Simulation of Progressive Cutting on Surface Mesh Model

Hui Zhang, Shahram Payandeh and John Dill
Robotics and Computer Graphics Laboratories,
School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6, Canada
{hzhang | shahram | dill}@cs.sfu.ca

Abstract

Surgical simulation is a promising technology for training medical students and surgery planning. An important requirement for such simulation systems is a method to generate realistic cuts through soft tissue models. In this paper, we propose a surface mass-spring model to simulate the virtual cutting operation using an input device such as a haptic device, multi-dimensional joystick, or mouse. We introduce novel algorithms to subdivide the surface and generate interior structures, which follow the motion of the input device. Two types of progressive cutting are supported in our simulator and implementation methods are discussed. Eventually, these cutting techniques will be coupled with a force feedback (haptic) device and be integrated into a training environment for both open surgery and minimally invasive surgery.

1. Introduction

Virtual reality has been utilized in many application domains; in medicine the use of virtual environments opens new possibilities in the areas of surgical training and planning. Surgical simulators are currently being developed at many research centers and companies to create environments to help train doctors with new surgical instruments and techniques [12][7]. There are many aspects in developing various surgical procedures.

The first challenge for a surgical simulation is that objects must look and behave realistically. Thus models must be based on physical laws governing the dynamic behavior of rigid and deformable objects. The choice between the two commonly used physics-based models, mass-spring model and the finite element method, has long existed. Though finite element method (FEM) offers more accurate modeling than mass-spring models it is computationally more demanding and requires several simplifications for real-time applications. Contrary to standard modeling approaches concerned with fixed mesh topology and which are used in interactions between virtual instrument and object (e.g. probing) cutting modifies the topology of the model significantly. Modification precludes any pre-computation for soft-tissue simulation. In general, simulators utilizing FEM need to introduce considerable simplifications for real-time applications [7] [8] [11]. For example, [1] introduced a novel idea, which maps 3D problem into a 2D auxiliary surface to simplify the FEM calculation. Mass-spring models are widely used in simulation of cutting since it is relatively simple and easy to implement [2] [4] [13] [15].

Soft tissue simulation can be implemented utilizing either surface or volumetric models. In general, volumetric models such as tetrahedral model are chosen since they can simulate objects with interior structure. However, topology modification of volumetric models is extremely complex. For example, tetrahedral elements cut by planar surfaces will fall into one of five different topological cases, based on the number of cut edges and intersected faces [3]. Even with the minimal new elements creation method introduced in [11], five to nine new elements are created for each cut element. To maintain the model is composed only by tetrahedral after topology modification, one interacted tetrahedral is finally split into 17 new ones [6]. Surface mesh models are relatively easy to manipulate compared to volumetric model. The reason hinder people using surface model is that normal surface model cannot denote object interior structure for cutting results.

Less attention has been given to the problem of modifying the topology of deformable models. For example, [11] detailed a method to modify the tetrahedral model simultaneously with instrument movement by introducing a temporary subdivision. Progressive cutting can be denoted in two categories. One meaning is that cuts should occur as the user moves the cutting instrument through the object without lag, as described in [11]. The other is that users may try to follow a specific cutting trail by performing several cuts and join them together. The simulator should allow those separate cuts to be joined together if they are close enough to each other.

Therefore, how to implement a realistic simulation of surgical cutting, which is an important component of a typical surgical simulator, is still an open area. In this paper, we propose a new method and model to generate interior structure within a surface mesh model according to the model's interaction with a cutting instrument. Our simulator can model the two types of progressive cutting we just described. The topology modification method will be described below.

This paper is organized as follows: Section 2 explains the notation we use for cutting. Section 3 introduces our surface model. Section 4 describes our simplification for collision detection and collision response. Section 5 details our method of progressive cutting and generating interior structures. Finally, a discussion and suggestions for future work are given in Section 6.

2. Conceptual Overview

Our surgical training environment is intended to simulate cutting operations in both open surgery and minimally invasive surgery. There is no difference in the underlying topology modification schemes for these two types of simulation; the differences lie in the cutting instruments used. The parameters of the physics-based models, such as spring constants, can be adjusted to simulate different material properties that the system tries to simulate

In our simulation, a cutting operation is defined as the movement of a cutting instrument while penetrating an object. Two contact states between cutting instrument and object exist. One is when the force exerted on the object by instrument is not big enough to penetrate the surface; we call this a *contact state*. Here, the instrument deforms the object. In the other state, the cutting instrument has penetrated the object surface, and cutting is allowed, we call this a *penetration state*.

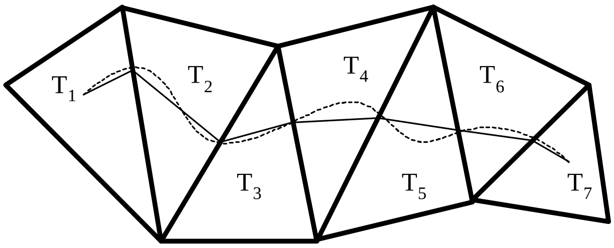

Figure 1: Example of cutting operation.

Figure 1 shows an example of part of an object surface model composed of triangles. Once the instrument makes contact with the object, e.g. triangle T_1, we deem it as the start of one cut. As the instrument moves, the underlying mesh that represents the object is modified by local subdivision. This subdivision algorithm will be explained in Section 5. Depending on the relative size of the mesh and the instrument size, the instrument path may not be a straight line inside a triangle, e.g. the dashed line in Figure 1; however, we approximate any path within a triangle by a straight line segment connecting the first and the last intersection points in the triangle (solid line). If the instrument is lifted up and loses contact with the object, the algorithm enters the *termination state*. For example, triangle T_7 in Figure 1 is the last triangle that the instrument has contacted. The cutting operation in the example starts at triangle T_1, goes through triangles $T_2 \ldots T_6$ and finally finishes at triangle T_7. In this paper, triangles like T_1 and T_7 are called *end* triangles, $T_2 \ldots T_6$

are referred to as *midway* triangles.

2.1 Surface Mesh Model

A surface mesh model is relatively simple for calculating both the vertex displacements and topological modification. This is done through solution of differential equations that model the mass-spring system. However, the surface mesh model lacks volume coherency: if the model is put into a gravitational field, it collapses if no modification is added to the model. We use a rigid kernel which represents the initial rest shape of the object [13], and which is used in the mass-spring model to solve for deformed states with respect to this reference configuration.

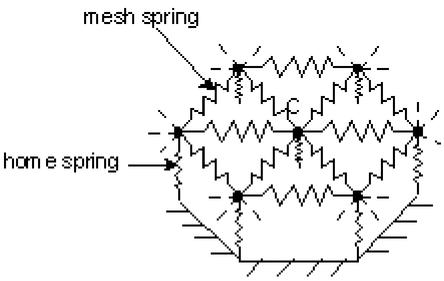

Figure 2: A spring model with rigid kernel

In our model, the simulated surface is divided into small triangles, where each vertex is a mass point (a node). A linear spring is defined along each triangle edge. These springs are called "mesh springs" because they model the surface of the object. When a soft elastic object deforms, the interior of the object also contributes to the shape of deformation. To reflect this fact in our model, each node is also connected by a spring to its initial position, thus the name "home springs". The initial position is called "home position" (home position and rest shape refer to the same configuration). The set of home positions comprise the rigid kernel of the deformable model, which preserves the object's shape. We also added damping to the mesh by applying a force proportional to the velocity of each mass point [10]. Figure 2 shows part of the modeled surface.

2.2 Groove Generation

Surface models typically do not represent the interior of a cut. In our model, we propose a novel approach to generate interior surfaces, referred to as a "groove", which is a function of cutting instrument and object intersection.

This topology modification is implemented by surface subdivision and groove generation as explained in the following section. New vertices are added to the initial mesh. These new vertices are connected with vertices of the original mesh and they have *home springs* connecting them to their initial positions.

We have developed algorithms to subdivide the object surface and generate groove triangles as a function of the path of the cutting instrument.

3.Collision Detection

Cutting instruments come in different shapes and functionalities. As a result, different rendering and collision detection algorithms are needed for determining the state of contact between the tool and the surface.

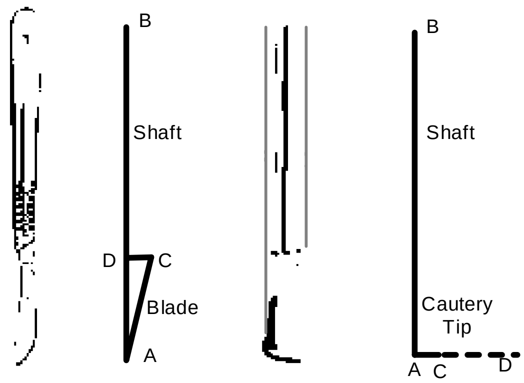

Figure 3: Instrument representation for collision detection

For the purpose of graphical display, instruments are represented by many triangles that can enhance the realistic graphic rendering of the instrument. For collision detection purposes, the instrument representation is simplified to a small set of straight line segments. Collision detection then reduces to detecting intersections between line segments and triangles.

For example, in Figure 3, Line segment AB represents the shaft of the instrument. If it makes contact with the object, it can deform the object without cutting. The user will be able to feel the reaction force, obtained by solving the surface mesh equation [9], using a haptic device. Line segment AC represents the scalpel blade or cautery tip.

```
if (AB intersects one object triangle)
  if (AC intersects that triangle)
        check the movement direction of the scalpel;
        if ( move in the “correct” direction)
           cut;
        else
           calculate the force feedback;
           deform the object;
  else
        if (BD intersects that triangle)
           AC has totally inside the surface,
      calculate the force feedback;
           push AC out of the surface;
        else
            calculate the force feedback;
      deform the object with the scalpel
else
      do collision detection again
```

Figure 4:Pseudo-code for cutting (including haptic feedback)

In the scalpel model, if AC intersects the object surface, the first step is to decide whether the cutting threshold is exceeded. This threshold represents the border between deformation and cutting. The value of the threshold can be tuned experimentally based on real measurements. In our simulation model, the threshold is currently set on the amount of pushing force exerted on the object surface by the instrument. An alternative threshold could be set on surface displacement. Line CD can be used to constrain the movement of the scalpel since only one edge (AC) is considered sharp. This constraint could be conveyed to the user through a haptic force feedback device. The pseudo-code in Figure 4 illustrates this part of the algorithm.

Collision detection for a cautery hook is carried out between line segment CD, which is the extension of line segment AC (cautery tip), and the object surface. When the segment CD intersects the object surface, a cut is allowed. D is the virtual tip for the cautery hook. AC here can also act as a "probe" in that it can deform the surface without cutting and the user can feel such deformation if a force feedback device is used. The length of segment CD is proportional to the intensity of the cauterization. Point D is the virtual tip of the cautery hook.

In all of the above algorithms, we are assuming that all motions are quasi static, i.e. the instruments move slowly in at the surgical site [12] [13]. Hence, movements of the instruments are assumed to be continuous and smooth so the collision detection can be continued locally with neighboring triangles of the current intersected triangle. Global collision detection is only used to find the first intersection triangle, enhancing system performance.

4. Surface subdivision and groove generation

We subdivide surface triangles to simulate the “tearing open” process, i.e. cutting. Due to the nature of the cutting task described in section 2, separate subdivision algorithms are needed for triangles, e.g. end and midway triangles.

As shown in Figure 5a, for example, if the instrument path P_1P_2 (P_1 is the initial contact of the instrument with the surface) only intersects one edge of the triangle (ABC), this triangle can be either the start or the end of a cut. This triangle is subdivided into four new triangles; say (ABV_1, ACV_1, BV_1V_2, CV_1V_3) in b, where V_1 is at the position of P_1 and V_2 and V_3 are corresponded to P_2. Here the newly generated vertices V_2 and V_3 are coincident where they are connected to the object surface by springs along the edges of the triangle. Though V_2 and V_3 are initially coincident, spring forces, obtained through solution of the new mesh equations, will move them apart (Figure 5c and d).

As mentioned above, since the objects are simulated by a surface mesh, when they are cut, the inside of the cut is not represented, i.e. appears be empty. We have proposed a method to generate a cutting groove in the opening as a function of the depth of the instrument penetration, see Figure 5c. When intersection of the surface and tip of instrument is below the surface, we use the positions at the start and end of cut inside this triangle (ABC) to define the bottom of the groove (G_1G_2). In Figure 5, G_1 is the corresponding tip position of the instrument when it is at the intersection point P_1 and G_2 for P_2. As a result, we can generate four triangles inside the opening as groove triangles, as ($V_1V_2G_1$, $V_1V_3G_1$, $G_1G_2V_2$, $G_1G_2V_3$).

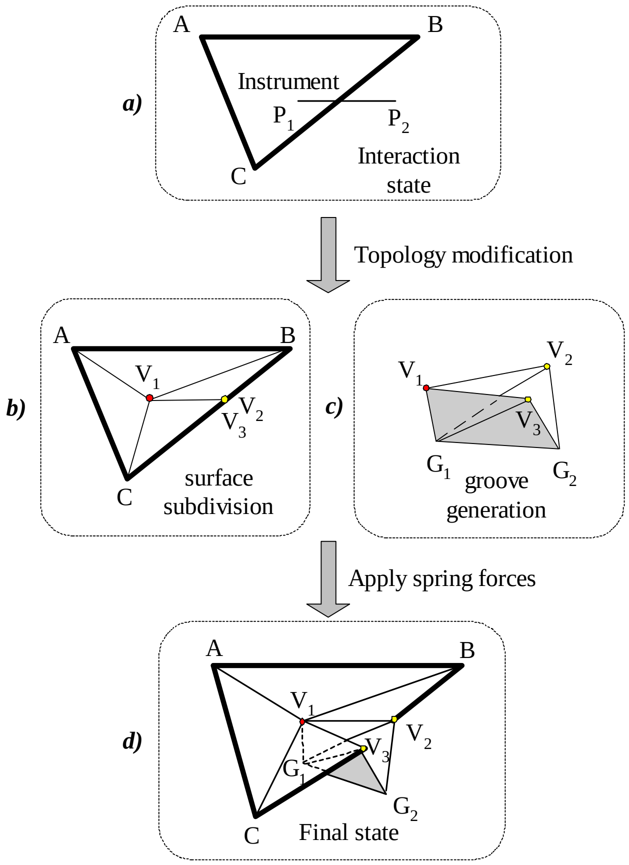

Figure 5:Surface subdivision and groove generation for start and end triangles

The algorithm we just described deals with either the start or the end triangle of a cut. We need another algorithm to subdivide surface triangles and continue the groove generation for midway triangles. If the instrument path intersects two edges (AB and AC in Figure 6a) of a triangle, that triangle must be in the middle of a cutting process.

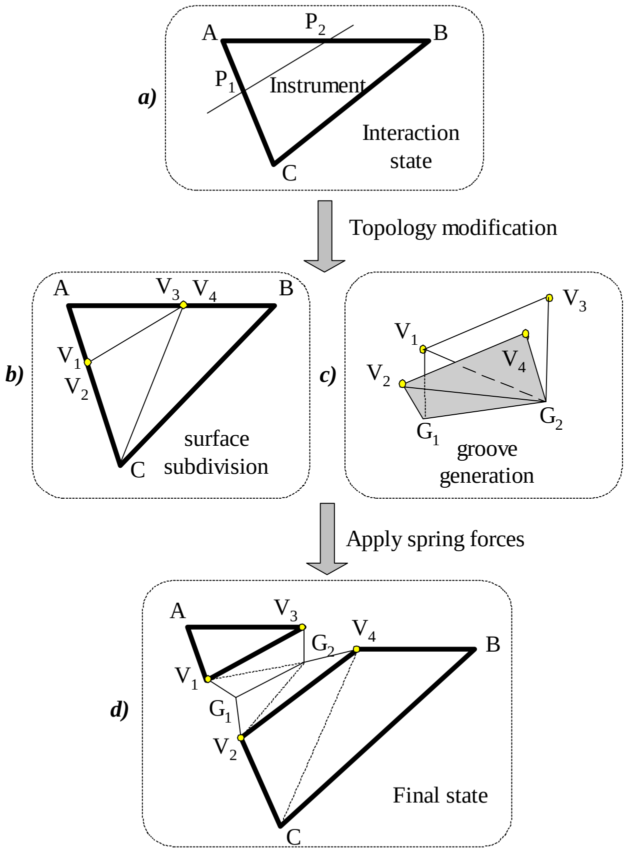

Figure 6: Surface subdivision and groove generation for midway triangles

As shown in Figure 6b, triangle ABC is subdivided into three new triangles (AV_1V_3, V_2V_4C, BCV_4). Only two new vertices (V_3 and V_4) are created since the other two (V_1 and V_2) were created by the previous subdivision algorithm. Similar to the previous algorithm, the two new vertices are initially coincident but will move apart later. The bottom of the groove is generated at instrument tip positions (G_1 and G_2) as before, as are the four groove triangles ($V_1V_3G_2$, $V_1G_1G_2$, $V_2V_4G_2$, $V_2G_1G_2$). After applying spring forces, the midway triangle will open up with its groove, shown as the final state in Figure 6d.

Figure 7 shows a shaded wire frame image of a cut on an ellipsoid.

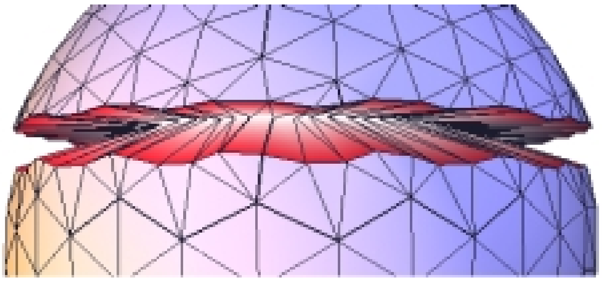

Figure 7: Screen capture illustrating surface subdivision and groove generation

5. Progressive Cutting

In cutting through soft tissue, the user expects immediate visual and haptic feedback as the cut processes, and a simulator that supports this progressive cutting is desirable. The first type of progressive cutting is implemented by generating *temporary subdivisions* within the element that is currently intersected by the instrument. The temporarily generated elements will be replaced by permanent ones when the instrument moves out of that triangle. The second type of progressive cutting, which results in joining two cuts, is implemented by a *re-subdivision method*. We will explain these two types of progressive next.

5.1Progressive cutting with temporary subdivision

We have devised a method of progressive cutting to split triangle exactly when following cutting instrument motion. If the instrument is inside one triangle, that triangle is subdivided according to the current position of the instrument. However, this subdivision is a temporary since the final expected intersection of the instrument and the triangle is still unknown.

The main idea of this progressive cutting is to temporarily subdivide the triangle, which is intersecting the instrument. As shown in Figure 8, if the instrument goes from triangle T_1 to triangle T_2, (dashed lines represent the path of the instrument ($P_1P_2P_3$)), triangle T_2 will be subdivided by assuming the cut is in terminated state. See Figure 9a for better understanding. P_2 and P_2' are at the same initial position as P_2 in Figure 8. If the instrument moves within triangle T_2, the current topology is maintained and the temporary elements are updated using the new position of the instrument, which results in the new point P_3, e.g. Figure 9b.

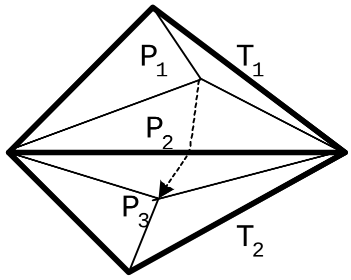

Figure 8:Progressive cutting illustration -1

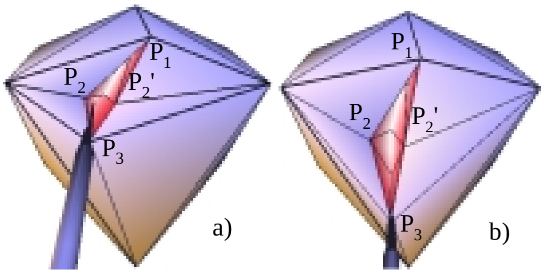

Figure 9: Example of progressive cutting – 1

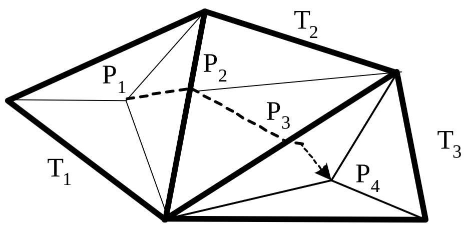

Figure 10: Progressive cutting illustration -- 2

When the instrument goes into another triangle (shown in Figure 10), say triangle T_3, triangle T_2 will be restored to its original shape before subsequent subdivision. Temporarily generated elements (four new surface triangles and groove triangles) are deleted. The procedure then repeats where for example, triangle T_2 is subdivided as a midway triangle, and triangle T_3 will repeat the same temporary subdivision process as triangle T_2. When the whole cut is completed, i.e. the instrument is pulled out from the object, it generates the last triangle which has already been subdivided as an end triangle and the last vertex (P_4 in Figure 10) is kept as the last intersection point position.

An example of this process is shown in Figure 11.

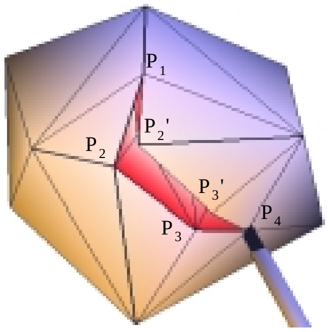

Figure 11: Example of progressive cutting -- 2

5.2 Joining two progressive cuts

So far, our subdivision algorithms result in "free form" cuts composed of sequences of straight line segments. We have defined a cutting operation as the movement of the instrument while it is in contact and penetrating the object. However, with the current algorithm, two individual cuts cannot be joined together and it is impossible to form a loop cut if the instrument goes back into the start triangle.

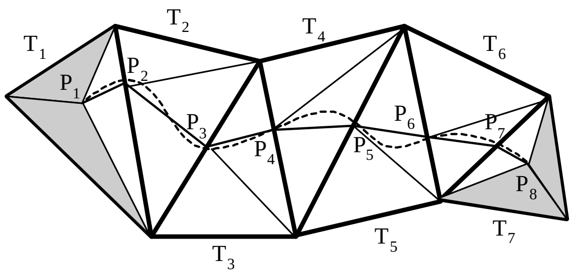

Figure 12: Example of one cut after subdivision

We now extend our current algorithm to deal with this case. Dashed lines in Figure 12 represent the instrument path of a cut. P_1 is the start point and P_8 is the end point. Each cut has four such end triangles (shown in gray shaded area). If the cutting instrument goes into any existing end triangles, the new cut will join together with the previous cut.

As shown in Figure 13, triangle ABC has already been subdivided in previous cut. If the cutting instrument goes into the triangle ABV_1 from edge AB side, AB V_1 is subdivided into two new triangles and vertex V_1 is split into two separate vertices (V_1 and V_1'). Triangles inside the groove also need to be changed to join newly generated groove triangles. The two cuts are then joined together to form a loop cut (Figure 14).

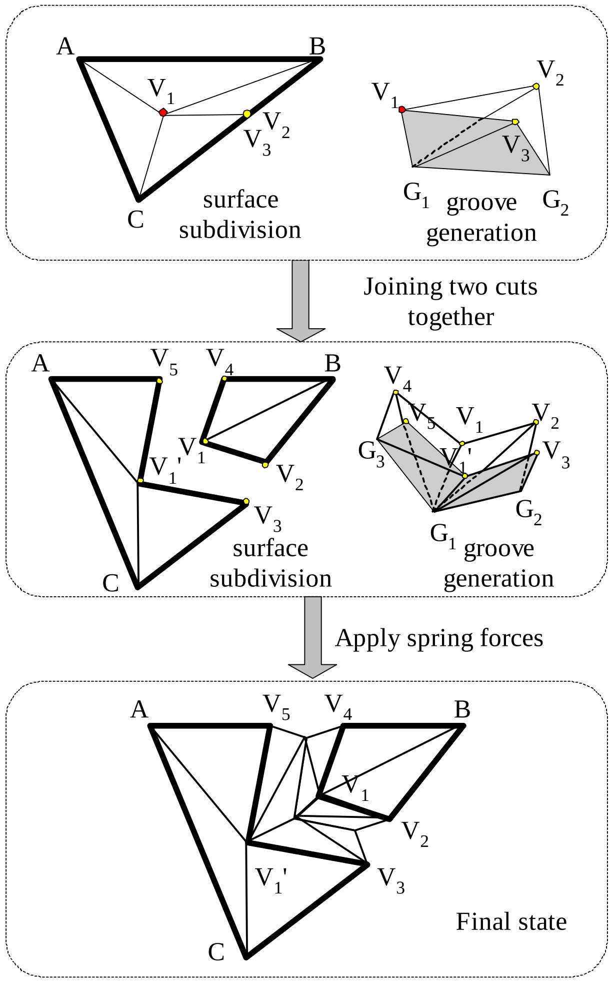

Figure 13: Joining two progressive cuts

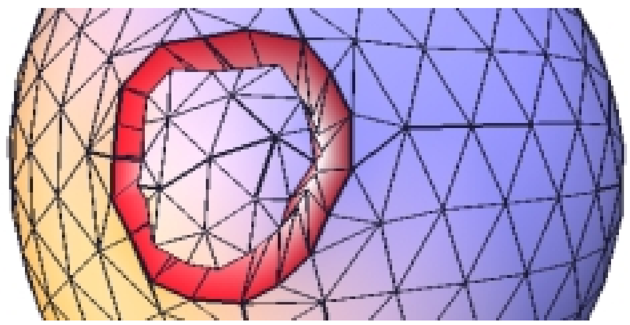

Figure 14: Screen capture of a loop cut

6.Conclusions and future work

To simulate progressive cutting of a deformable object we have extended the surface mesh model such that the interior structure of a cut is modeled and can be visualized. The method to subdivide the surface mesh and generate groove topology is novel and efficient compared to the widely utilized volumetric model. We have implemented a simulator that may be useful as part of a surgical training environment. It can be applied to both open surgery and minimally invasive surgery.

There are several areas for future work. An immediate task is adding support for generating forces to be used to generate haptic feedback. Our current design is aimed at doing this. A second addition would be the ability to “continue” a cut, i.e. to support performing a series of short, joined cuts, as is often done in real surgical cutting. Other areas include investigating behaviors of other cutting instruments and operation methods and dividing an object into two parts.

References

[1] Basdogan, C., Ho, C., and Srinivasan, M.A, “Simulation of Tissue Cutting and Bleeding for Laparoscopic Surgery Using Auxiliary Surfaces”, *Medicine Meets Virtual Reality*. IOS Press, Amsterdam (1999) pp. 38-44.

[2] Beisler, D., Gross, M, “Interactive Simulation of Surgical Cuts”, *Proceedings of Pacific Graphics, IEEE Computer Society Press*, (2000) pp. 116-125.

[3] Beisler, D, V.A. Maiwald, and M.H Gross. “Interactive cuts through 3-dimensional soft tissue”, Proceedings of the Eurographics ’99, volume 18, pages C31-C38, 1999

[4] Bruyns, C; Montgomery, K, "Generalized Interactions Using Virtual Tools Within the Spring Framework: Cutting", *Medicine Meets Virtual Reality (MMVR02)*, Newport Beach, CA, January 23-26,2001.

[5] F.Ganovelli, C.O'Sullivan, "Animating cuts with on-the-fly re-meshing", *Eurographics 2001 - short presentations, pp. 243-247.*

[6] F.Ganovelli, P.Cignoni,C.Montani and R.Scopigno, "Enabling Cuts on Multiresolution Representation" *Proceedings of CGI (Computer Graphics International) page. 183-191*

[7] G. Picinbono, H. Delingette, and N. Ayache, “Improving Realism of a Surgery Simulator: Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation”. *Technical report RR-4018, INRIA*, 2000

[8] Han-Wen Nienhuys and A. Frank van der Stappen, “Supporting cuts and finite element deformation in interactive surgery simulation”. Tech Report 2001/16

[9] Hui Zhang, “Simulation of Soft Tissue Deformation”, Robotics and Computer Graphics Laboratories, Internal Report, Simon Fraser University, 2002

[10] Jian Zhang, Shahram Payandeh, and John Dill, "Haptic Subdivision: An Approach to Defining Level-of-detail in Haptic Rendering", *IEEE HAPTICS 2002*, Florida, March 2002.

[11] Mor, A., Kanade, T, “ Modifying Soft Tissue Models: Progressive Cutting With Minimal New Element Creation”. *MICCAI 2000*, LNCS 1935, Springer, Berlin Heidelberg, (2000) pp. 598-607.

[12] Montgomery, K; Bruyns, C; Brown, J; Thonier, G; Tellier, A; Latombe, JC; "Spring: A General Framework for Collaborative, Real-Time Surgical Simulation", *Medicine Meets Virtual Reality (MMVR02)*, Newport Beach, CA, January 23-26,2001.

[13] P. Meseure et C. Chaillou, "Deformable Body simulation with Adaptive Subdivision and Cuttings", *Proceedings of the WSCG'97* , Plzen, République Tchèque, 10-14 février 1997, pp 361-370.

[14] S. Cotin, H. Delingette, and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation”, *IEEE Transactions On Visualization and Computer Graphics*, January-March 1999,5(1):62--73.

[15] Voss, G., Hahn, J.K., Muller, W., Lineman, R.W, “Virtual Cutting of Anatomical Structures”, *Medicine Meets Virtual Reality*, IOS Press, Amsterdam (1999) pp.381-383.