Virtual Destruction of a 3D Object with a Stick

Tohru Miyazaki, Toyohisa Kaneko, and Shigeru Kuriyama

Dept. of Information and Computer Sciences
Toyohashi University of Technology
{miyazaki,kaneko,kuriyama}@vcl.ics.tut.ac.jp

Abstract

This paper is concerned with a real-time method for realizing virtual destruction of a 3D object with external force. A target object is a soft $tofu$ (bean curd) cube which is easily destructable or breakable with a stick. A spring-mass network model is used to represent such an object. Destruction is realized by cutting a spring when its length exceeds its maximum length due to excess stretch force. A system was implemented on a PC with 2 CPU's and a PHANToM, a force feedback device. A set of optimal parameters are experimentally identified. It is concluded that real-time destruction realized with the presented method can provide destruction close to real one.

Keywords

Virual Destruction, Spring Network, Real-time Destruction, Bubble Mesh

1 Introduction

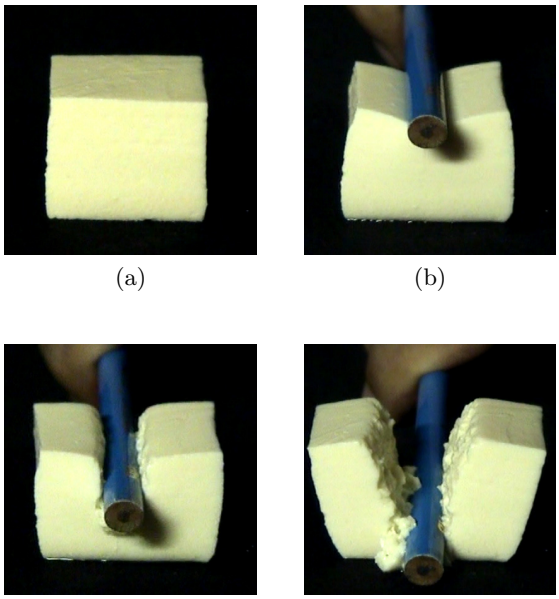
Virtual reality (VR) technology has found its important applications in such areas as medicine (e.g. virtual surgery), amusement (e.g. virtual driving), and manufacturing (e.g. virtual factory). It requires typically real-time interaction devices: high speed CPU, visualization devices (e.g. projector, display), and force feedback devices (e.g. PHANToM).

This paper addresses the problem of realizing virtual real-time destruction of an object with external force. There have been a number of papers concerned with deformation [JP99, DDCB01, SP86, KH96] and cutting [BM02, ML03, MK00, THK98, WO04] in VR, but little works on destruction. For a cutting operation, the affected part on an object is localized only on its touching part of a cutting device (e.g. a knife)[THK98]. However, for destruction, the affected part could be on other parts.

In the area of Computer Graphics where real-time operation is not required, there have been works on deformable objects [TF88a] and fracture [TF88b] using a spring network model. Based upon a stress-strain FEM model, fracture of ductile objects has been treated by O'Brien et al. [OBH02]. A similar work is found on fracture on brittle objects [OH99]. Cracks on drying objects such as clay has been addressed by Hirota et. al. [HTK98].

This paper deals with real-time virtual destruction of soft, brittle objects such as tofu and soft cream cheese with a stick. We employ a PHANToM as a manipulator which enables force feedback with the acquisition capability of 3D position.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.



(a) (b) (c) (d)

Figure 1: Real Destruction of a tofu lump.

2 Method

2.1 Approach

As an initial step, we observed the destruction process of a real tofu cube of 3cm edge length with a stick of 8mm diameter. Figure 1 shows real destruction scenes.

Destruction of a tofu cube is shown in Figure 1. Figure 1(a) shows the original shape. Figure 1(b) shows the initial phase where external force through a stick is placed on the top part of tofu to start sinking down slightly. Figure 1(c) and (d) show subsequent destruction by a half way and completely to the bottom, respectively. The phenomena will be physically simulated with a spring network model as follows.

2.2 Geometric Model and Network

To simulate destruction, we adopt an approach to represent an object with a network of springs. To construct a spring network, an object of interest is decomposed into a set of tetrahedrons with similar sizes. Take a cube of 3cmx3cmx3cm as an example. As shown in Figure 2(a), it is fitted with a close-packed structure where the mutual distance is 1.5mm. (The close-packed structure can pack the largest number of balls or atoms in a fixed volume.) It is analogous to carving a cube out of a closed packed crystal solid. Figure 2(a) shows balls of 2.2mm radius and Figure 2(b) shows the resulting network. The number of balls is 306 on the surface and 294 in the internal domain. The resulting network contains 6994 edges or springs.

The second process is to move all the balls situated within a radius of 1.7mm from the surface to the surface. Then an iterative algorithm called the bubble mesh algorithm[SG95, MIKK04] is executed to align all the balls on the surface so that their mutual distances are as equal as possible (to 3.4mm). Then the same iteration aligns all the internal balls which are located three-dimensionally. The resulting ball alignment and network are shown in Figure 2 (c) and (d), respectively. See reference [SG95, MIKK04] for details of this iterative algorithm.

Since the mutual relationship between the balls (or bubbles) in the close-packed structure is known, the 3D structure resulting from the bubble mesh algorithm can be broken in a set of tetrahedrons, which are the basic building block of a 3D object. Then each tetrahedron is represented with a network of four edges as shown in Figure 2(e). A spring and a damper are allocated to each edge of a tetrahedron as shown in Figure 2(f). Note here only a set of a spring and a damper is assigned to an edge, although an edge is shared by a multiple of tetrahedron edges.

2.3 Kinematics

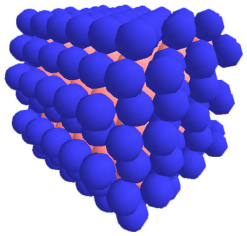
Once an object of interest is represented with a spring network, then the next to be investigated is its kinematics behavior. Consider node i. Its force $\mathbf{f}_i$ is given as:

$$\mathbf{f}_i = m_i \frac{d\mathbf{v}_i}{d\tau} + c_i \mathbf{v}_i \quad (1)$$

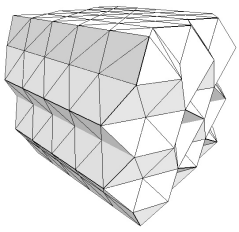
$$c_i = 2\sqrt{k_{max} m_i \alpha^2} \quad (2)$$

where m_i is the mass assigned to node i, $\mathbf{v}_i$ is the velocity vector of node i, τ is a small incremental time slice, c_i is the viscosity of node i, and k_{max} is the maximum spring constant among the springs connected to node i. The damping coefficient α will be treated as a variable whose optimal value is set experimentally (as will be described in Section 3). Equation (1) can be rewritten as:

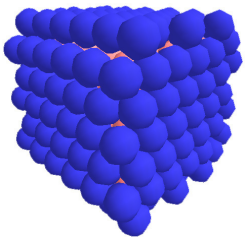
$$\mathbf{v}_i = \mathbf{v}_i{}^p + \left(\mathbf{v}_i{}^p - \frac{\mathbf{f}_i}{c_i}\right) \left\{\exp\left(-\frac{c_i}{m_i}\tau\right) - 1\right\} \quad (3)$$



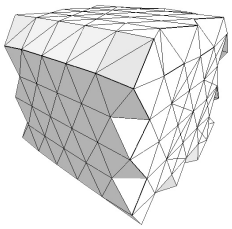
(a) Initial balls with close-packed structure



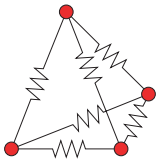
(b) The resulting network



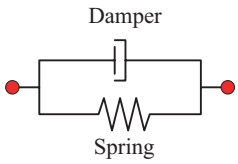
(c) Ball position after bubble-mesh algorithm



(d) The resulting network



(e) Tetrahedron structure



(f) Damper and spring for each edge of a tetraheron

Figure 2: Modeling of a destructive object.

$$\mathbf{x}_i = {\mathbf{x}_i}^p - \frac{m_i}{c_i}\left({\mathbf{v}_i}^p - \frac{\mathbf{f}_i}{c_i}\right) \left\{\exp\left(-\frac{c_i}{m_i}\tau\right) - 1\right\} + \frac{\mathbf{f}_i}{c_i}\tau \quad (4)$$

where ${\mathbf{v}_i}^p$ and ${\mathbf{x}_i}^p$ are the velocity vector and the position vector, respectively, at time $(t-\tau)$. Then $\mathbf{f}_i$, which is the sum of forces at node i from the connected springs from node j, is given as.

$$\mathbf{f}_i = -\sum_j k_{ij}\frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}\left(|\mathbf{x}_i - \mathbf{x}_j| - l_{0ij}\right) \quad (5)$$

where k_{ij} is the spring constant of node i to node j, and l_{0ij} is its natural length.

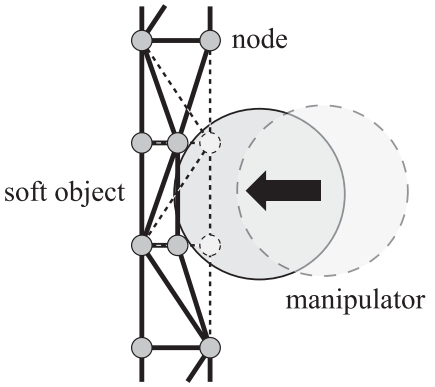


Figure 3: Manipulator position: some object nodes are inside the manipulator.

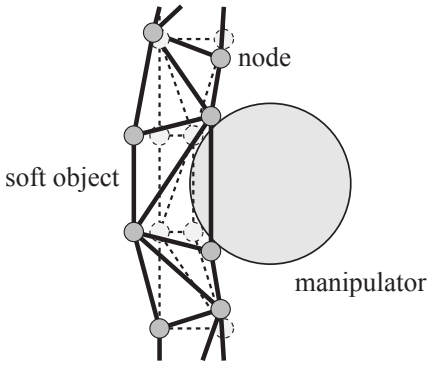


Figure 4: Manipulator position: Nodes are moved to the surface.

2.4 External Force

For the case of placing external force on an object of interest with a manipulator, the first step is to detect the collision between the object and the manipulator. In order to reduce computation time, the collision between nodes and surface polygons is detected rather than that between two sets of surface polygons, which is computationally very time-consuming. While all the surface polygons of the manipulator are considered, only those nodes on the object that collided are considered by identifying them in the following manner. At time T, some nodes shown with dotted circles in Figure 3 which are inside the manipulator can be detected based upon the decision on which sides each node is with respect to each polygon of the manipulator. This decision needs to be carried out by the

number of polygons of the manipulator. Then the nodes located inside are moved along the direction of the manipulator motion and placed on its surface as illustrated with real lines in Figure 3.

The locational displacement of these nodes changes the kinematics balance of the spring network. Equations (3) and (4) are executed iteratively to find the new equilibrium as shown in Figure 4. Here if some edges may be stretched beyond the maximum length allowed, they are cut in the middle as shown in Figure 5.

$$\mathbf{F} = \sum_{i=1}^{n} \mathbf{f}_i \tag{6}$$

The force given to the force feedback device is the average of the reaction forces between the current T and $(T - 4\tau)$ in order to reduce possible noise. Namely $\mathbf{F}_b$ is given as:

$$\mathbf{F}_b = \frac{\sum_{i=0}^{4} \mathbf{F}_{T-i\tau}}{5} \tag{7}$$

where $\mathbf{F}_{T-i\tau}$ is the reaction force at time $(T - i\tau)$.

2.5 Visualization

As was mentioned in section 2.1, the destruction of an object is represented by cutting a set of springs in the spring network model. A spring is cut when its length exceeds its maximum length l_{max}. Actually, it is measured based on the ratio as:

$$l_{max} = l \times l_p \tag{8}$$

As is shown in Figure 5(a), the reaction force is generated at either end of the terminals if $l \leq l_{max}$ holds. For the case $l > l_{max}$, the spring is cut as shown in Figure 5(b). In this case, the reaction force due to the spring at either end of the terminal points becomes null.

When a spring is cut, a new end point is created at the point from each formerly connected end node point by a length of $l_0/2$ as shown in Figure 2(b). Visualization is carried out for each tetrahedron as illustrated in Figure 6 where there are 10 different patterns shown, depending upon which springs are cut.

2.6 Process

The process for virtual destruction mentioned above is summarized as:

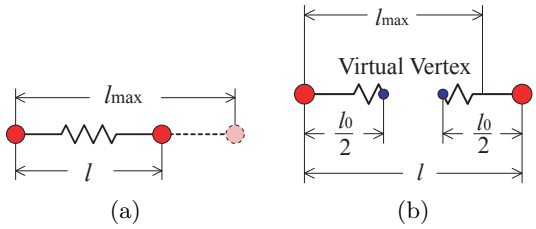


Figure 5: Cutting a spring to realize destruction.

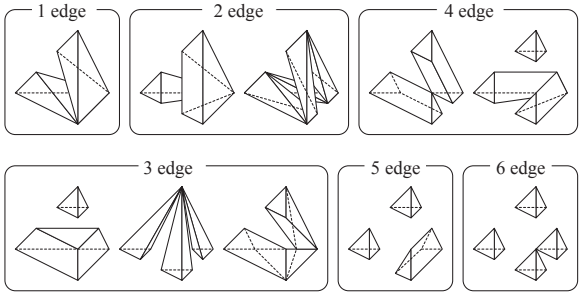


Figure 6: The drawing pattern of a tetrahedron.

(1) The position and direction of the manipulator is updated based on the force feedback.

(2) The collision detection is carried out. If some nodes are inside the manipulator, they are moved back to the manipulator surface.

(3) The reaction force is computed based upon the above node motion. New node positions are iteratively computed using Equations (3) and (4).

(4) If an edge exceeds its maximum length, it is cut in the middle.

(5) The sum of reaction forces from all the nodes on the manipulator surface is fed back to the arm of the force feed device.

The above process must be carried out with a frequency exceeding 300Hz[CDA99]. On the other hand, the visualization speed needs to be in 60Hz.

3 Experiments

3.1 System

A system was implemented on a PC with 2 CPU's (Intel Xeon 2.8GHz) and Microsoft Windows XP. The software was thread-based in order to exploit full power of 2 CPU's. The force feedback device is a PHANToM(SensAble Technologies Inc.). The total VR system is illustrated in Figure 7. The manipulator position can be controlled with the

Figure 7: The situation of execution.

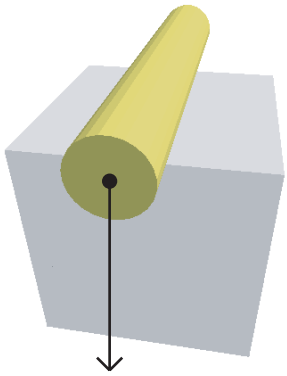


Figure 8: Simulation for deriving appropriate parameters.

PHANToM shown on the right and the visualization result is on the monitor on the left.

Three kinds of manipulator can be selected:(1) a round stick of 8mm diameter, a square stick of 6mmx6mm, and a diagonal stick of 3mmx10mm. The length of each stick is 5cm long.

The computation time per cycle (which is equal to τ in Equations (3) and (4)) is proportional to the number of edges in the spring network. It was found that it is 2.9 msec. with 8000 edges, resulting to a frequency of 350Hz.

3.2 Optimal parameters

For destruction of a tofu cube, it is difficult to measure parameters such as spring constant k, viscosity α, and maximum length ratio l_p without specially designed equipments. Therefore, we estimated these parameters experimentally by observing the way of destruction visually.

For conducting these experiments, we employed a cube of 3cmx3cmx3cm as an object and a stick of 8mm diameter as the manipulator. The same procedure described in section 2.2 was applied to get a spring network where the node lengths are reasonably similar. The weight of the object is 30 grams. The manipulator was moved downward with a speed of 1mm/sec. with a control from the application rather than manual operation in order to avoid unnecessary human errors. 1.5 and 3.0 are set as dynamic and static friction, respectively in order to account for the friction between tofu and the manipulator surface.

Experiments for 500 variations were carried out:(1) ten variations in spring constant k from 10 to 100 with an increment of 10, (2) ten variations in viscosity α from 5 to 50 with an increment of 5, and (3) five variations in maximum length ratio l_p from 1.1 to 1.5 with an increment of .1.

Table 1: Parameter applied to a system.

name	value
spring limit length ratio	1.3
spring coefficient	30
damper coefficient (α)	30
dynamic friction coefficient	1.5
static friction coefficient	3.0

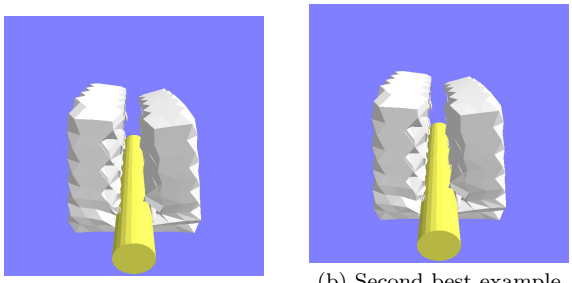
The 500 results were rated according to their visual similarity with the actual (see Section 2). Figure 9 shows the best, second best, and two failed examples and their parameters.

Figure 10 shows a temporal change of the total downward reaction force on the manipulator. The downward reaction force generally increases as the manipulator is pressed more downwardly. This trend of reaction force agrees with the actual case.

3.3 Experimental Results

Based upon the parameters selected in the above, experiments were conducted. In this experiment, a stick is operated by human. The object is a hexahedron with a dimension of 3cmx3cmx1cm, and a weight of 11grams. The bubble mesh algorithm was executed with the initial close-packed structure with a radius of 1.5mm. The total number of balls was 382 on the surface and 314 internally. And there are 7954 edges in the resulting spring network. A stick of 8mm diameter and 5cm length was used as the manipulator.

Figure 11 shows four phases of destruction sim-



(a) Best example where $k = 30, \alpha = 30, l_p = 1.3$

(b) Second best example where $k = 50, \alpha = 30, l_p = 1.3$

(c) Failed example due to oscillation where $k = 90, \alpha = 30, l_p = 1.3$

(d) Failed example due to excess pull force where $k = 60, \alpha = 30, l_p = 1.5$

Figure 9: Simulation results

ilar to the real destruction shown in Figure 1. Figure 11(a) is the initial state of the cube, (b) shows the manipulator to start pressing the cube slightly, and (c) shows the phase that the manipulator moved downward about a half of the vertical length, and (d) shows the final phase that the manipulator was pressed to the bottom. The similarity between the four phase figures in Figure 1 and Figure 11 is reasonably good.

Figure 12 shows a temporal change of the total reaction force on the manipulator where the force along X, Y, and Z axis is indicated by thin dotted line, real line, and dense dotted line, respectively. The X, Y, and Y axis represent left-right, downward, and forward-back direction of the object, tofu, shown in Figure 11. The downward trend (Y-axis) agrees with that of Figure 10.

As an example of freeform objects, a tofu globe of 2cm radius was employed. Balls of 2.2mm radius are packed in the globe for the bubble mesh algorithm. The resulting globe contains 370 balls on the surface and 305 in the internal domain. The resulting network contains 8041 edges. The total weight is set to be 36grams. This globe was destructed with a stick of 8mm radius with 10cm length. The resulting destruction is shown in Figure 13. It was judged that this destruction is reasonably close to the real one, which is not shown here, though.

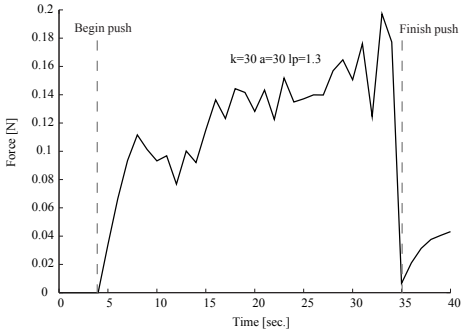


Figure 10: Reaction force along the downward direction.

It is seen that real-time virtual destruction is possible using a force feedback device like PHANToM.

4 Conclusion and Future Work

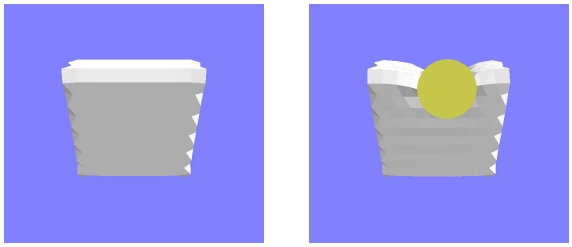
It has been shown that virtual real-time destruction of objects is possible by operating a manipulator. It utilizes a spring network model, where destruction is represented by cutting edges. Visualization is carried out on the basis of tetrahedrons, the basic unit of objects. Appropriate parameters such as spring constant and viscosity were selected experimentally by observing the manner of destruction with varying these parameters. A system with 2 CPU's and a force feedback device was implemented and its real-time operability was demonstrated.

Future works include treating more freeform objects of various sizes.

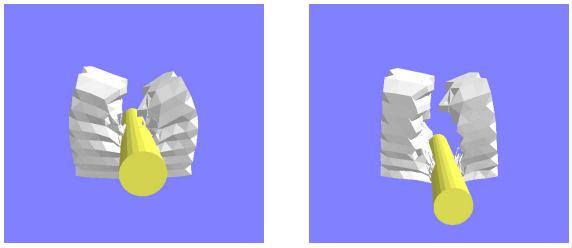
References

[BM02] C. D. Bruyns and K. Montgomery. Generalized interactions using virtual tools within the spring framework: Cutting. *Medicine Meets Virtual Reality*, January 2002.

[CDA99] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization*



(a) Initial object state (b) Starting position



(c)The stick pressed downward about a quater (d)The stick pressed downward almost totally

Figure 11: Simulation results.

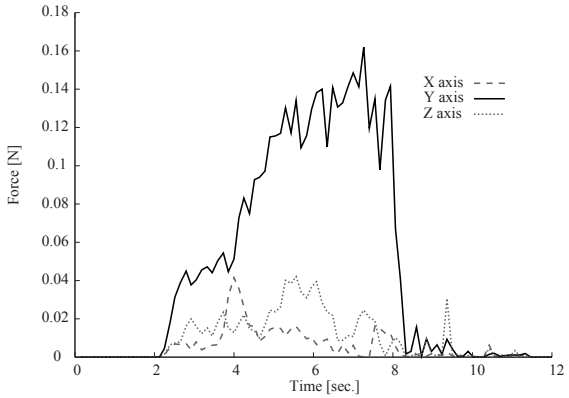


Figure 12: Temporal change of the reaction force along three diections

and Computer Graphics, 5(1):62–73, January-March 1999.

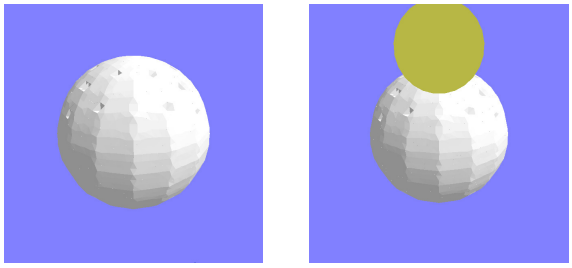
[DDCB01] G. Debunne, M. Desbrun, M. P. Cani, and A. H. Barr. Dynamic real-time deformations using space and time adaptive sampling. *ACM SIGGRAPH 2001*, pages 31–36, August 2001.

[HTK98] K. Hirota, Y. Tanoue, and T. Kaneko. Generation of crack patterns with a physical model. *The Visual Computer*, 14:126–137, 1998.

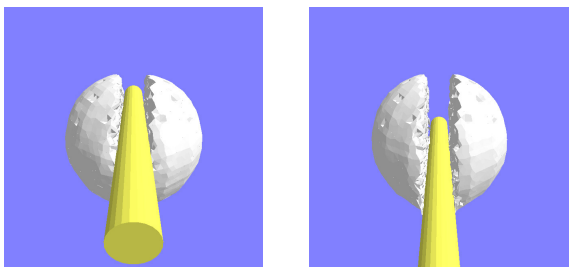
[JP99] D. L. James and D. K. Pai. Artdefo: Accurate real time deformable objects. *ACM SIGGRAPH'99*, pages 65–72, August 1999.

[KH96] T. Kaneko and K. Hirota. Simulation of surgical operation onto soft and transforming tissues. *Proc. VSMM'96*, pages 283–287, 1996.

[MIKK04] I. Mizuno, Y. Iwasaki, T. Kaneko, and S. Kuriyama. Volume preserving deformation of 3d elastic objects by external force. *IEICE Tr. o Information and Systems(Japanese Edition)*, J87-D-II(6):1319–1328, June 2004.



(a) (b)



(c) (d)

Figure 13: Four phases of a sphere destruction.

[MK00] A. B. Mor and T. Kanade. Modifying soft tissue models: Progressive cutting with minimal new element creation. *Medical Image Computing and Computer-Assisted Intervention*, 1935:598–607, October 2000.

[ML03] C. Mendoza and C. Laugier. Simulating cutting in surgery applications using haptics and finite element models. *IEEE Virtual Reality*, 2003.

[OBH02] J.F. O'Brien, A.W. Bargteil, and J.K. Hodgins. Graphical modeling and animation of ductile fracture. *ACM SIGGRAPH 2002*, pages 291–294, July 2002.

[OH99] J.F. O'Brien and J.K. Hodgins. Graphical modeling and animation of brittle fracture. *ACM SIGGRAPH 1999*, pages 137–146, August 1999.

[SG95] K. Shimada and D. C. Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. *Proc. Solid Modeling and Applications*, pages 409–419, October 1995.

[SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *ACM SIGGRAPH'86*, 20(4):151–160, August 1986.

[TF88a] D. Terzopoulos and K. Fleischer. Deformable models. *The Visula Computer*, 1:306–331, 1988.

[TF88b] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *ACM SIGGRAPH 1988*, pages 269–278, 1988.

[THK98] A. Tanaka, K. Hirota, and T. Kaneko. Virtual environment for cutting operation with force feedback. *Proc. VSMM'98*, 1:164–169, 1998.

[WO04] D. J. Weiss and A. M. Okamura. Haptic rendering of tissue cutting with scissors. *Medicine Meets Virtual Reality*, Janualy 2004.