Shape Forming by Cutting and Deforming Operations

Koichi Hirota*, Atsuko Tanaka**, Toyohisa Kaneko***, Michitaka Hirose*

*Research Center for Advanced Science and Technology, The University of Tokyo
**Department of Information and Computer Sciences, Toyohashi University of Technology
***Central Research and Development Laboratory, OMRON Corporation

Abstract

In this paper, an implementation of visual and haptic feedback of deforming and cutting operations is discussed. In the implementation of the deforming operation, the surface shape is represented by a geometric model while the physical reaction is simulated using a spring model. The deformation of the spring model is reflected onto the geometric model by using the interpolation technique. In the implementation of the cutting operation, we realize visual and haptic feedback of the cutting operation remarking on the geometric and physical aspects, respectively. Combining the deforming and the cutting environment, we successfully implemented a work space in which we can form and design shapes through operations similar to clay modeling.

Key words: cutting, deforming, designing shape, virtual environment, force feedback

1. Introduction

Shape forming is one of promising application area of virtual reality, and various studies on virtual clay-modeling has been carried out. Deforming and cutting operations have been typical means to create shapes, and many modeling softwares provide the deforming and cutting operations. However, most of them do not provide the direct manipulation interface for those operations.

When we are going to realize realistic cutting and deforming operations in environments, we need to implement object models that behave similarly to the real objects according to operations by the user. Since such behavior derive from the physical nature of objects, physically based modeling and simulation is desirable to increase reality in virtual environment. Also, if we are going to feedback the sensation of force in the interaction, computation of force based on the physical model is indispensable. However, it has been a problem that the physically based simulation of cutting and deforming operations generally requires more computation cost compared with the simulation only by geometrical models. Consequently, if we apply the physical model, the complexity of the shape with which we can interact in real-time is strictly limited. On the other hand, importance of presenting force sensation during operations came to be recognized[1], and the computation algorithm of haptic rendering came into an important topic of study[2].

In our study, we investigate methods to simulate cutting and deforming operations with force feedback. Also, by integrating these simulation methods, we implement a virtual modeling environment. Although various tools are used for cutting operations in the real world, cutting operations using a knife or a fret saw is intended in this study.

As we stated above, the complexity of the physical model is limited because of the computation cost. A problem of previous approaches to implementing physically base models is that both of physical and geometrical models are sharing a same structure (i.e., the model of same complexity).

We propose an idea to use two models of different complexity for physical and geometric simulations, respectively[3] (i.e., physical model and geometric model). Also, the geometric model is shared by both cutting and deforming operations. We employ a model in which the shape of objects are defined as a collection of tetrahedral elements (i.e., tetrahedron model).

2. Deforming Operation

These are several studies on the implementation of deformable objects as follows. As a geometric approach, the idea of Free Form Deformation [4] has been proposed in which smooth deformation of shape is realized by applying an interpolation technique to the computation of deformation. However, it is a control-point based approach, and we can not use this approach for direct operation in the virtual environment. Also, to solve the problem, the idea of Direct Deformation Method[5] is proposed. However, if we are going to feedback force, these models must be combined with other model that can compute the interaction force.

There are may studies to introduce physically based models to simulate deformation in computer graphics and virtual reality. In those studies typically two models have been applied: Spring Network model[6]

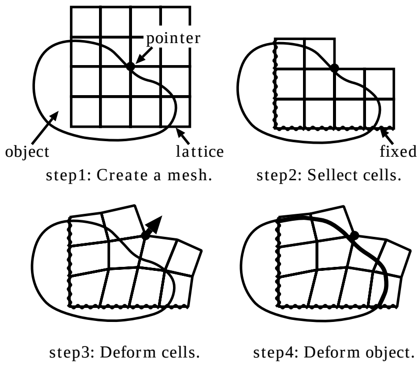


Fig.1: Process of Deforming Operation

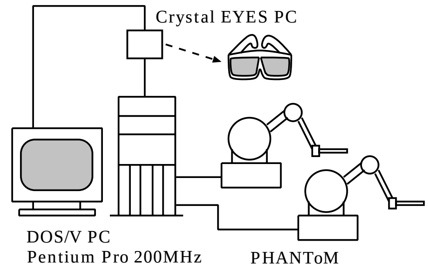


Fig.2: System Construction

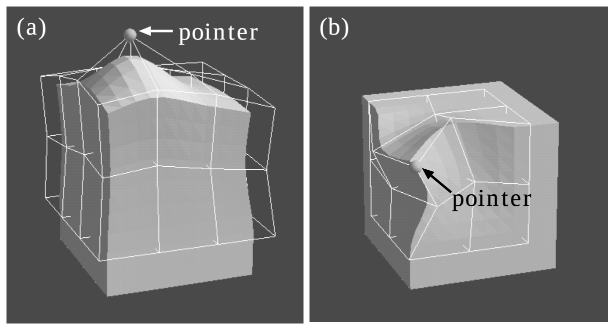


Fig.3: Examples of Deforming Operation

and Finite Element Method model. According to previous studies, the FEM model is not suitable for real-time simulation because it requires high computation cost. Although there is a study to investigate a fast computation method of linear FEM model[7], this approach is not applicable to the simulation of large deformation. In contrast, the Sprig-Network model requires less computation time, and consequently the higher update rate is attained. This is why most studies on the haptic interaction with deformable objects have applied the Spring Network model.

As we stated above, in previous studies, processes of physical simulation and geometric representation are sharing a same structure (e.g., the network of spring is constructed along edges of the polygon model). In this approach, as the resolution of geometric model becomes higher, the physical model also becomes complex.

In our approach, we use separate models for the deforming simulation and the representation of shape, respectively. We employ a spring network model for the physically based simulation of deformation and the result of the simulation is reflected on the precise geometry model. Also, by introducing the condition of breaking into the spring network model we realize the tearing operation.

2.1 Implementation of Deforming Operation

The spring network model consists of cubic cells that are connected with each other at vertices. In each cell, 28 springs are spanned between all of the combination of two nodes among eight nodes. Figure 1 shows the process of deforming computation schematically. Firstly, a spring network that covers a cubic area is created the stylus tip. Next, spring cells that is out of the object volume is deleted so that the shape of the spring network becomes more close to the object shape (i.e., so as to attain better approximation in deforming characteristic). Also, the boundary condition of the spring network is defined (e.g., the vertices at the stylus tip is fixed to the stylus).

During the operation, the position of stylus tip is updated according to the motion of the user, and the deformation of the spring network is simulated. Further more, the shape of the geometric model is changed by computing the position of each node of the tetrahedron model using the interpolation technique based on the algorithm of 3-D Coons Patch.

The block diagram of the system for the prototype implementation is shown in Figure 2. In the system, we use a PC (AT compatible, dual Pentium Pro 200MHz) with an accelerated graphics card (Fire GL 1000, Diamond Multimedia) for the simulation and visual rendering, two PHANToM devices (1.5A, SensAble Technologies)[8] for haptic interaction, and a LC shutter grasses (Crystal Eyes PC, Stereo Graphics) to provide stereoscopic image.

The beginning and the ending of the deforming operation are transmitted to the system by pressing and releasing the button switch of the stylus, respectively.

Figure 3 shows examples of the deforming operation. In figure (a), the user is pulling up about the center of top surface, where a small sphere indicates the position of stylus tip and the spring network for the deforming simulation is represented as a wireframe mesh. We could obtain smooth deformation of polygon model from the deformation of the coarse spring network model. Figure (b) shows an example

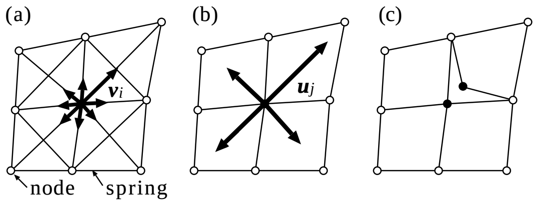


Fig.4: Dividing Spring Model

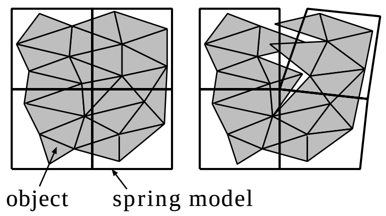


Fig.5: Dividing Tetrahedral Model

of the twisting operation. Since the device we used in this experiment is not capable of representing torque, the sensation of twisting moment is not fedback to the user.

2.2 Implementation of Tearing Operation

Tearing operation is often observed in the clay modeling especially to adjust the volume of clay during the modeling task. The material is torn when the internal stress caused by the external operation exceed the maximum stress that the material can bear. In our experiment, we realize the tearing operation by computing the internal stress of the spring network and locally dividing the network according to the stress.

As is described above, the spring net work consists of cubic cells, and neighboring cells are connected at vertices with each other. In our model, we assumed that the material is broken when the tensile stress exceeds a limit(see Figure 4). We computed the maximum tensile stress in a approximate way. Also, we assumed that the crack is caused perpendicular to the orientation of tensile stress. This assumption is introduced into the model as the algorithm of grouping cells sharing a vertex when the connection at the vertex is cut by the stress. Namely, the grouping is performed based on whether the contribution of each cell to the stress is positive or negative.

The division of the spring network is reflected on the geometry model (i.e., the tetrahedron model) by dividing the tetrahedron model and distributing tetrahedral that are crossing the surface of spring cells exposed by the division (see Figure 5).

Since the volume that is divided by the tearing operation depends on how the user grasps the object. In our implementation, the user grasps the object by two stylus points corresponding to two PHANToM devices. Figure 6 shows the steps of this operation, where the stylus tips are represented by small spheres. Also, it is possible for the user to tear an object apart left and right (see Figure 7).

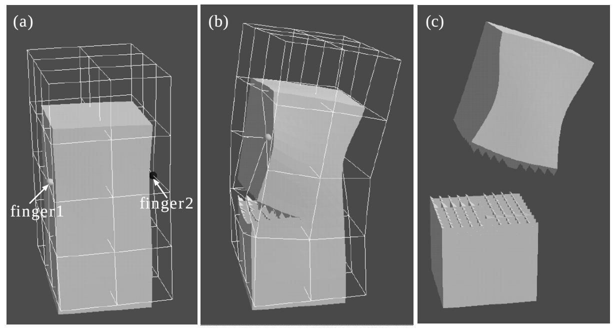


Fig.6: Process of Tearing Operation

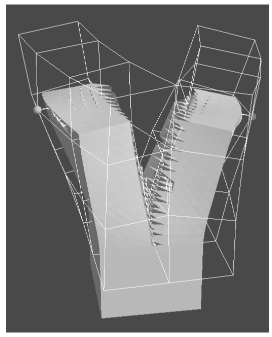


Fig.7: Tearing Operation with Both Hands

3. Cutting Operation

Cutting is one of the most basic operations among various tasks involving shape forming and surgical simulation. However, there are few studies that deal with the force applied during the cutting operation. There is an investigation that implemented a sculpturing operation in a virtual environment[9], where the voxel-based model was used to define the shape. There is also an investigation that introduced force feedback during the sculpturing operation[10]. In that investigation, the force fed back to the operator was determined only from the velocity of cutting the object.

One approach to implementing the cutting operation is to divide the objects geometrically based on the trajectory of the cutting tool. There is an investigation in which the geometrical cutting operation was implemented via a boolean operation on the polygon-based model[11]. However, in that investigation, the analysis on the cutting force was insufficient.

3.1 Computation of Cutting Force

We define the cutting edge as a finite set of discrete points (i.e., discrete edges). By computing the force on each discrete edge, we obtain the approximate distribution of force on the edge. We assume a line-type cutting edge. Namely, the cutting edge is omni-directional and the rotation of the cutting edge around its axis does not cause reacting torque. Also, we assume that the discrete edges are independent of each other. Namely, the status of a discrete edge does not affect the computation of other discrete edges.

The object deforms when the force from the cutting tool is applied. We assume that the force affecting on a discrete edge is proportional to the displacement at the point where the discrete edge collides with the object. To represent this relationship, we introduce the stiffness coefficient.

In the simulation, each discrete edge holds the position of two points. One is the present position of the edge. This is the same as the position where the cutting edge collides with the deformed object. The other is the position of the present colliding point when the deformation is relaxed. This is the same as the position where the cutting edge collides with the object in a nondeformed state. Consequently, the deformation of the object on each discrete edge is calculated as the disparity between those positions.

We modeled three kinds of typical forces that affect the cutting edge: fractional force , cutting resistance, and viscous drag[12]. The progress of the cutting operation is represented by moving the cutting edge in the object, namely, by updating the position of the colliding point based on the force affecting the discrete edge.

The fractional force is introduced to represent the friction between the cutting edge and the object. The force does not contribute to the destruction of the material (i.e., does not contribute to cutting).

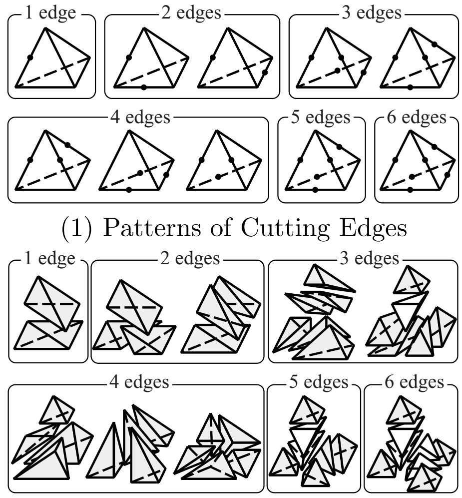
Mechanical cutting is an operation that destroys a part of the material due to the force applied from the cutting edge. This destruction is governed by the shearing force. In our model, the shearing force is approximately computed and the part of the material is destroyed when the shearing force exceed the maximum shearing force that the material can bear.

Viscous drag is the force that is caused as a function of the velocity of the cutting edge. In our model, we assume that the viscous friction is proportional to the velocity of the cutting edge.

3.2 Geometric Cutting

In our implementation, the geometric change caused by the cutting operation is represented by dividing tetrahedra colliding with the trajectory of the cutting edge. The dividing patterns of each tetrahedron is summarized in Figure 8.

Following is the computation flow of the cutting operation (see Figure 9): Firstly, the trajectory of the cutting edge is recorded, and the trajectory surface is defined as a set of triangular patches. Next, the cross points between those triangular patches and edges of tetrahedral cells in the object model are computed. Each cell is divided into parts on those cross points, and each part is re-divided into tetrahedral cells. Finally, the neighboring relation of cells is updated, and the whole object is divided into fragments. The proposed algorithm provides a fast method to compute intersection between the cutting edge and the object approximately.



(2) Patterns of Dividing Tetrahedral

Fig.8: Dividing Patterns

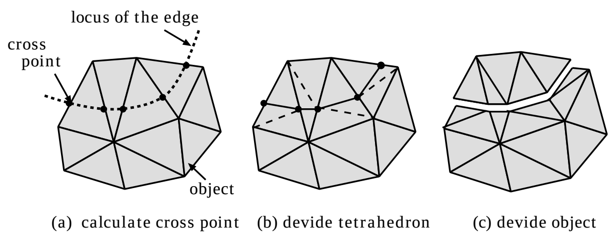
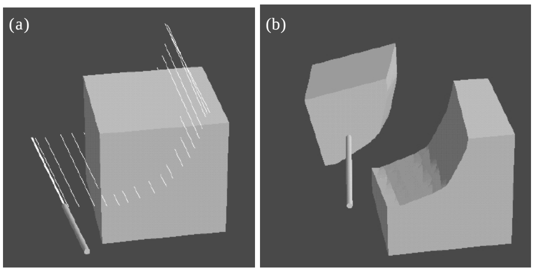


Fig.9: Process of Cutting Operation

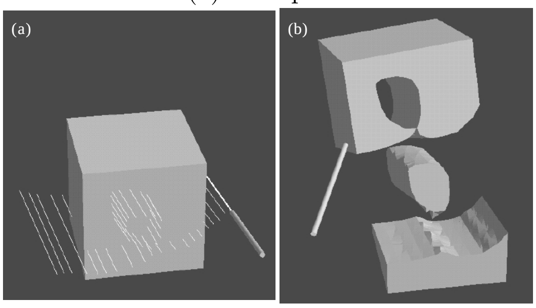
Figure 10 shows examples of cutting operations and resulting shapes. In the case of (1), the shape consisting of 6000 tetrahedral is colliding with the trajectory surface consisting of 18 polygons and took about 4 seconds for the geometric processing.

3.3 Representation of force while cutting

By combining the algorithm of haptic and geometric computations, we implemented a cutting environment. For the fast collision detection between objects and the cutting edge, we employ the voxel mesh surrounding the object. Voxels containing a part of an object are marked in advance to the operation, and we regard that the each discrete edge is colliding with the object when it is in a marked voxel.



(1) example 1



(2) example 2

Fig.10: Cutting Process

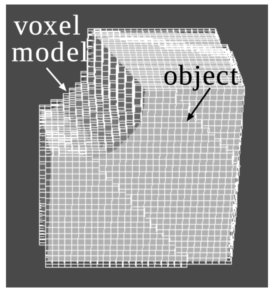


Fig.11: Voxel Model for Force Feedback

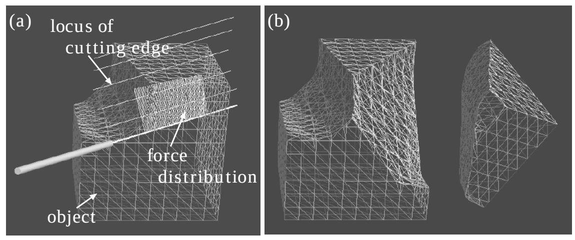


Fig.12: Representation of Cutting Force

An example of the voxel model is shown in Figure 11. Also, an example of cutting operation with force is shown in Figure 12. As is observed in this figure, the distribution of force on the cutting edge is restricted to the part that is colliding with the voxel model. After the operation, the voxel model is deleted.

4. Application to Shape Forming Task

We integrated the algorithms proposed in previous sections into a virtual environment, and experimentally applied the environment for a shape forming task. An example of the process and the result of a user's operation is shown in Figure 13 and 14. The shape of a petal is created from a square panel by deformation. Also, original shapes of the leaf were quarryed from a rectangular object, and they were stretched and flattened so that they look like leaves. The stalk was created in a similar way. Finally, all those elements created above are arranged in a space.

5. Conclusion

We proposed an approach to realize cutting and deforming operations with force feedback. To attain both of the fast update rate of physical simulation for force feedback and the precise representation of geometric shape, we defined coarse physical model and fine geometric model and combined them with each other. Also, by sharing a geometric model in both cutting and deforming operations, it became possible to switch these two operations without the transforming the internal representation of the object.

One of studies to be carried out in the future is to observe how the sensation of force is used in shape forming tasks and to evaluate the contribution of the sensation to the efficiency. The prototype system implemented in this study will provide an environment for this kind of future study.

References

1. Burdea G.: *Force & Touch Feedback for Virtual Reality*, A Wiley-Inter-Science Publication, New York, 1996.
2. Salisbury K., Brock D., Massie T., Swarup N., Zilles C.: Haptic Rendering: Programming Touch Interaction with Virtual Objects; *Proc. Symp. Interactive 3D Graphics*, pp.123-130, 1995.
3. Tanaka A., Hirota K., Kaneko T.: Deforming and Cutting Operation with Force Sensation; *J. Robotics and Mechatronics*, Vol.12, No.3, pp.292-303, 2000.
4. Sederberg T.W., Parry S.R.: Free-Form Deformation of Solid Geometric Models; *SIGGRAPH '86*, pp.151-161, 1986.
5. Yamashita J., Fukui Y.: A Direct Deformation Method; *Proc. VRAIS'93*, pp.499-504, 1993.
6. Norton A., Turk G., Bacon B., Gerth J., Sweeney P.: Animation of fracture by physical modeling; *Visual Computer*, Vol.7, pp.210-219, 1991.
7. Hirota K., Kaneko T.: Virtual Elasticity; *Proc. HCI'99*, pp.1040-1043, 1999.

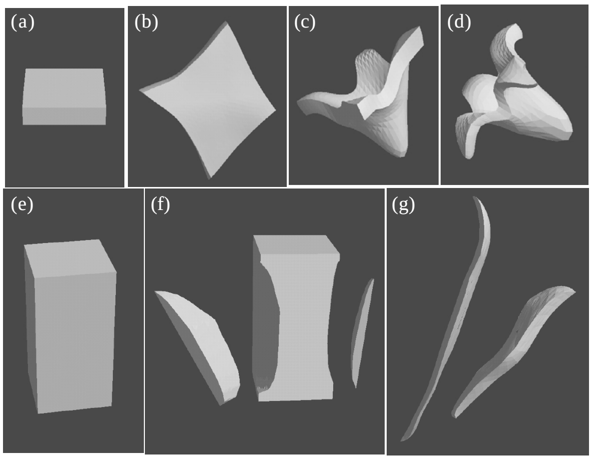


Fig.13: Steps of Forming Shapes

Fig.14: Example of Created Shape

8. Massie T.H.: Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction; Master's Thesis, M.I.T., 1996.

9. Galyean T.A.: Sculpting: An Interactive Volumetric Modeling Technique, *Computer Graphics*, vol.25, no.4, pp.267-274, 1991.

10. Yamamoto K., Ishiguro A., Uchikawa Y.: A Development of Dynamic Deforming Algorithms for 3D Shape Modeling with Generation of Interactive Force Sensation; *Proc. VRAIS '93*, pp.505-511, 1993.

11. Tanaka A., Hirota K., Kaneko T.: Virtual Cutting with Force Feedback; *Proc. VRAIS '98*, pp.71-75, 1998.

12. Hirota K., Tanaka A., Kaneko T.: Representation of Force in Cutting Operation; *Proc. IEEE VR '99*, p.77, IEEE, 1999