in vivo human liver. *Medical image analysis*, 35:620–632, 2017.

[NMK+06] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum*, 25:809–836, 2006.

[NOB16] Rahul Narain, Matthew Overby, and George E. Brown. ADMM $\supseteq$ projective dynamics: Fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 21–28, Aire-la-Ville, Switzerland, Switzerland, 2016. Eurographics Association.

[NvdS01] Han-Wen Nienhuys and A Frank van der Stappen. A surgery simulation supporting cuts and finite element deformation. In *International conference on medical image computing and computer-assisted intervention*, pages 145–152. Springer, 2001.

[Per06] Per-Olof Persson. Mesh size functions for implicit geometries and pde-based gradient limiting. *Engineering with Computers*, 22(2):95–109, 2006.

[PKS10] Simon Pabst, Artur Koch, and Wolfgang Straßer. Fast and scalable cpu/gpu collision detection for rigid and deformable surfaces. In *Computer Graphics Forum*, volume 29, pages 1605–1612. Wiley Online Library, 2010.

[QBY+15] Kun Qian, Junxuan Bai, Xiaosong Yang, Junjun Pan, and Jianjun Zhang. Virtual reality based laparoscopic surgery simulation. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, pages 69–78. ACM, 2015.

[RH01] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

[RNLL10] Nicolas Ray, Vincent Nivoliers, Sylvain Lefebvre, and Bruno Lévy. Invisible seams. In *Computer Graphics Fo-*

rum, volume 29, pages 1489–1496. Wiley Online Library, 2010.

[SB12a] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20:1–20:50, New York, NY, USA, 2012. ACM.

[SB12b] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, page 20. ACM, 2012.

[SCOL+04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[SDF07] Eftychios Sifakis, Kevin G Der, and Ronald Fedkiw. Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 73–80. Eurographics Association, 2007.

[Set96] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[SHGS06] Denis Steinemann, Matthias Harders, Markus Gross, and Gabor Szekely. Hybrid cutting of deformable solids.

In *Virtual Reality Conference, 2006*, pages 35–42. IEEE, 2006.

[SL14] Charles de Rousiers Sébastian Lagarde. Moving frostbite to physically based rendering. In *SIGGRAPH '14: ACM SIGGRAPH 2014 Courses*. ACM, 2014.

[SLF08] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. *ACM Transactions on Graphics (TOG)*, 27(3):64, 2008.

[SO14] Sam Snodgrass and Santiago Ontanon. A hierarchical approach to generating maps using markov chains. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[SOG08] Denis Steinemann, Miguel A Otaduy, and Markus Gross. Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/eurographics symposium on computer animation*, pages 87–94. Eurographics Association, 2008.

[SP06] Alla Sheffer and Konrad Polthier, editors. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006*, volume 256 of *ACM International Conference Proceeding Series*. Eurographics Association, 2006.

[Spe78] F.C. Spencer. Teaching and measuring surgical techniques - the technical evaluation of competencel. *Bulletin of the American College of Surgeons*, 63:9–12, 1978.

[TAOZ12] Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. In *Computer*

Graphics Forum, volume 31, pages 1735–1744. Wiley Online Library, 2012.

[THM+03] Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomerantes, and Markus H. Gross. Optimized spatial hashing for collision detection of deformable objects. In Thomas Ertl, editor, *VMV*, pages 47–54. Aka GmbH, 2003.

[TKH+05] Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, M-P Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, et al. Collision detection for deformable objects. In *Computer graphics forum*, volume 24, pages 61–81. Wiley Online Library, 2005.

[TMC10] Jeff Trinkle, Yoky Matsuoka, and José A. Castellanos, editors. *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*. The MIT Press, 2010.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *ACM Siggraph Computer Graphics*, volume 21, pages 205–214. ACM, 1987.

[TTWM14] Min Tang, Ruofeng Tong, Zhendong Wang, and Dinesh Manocha. Fast and exact continuous collision detection with bernstein sign classification. *ACM Transactions on Graphics (TOG)*, 33(6):186, 2014.

[TW14] Wen Tang and Tao Ruan Wan. Constraint-based soft tissue simulation for virtual surgical training. *IEEE*

Transactions on Biomedical Engineering, 61(11):2698–2706, 2014.

[USS14] Nobuyuki Umetani, Ryan Schmidt, and Jos Stam. Position-based elastic rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 21–30. Eurographics Association, 2014.

[VKZHCO11] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[Wan15] Huamin Wang. A chebyshev semi-iterative approach for accelerating projective and position-based dynamics. *ACM Transactions on Graphics (TOG)*, 34(6):246, 2015.

[WHRO10] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics (TOG)*, 29(4):107, 2010.

[WJST14] Yuting Wang, Chenfanfu Jiang, Craig Schroeder, and Joseph Teran. An adaptive virtual node algorithm with robust mesh cutting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 77–85. Eurographics Association, 2014.

[WRK+14] D Wilhelm, S Reiser, N Kohn, M Witte, U Leiner, L Mühlbach, D Ruschin, W Reiner, and H Feussner. Comparative evaluation of hd 2d/3d laparoscopic monitors and benchmarking to a theoretically ideal 3d pseudodisplay: even well-experienced laparoscopists perform better with 3d. *Surgical endoscopy*, 28(8):2387–2397, 2014.

[WTTM15] Zhendong Wang, Min Tang, Ruofeng Tong, and Dinesh Manocha. Tightccd: Efficient and robust continuous collision detection using tight error bounds. In *Computer Graphics Forum*, volume 34, pages 289–298. Wiley Online Library, 2015.

[WWD15] Jun Wu, Rüdiger Westermann, and Christian Dick. A survey of physically based simulation of cuts in deformable bodies. *Comput. Graph. Forum*, 34(6):161–187, 2015.

[WWM79] Alan Wineman, Douglas Wilson, and John W Melvin. Material identification of soft tissue using membrane inflation. *Journal of Biomechanics*, 12(11):841–850, 1979.

[WY16] Huamin Wang and Yin Yang. Descent methods for elastic body simulation on the gpu. *ACM Transactions on Graphics (TOG)*, 35(6):212, 2016.

[WZ09] Rene Weller and Gabriel Zachmann. A unified approach for physically-based simulations and haptic rendering. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, Sandbox '09, pages 151–159, New York, NY, USA, 2009. ACM.

[XKH+16] Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, page 4. ACM, 2016.

[XLCB15] Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič. Interactive material design using model reduction. *ACM Trans. on Graphics*, 34(2), 2015.

[YMYK14] Yusuke Yoshiyasu, Wan-Chun Ma, Eiichi Yoshida, and Fumio Kanehiro. As-conformal-as-possible surface registration. In *Computer Graphics Forum*, volume 33, pages 257–267. Wiley Online Library, 2014.

[ZF03] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.

[ZOF01] Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pages 194–201. IEEE, 2001.