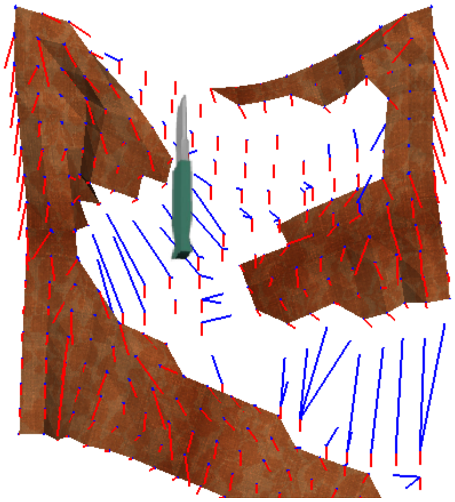
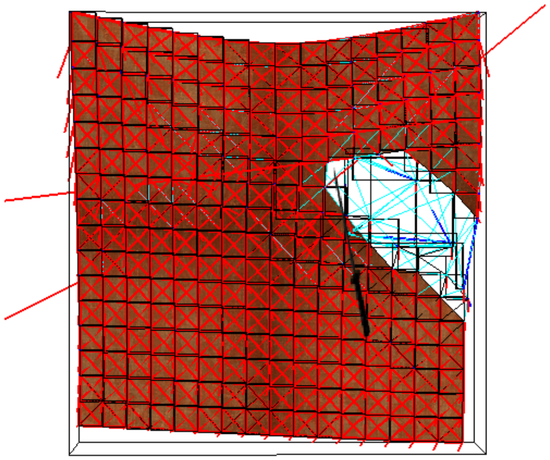
Interactive Cloth Cutting Simulation

Devon Connell Derek Connell





Abstract

Our aim for this project was to create a simulation which was interactive as well as entertaining. The cloth spring system is based off of Provot’s Mass-Spring model, and an actual 3d mesh is used to simulate cuts and splits. We strived for realism, but wanted the user to maintain full control. There is a cutting “vertex” on the blade which, when it comes into contact with a cloth edge, will seek out the nearest cloth vertex and delete all appropriate structural, bend, and shear spring relationships. Pieces of cloth without any connection to a fixed point will fall down due to gravity.

1. Motivation

We were very impressed by “Teddy: A Sketching Interface” (Igarashi et al.). The results of the paper could be felt instantly through an interactive demo. Interactive applications are a very engaging medium with great potential for presenting methods, algorithms, and theories. Unlike a selected render or movie clip, an interactive program cannot be pre-picked to show off the “best cases”. There is an exposure in leaving your simulation in the hands of the user, where the good, the bad, and the ugly can (and will) become apparent. Some of those who used the Teddy program loved it, others pointed out its flaws and limitations. But all were deeply involved in their opinion. This is the effect we wish to bestow on our peers and the readers of our work. Perhaps they’ll be inspired and wish to further what we’ve accomplished in this paper.

2. Previous Work

Our project is based on previous work in cloth spring-mass models and on blade sharpness.

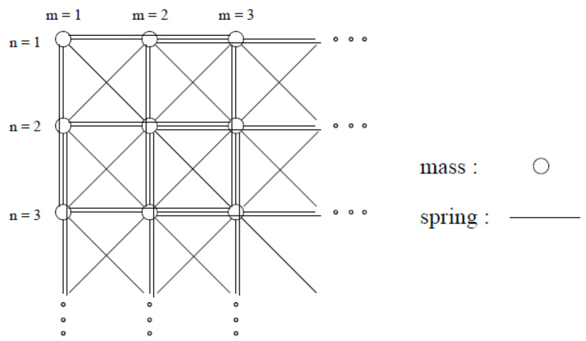


Figure 1: Traditional Provot mass-spring model†

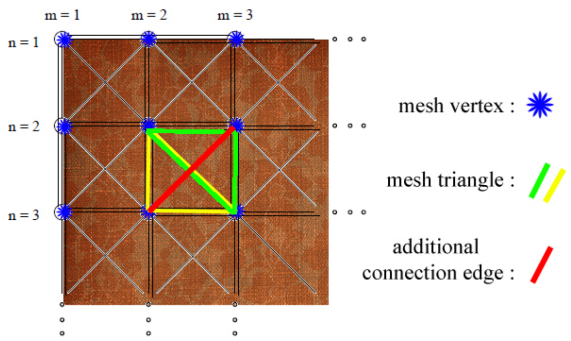


Figure 2: Textured 3d Mesh with aligned vertices

2.1 Mass-Spring Model

The cloth representation is based off Provot's mass-spring model. It's an elastic model made of m x n virtual masses (Figure 1), with mass-less springs linking each together. These links are either structural, shear, or bend springs.

However in the mass-spring model, edge deletion from any kind of cut could not create empty spaces. Spring relationships could be deleted, but that would just cause an unrealistic stretching of the model. To create the visual appearance of cut cloth a traditional 3d mesh must be overlaid on top of this system. Figure 2 illustrates how we create the mesh from the initial cloth particles used in Provot's model. Note the additional 'shear' edge added to the triangles. This is necessary to allow one-to-one relationships between a vertex and cloth particle. Each vertex has a pointer to its corresponding cloth particle and vice versa. The shear edge allows a vertex to have mesh edges in the same manner a cloth particle has its structural and shear connections. For those unfamiliar with Provot's spring relationships, see Figure 3. For Bend connections, we assume if the structural spring in a direction from any cloth particle is cut, so too is it's bend in the same direction.

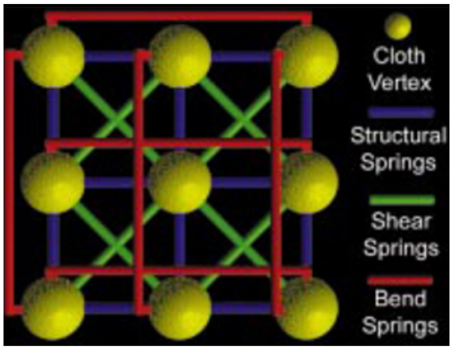


Figure 3: Provot model spring relationships‡

2.1 Sharpness Parameters

In our effort to model realism we looked to a medical paper on blade sharpness (McCarthy et al.). This issue of sharpness is of importance for precision tools such as scalpels, which is the primary blade for our simulations. The paper states an exact definition for "sharpness" as the amount of force necessary to penetrate through a certain distance in a material. We used this understanding of sharpness and the findings in Figure 4 for parameters in our simulation. A very sharp instrument would be able to cut cloth cleanly and with a minimum amount of force, where a more duller blade

Type of Blade	Typical sharpness levels (N)
Razor blade	0.3 - 0.4
Scalpel blade	0.4 - 0.6
Utility razor blade	1.0 - 1.4
Kitchen knife (new)	1.8 - 3.0
Hunting knife	1.9 - 3.5
Well worn blade	6.0 - 11.0

Figure 4: Blade sharpness parameters

would require a stronger force and notably affect neighboring particles. We modeled this by changing the force that is applied to a cloth particle during a cut depending on the blade parameter used.

3. Interactivity

We made the interface a first priority with this project. We wanted the user's mouse to represent an actual blade for simulation. However, a mesh on mesh collision detection or ray casting scheme proved too involved and processing intensive for our purposes. To prevent slowdown, we chose to use a 2d image of a blade and replace it with the mouse cursor. When the left button is clicked, the picture changes to a different image of the blade pointed down. This gives a satisfying feeling when holding down and "slicing". A 2d spot on the screen near the mouse pointer is converted into 3d space. This allows for fast edge cutting detection calculations. The blade texture is blended on top of the cloth texture, allowing for transparency.

By use of constantly updated bounding boxes for each cloth square, we are able to easily find which region the user has chosen to cut. Our program creates a map of selected bounding box cloth particles with their Z distance (depth). Using the x and y position of the mouse and the current bounding box it's in, we can find the nearest cloth particle (our target) as well as which edge and direction to cut along. To find our target cloth particle we need only find the shortest distance from the selected spot to the corner particles. Edge and direction is computed by simply taking two spot instances of the mouse in the bounding box. Based on the x and y differences of the spots we are able to calculate if it's a horizontal, vertical, or diagonal cut, and its direction based on a positive or negative result.

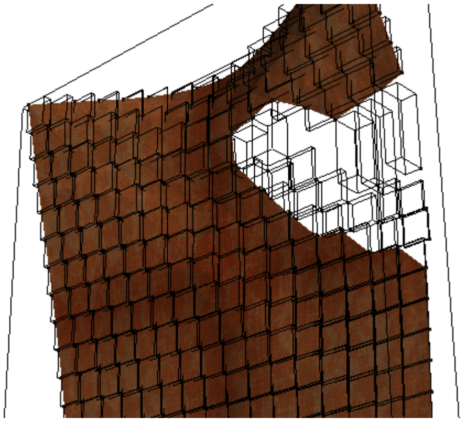


Figure 5: Cloth particle bounding boxes

4. Algorithms

After the cloth particle, the edge, and the direction of a cut is found, we can proceed to remove the corresponding mesh edge. Then we proceed to remove the corresponding structural or shear spring connections. There must be a check to see if a hole should become visible on the mesh. We use a map of vertices to booleans, all initially set to false. We then perform the following procedure:

Mesh triangle deletion algorithm

Iterate through all the triangles of the mesh

- Set it's vertices in the map to true
- Check if any two edges of a triangle are deleted.
    - If they are, remove the triangle and delete all structural and shear relationships to the vertex that had both of its adjacent edges in the triangle deleted. This vertex's Boolean **completeCUT** is set to true.
    - To help improve accuracy and ensure cleanly cut connections we also remove adjacent triangles by erasing those which contain a vertex that is **completeCUT**. We remove the corresponding shear and structural connections.
- Afterwards, we traverse the map and for any vertex still set to false (that is, not in any triangles) we remove every other cloth particle's connections with it.

For cloth animation, we must first ensure that the mesh vertices are moving along with their corresponding cloth particles. If they're part of a triangle, then they will show accurate movement of cloth. To maintain accuracy we constantly update each mesh vertex position to that of its cloth particle.

To animate cloth, we make use of Provot's techniques. Here is a short summary of this animation process is: go through each cloth particle, calculate the forces due to gravity and from each structural, shear and bend spring connections. By using provot's correctional spring technique, there is a threshold on how far apart two cloth particles can be. However, because our cutting doesn't actually remove cloth particles this prevents realistic movement due to cutting.

We remedy this by checking if the connected cloth particle's vertex is a **completeCUT**. If this is the case, don't apply forces or Provot's adjustments from that connection. As a final check, make sure vertices that have no structural or shear connections are set to **completeCUT**.

5. Conclusions

We set out to create a user-interactive cloth cutting simulation with some degree of realism, and we believe we've accomplished that goal. Due to time constraints, we were unable to implement an involved edge detection system and interface. Future work would involve mesh refinement when an edge is cut, creating and altering nearby triangles for greater accuracy. The greatest hindrance we found to this is the underlying cloth particle system and keeping the simulation real-time. Derek worked on the interface and sharpness parameters. Devon work on edge removal and corresponding spring deletion.

6. References

[1] “Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior” Provot, 1995.

[2] “An Investigation into Scalpel Blade Sharpness using Cutting Experiments and Finite Element Analysis” McCarthy et al, 2005.

[3] "Teddy: A Sketching Interface for 3D Freeform Design", Igarashi et al., SIGGRAPH 1999

†Image taken from Provot ‡Image taken from Lander (http:www.darwin3d.com)