Cloth Tearing Simulation

Emre Onal*, Veysi Isler

Abstract

Among different physical simulation topics, cloth simulation is one of the most popular subjects in computer graphics. There are many different studies published on different aspects of cloth simulation, but there are not many studies focused on the tearing of cloth. Existing studies related to this topic have only dealt with some aspects of the problem [Metaaphanon et al. 2009] and have not provided general solutions. In this study, we provide a generic solution for different aspects of the problem of tearing cloth.

Some of the points we focus on in this study include: providing realistic tearing effect, preserving polygonal area consistency and texture integrity after the process of tearing, and handling the tear properly for either outer physical impacts or inner manipulations like allowing a user to drag the cloth interactively.

The technique proposed in this paper works with non-uniform cloth structures. It makes it easier to adopt the solution proposed here for many different simulation systems. The processing cost of the technique is quite small, so it is also appropriate for real-time simulation systems.

CR Categories: Computing Methodologies [Computer Graphics]: Animation—Physical Simulation;

Keywords: Cloth Tearing, Cloth Simulation, Spring Physics, Fracturing materials

Links:

1 Introduction

The motion of cloth has unique features when compared to the motion of other objects in the real environment. That is why cloth animation is a topic treated as a different area on computer graphics separately. There are many different studies on this topic [de Aguiar et al. 2010] [Provot 1995] [Bridson et al. 2002], but most of them are related to movement of cloth, interaction with other objects and performance optimizations. There are not enough studies related to the tearing of cloth. There are existing studies on fracturing materials, but since the nature of cloth physics differs from other kinds of objects, such methods are not directly applicable to handling the tearing of cloth.

There are different methods used for modelling cloth in simulations [Ng and Grimsdale 1996] [Nealen et al. 2006]. They have different advantages and drawbacks. We review different aspects of these methods and extend them to constitute the required changes to handle tearing in the cloth structure. This may result from an outer physical effect or directly by a user interaction.

Detecting where and when a rupture will occur, finding the path that the tear will propagate and making the related structural changes constitute the main parts of the problem. At the same time, preserving texture integrity after the tearing is also important.

* e-mail: emrek1@gmail.com

Another problem is to find a solution which could work on non-uniform cloth structures. It is a more generic solution for cloth modelling and also provides performance gain by allowing different resolutions on different parts of a cloth. We developed a model which uses triangular meshes and spring constraints in traditional continuum sheet model together.

Damaged and undamaged parts of the cloth respond to an impact in different ways. While triggering a tear on an undamaged region, we face strong resistance, whereas damaged parts of a cloth show less resistance against tearing. For some materials with unique characteristic features, this could be different. However, this is the case for a majority of cloth-like materials. We take into consideration these issues and apply different constraints according to the condition of the cloth to reflect a more realistic result.

2 Background and Related Work

2.1 Background

Continuum sheet model with springs on a regular grid [Provot 1995] is a well-known and widely used cloth model. It has some constraints to simulate realistic cloth behaviour.

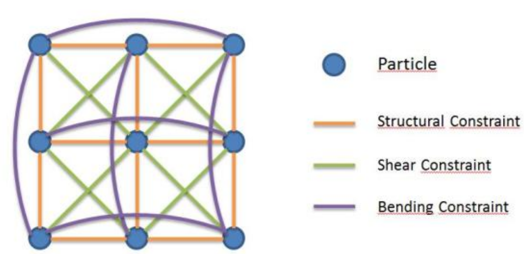

Figure 1: Structural, shear and bending constraints on the cloth model.

This model gives visually satisfactory results, but it is not designed to handle structural changes in the cloth like tearing. The connections between particles are not designed to be separated. Also, in this model all parts of cloth need to be modelled with the same resolution. There are some other studies which uses irregular triangular meshes [Baraff and Witkin 1998] [Narain et al. 2012]. Using irregular triangles makes it possible to model different parts of the cloth with different resolutions. The main advantage of this approach is the performance gain.

2.2 Related Work

There are not many studies directly related with the tearing of cloth. In an earlier study, Terzopoulos and Fleischer [Terzopoulos and Fleischer 1988] proposed some methods on modelling inelastic deformation. In one of the examples they showed fracture propagation on surfaces with a net falling over an obstacle. The fibers on the cloth were subject to fracture limits. The fibers were broken when the ball fell onto the obstacle and the net was torn, but this model was not a cloth with a polygonal surface.

Metaaphanon et al. [Metaaphanon et al. 2009] worked on the cloth tearing problem but in a specific condition for woven clothes. They used both the standard continuum sheet model and a yarn-level model. First, the cloth was completely modeled according to the standard continuum sheet model, then the area around the torn line was modeled according to the yarn-level model, and tearing occurred in this yarn-level model. Their main focus was to simulate the behavior of threads on the torn lines with their yarn-level model. In many studies related to cloth, the tearing of cloth is mentioned as a future study [Jain et al. 2005] [Haggstr¨om¨ 2009], but there is yet to appear a complete study that proposes a successful generic solution for tearing in cloth.

Hellrung et al. [Hellrung et al. 2009] designed a system for cracking and shattering objects. Zhaosheng Bao and Jeong-Mo Hong [Bao et al. 2007] proposed an algorithm to handle the fracture of stiff and brittle materials in which the objects are treated as rigid bodies, but such studies are not directly applicable to cloth tearing because of the difference between the nature of the cloth and the rigid materials. O'Brien and Hodgins had some studies on crack initiation and propagation [O'Brien and Hodgins 1999], and they evolved the existing techniques used for simulating flexible objects. That study was about fracturing brittle materials only and was not applicable to cloth-like materials like the other studies mentioned above, but this method was also extended to support ductile fractures by adding a plasticity model to the former finite-element method used in their former study [O'Brien et al. 2002]. Fracturing a brittle material cannot be used to represent a cloth tear, but it would be possible to use a ductile object fracture to look like a cloth tear by configuring the material properties, however this will not be appropriate for a whole cloth simulation system.

3 Methodology

3.1 Cloth Model

In our cloth model we use springs and irregular triangular meshes. Our method relies heavily on data relation and benefit from this while reconstructing the cloth structure during rupture. The data structures for particles, springs and triangles hold data related to each other. A spring has the data of its neighbour triangles and a particle has the data of the springs that is connected to itself. A triangle holds texture positions in addition to springs and particles constructing itself. These data are used in different steps like; searching the tear path, reconstructing the cloth structure, maintaining texture integrity after tearing, calculating an interpolated normal vector for a particle which is used for lighting by searching the surrounding triangles, and detecting the proper particles that will be used for bending and shear constraints.

Since we change the structure to a non-uniform model, we cannot use the shear and bending constraints as they are in the standard continuum sheet model. The two ends of bending and shear springs are calculated using indices on the grid, but in our structure we do not have a regular grid and indices anymore. Also, springs crossing each other irregularly would make it harder to process the segmentation of regions on the cloth. Not to sacrifice the visual quality maintained by those constraints, we developed a different approach for them.

Kelager et al. published a study about a triangle bending constraint model [Kelager et al. 2010] in which the difference between the normals of two adjacent triangles were used to determine the bending constraint between those triangles. The nature of that study is appropriate for our triangle model since we know the two neighbour triangles for any spring, and we also know the surrounding spring vectors for any triangle. However, we found a better solution that would meet our needs. Since we know the neighbour triangles for a spring, we can reach the opposite corners of those triangles. We create a spring connected to these two opposite corners like the spring shown with the red dotted line in Figure 2. The repulsion force of this spring meets the bending constraint between the two triangles. In Figure 2, this bend-shear spring applies a repulsion force until it reaches the length it has when the two triangles are on the same plane. An additional benefit of this spring is that the attraction force of this spring also meets the shear constraint. This way we are able to meet the two constraints at the same time with a low cost.

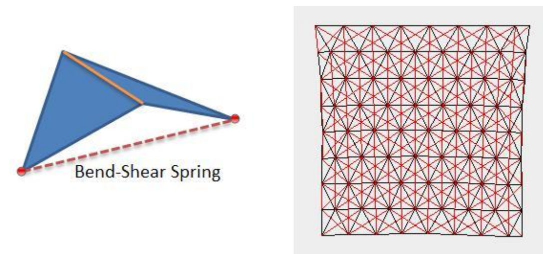

Figure 2: Bend-shear springs.

The non-uniform placement of particles makes it difficult to find the right particles to apply bending and shear constraints. Also, it is not possible to detect the springs to eliminate when a rupture occurs and configure them again according to the newly changed structure. In our solution, we relate this bend-shear spring with the spring on the intersection edge of the two triangles, as on the orange spring in Figure 2 on the left. This way we can keep track of these bend-shear springs and eliminate and recreate them when needed after a tearing occurred. On the right we see all the bend-shear springs in a cloth drawn with red, and the structural springs drawn with black.

Bend shear springs are created dynamically during the tearing. We use the length of the surrounding springs of the two adjacent triangles and the cosine rule to be able to calculate the length of a bend-shear spring we create between these two triangles. The length is the distance between the tips of the two adjacent triangles when they reside on a plane with no stress.

3.2 Tearing

3.2.1 Basic Tearing

During the simulation, as a result of the movement of the particles, the length of the springs change. We assume that when the length of the spring exceeds a certain threshold, the cloth would need to be ruptured around that area. In Figure 3, the cloth is dragged through the blue point and there is tension on the red line. After we stretched a little bit more, the deformation threshold is exceeded and the cloth is torn on that line.

Here the neighbour triangles of the stretched spring are eliminated and new triangles are created in place of them by dividing the former triangles into two.

We prepare new springs according to the new triangles and use them for the creation of those new triangles. In Figure 4 the red springs are the remainder of the spring, which is subjected to high

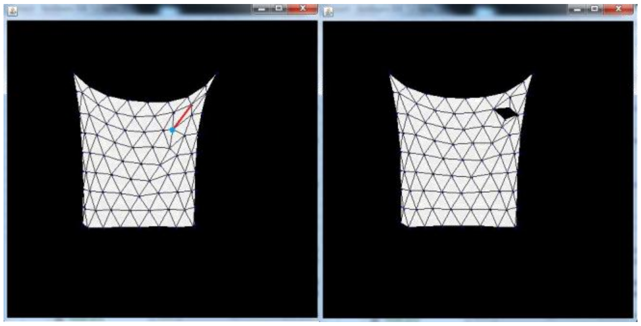

Figure 3: Tension on the cloth and the basic tear.

tension before the tearing. The green springs divide the former triangles into two. The two facing couples of green springs are on the same line of the texture of the cloth, but we need two separate springs for each texture line at those torn edges because they should not be connected anymore. That means both of the triangles on the opposite side of the tear should have their own springs so as not to be connected to the triangle on the other side of the tear line. In Figure 4, since m3 and m4 do not share an adjacent spring, they can move independently. On the other hand, since m3 and m5 share the same red spring, they are connected and they do not move independently.

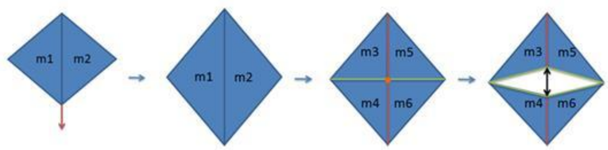

Figure 4: Basic tear.

There is a small calibration that needs to be mentioned. In this example, the spring on the intersection edge is elongated and ruptured because the length of it exceeded the tearing threshold. We create two new springs with the half length of the original spring indicated as red springs in the middle of Figure 4. If we do not change the positions of the particles at the middle of the tear at the orange point, the newly created springs will also be created and elongated more than the tearing threshold, so they will be ruptured again and again. To prevent this effect, we need to shorten the length of those springs so we move the tips at the middle of the tear closer to the other edge of the spring and create it that way so the new length will not be long enough to be ruptured again.

3.2.2 Weak Points

When you start to tear a cloth, for most kind of materials, the ruptured parts become weaker. When tension increases around these weak points, the tear tends to continue through these ripped parts and make the hole larger, rather than creating another hole next to the former one.

In Figure 5, when there is high tension on a string which is connected to a weak point, our former method would create another hole near the first, whereas in real life examples the tear would continue along the same tear line. Cloth can show less resistance at these weak points, and a tension which is not strong enough to tear an undamaged cloth would be enough to rupture a weak point, so we apply a smaller tearing threshold for springs that

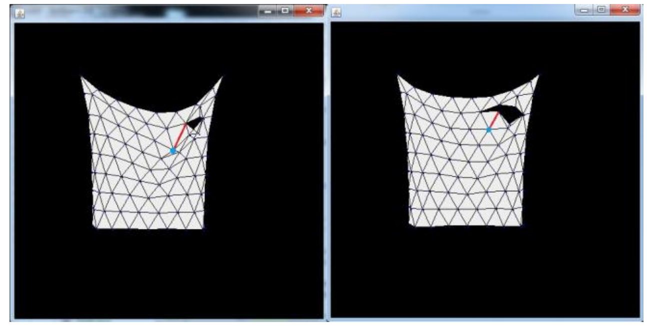

Figure 5: Tension and tear around a weak point.

are connected to weak points. Here, when the tension on the red line exceeds a certain threshold, the tearing will occur, but since one of the ends of this spring is on a weak point this time, the neighbour triangles of the stretched spring will not be divided into two as in the first basic method in Figure 4. Instead, we select one of the springs connected to this weak point and detach the two neighbour triangles.

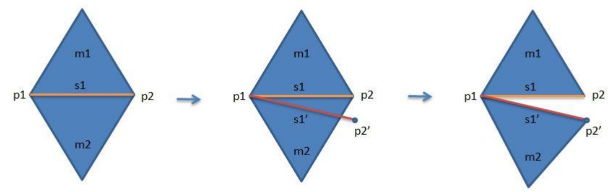

Figure 6: Detaching the triangles connected to the weak point p2.

Figure 6 demonstrates the recreation of structures after the spring selected for the tear path.

An important problem at this step is to determine the tear path. For example, in the case illustrated in Figure 7, there is tension on the red spring and there are eight triangles connected to the weak point, which is under pressure. Two of them reside on a torn edge, indicated as green on the figure. One of the springs is the spring with the tension, indicated as red. There are five springs left that the tear may continue to grow on.

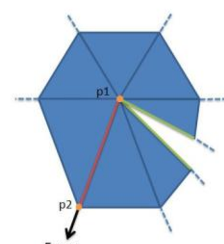

Figure 7: Possible paths for a tear on a weak point.

Here the selection of the spring is the problem. There is not a unique solution at this point, and the result may differ according to the characteristics of the material of the cloth. We propose a heuristic, which is similar to most of the cases we see in the real world and produces visually pleasing results.

The main idea in our solution is to find the closest spring to the perpendicular axis of the force direction on the right side. The idea is to find the path which maximizes the torque. We can observe this in Figure 8. The tension on the red spring is increased and so the weak point in the middle is pulled through the red spring. In the first step, we find the spring that shows the highest resistance against the force. We find it by calculating the projection vectors of spring forces on the force axis and detecting the highest one on the opposite direction of the force.

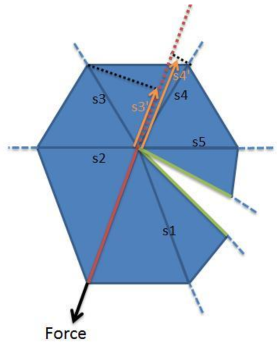

Figure 8: The projection of spring forces on the axis of tension.

Here s3' and s4' are the projections of force vectors of the springs s3 and s4. We did not show all of them for sake of clarity. For this example, we find that s4 is the spring that has the highest resistance against the tension on the red spring.

After this step, the remaining springs are divided into two groups. s1 and s5 are on the right side of the tension, s2 and s3 are on the left side of the tension. Since the tear connected to the weak point is on the right side of the tension, s1 and s5 are not under stress, tearing one of them is not logical, so we need to select either s2 and s3 for the tear path. We find the correct side by checking the connectivity between triangles. The red spring and s4 are not connected through the triangles on the right side, but they are connected through s2 and s3 on the left side.

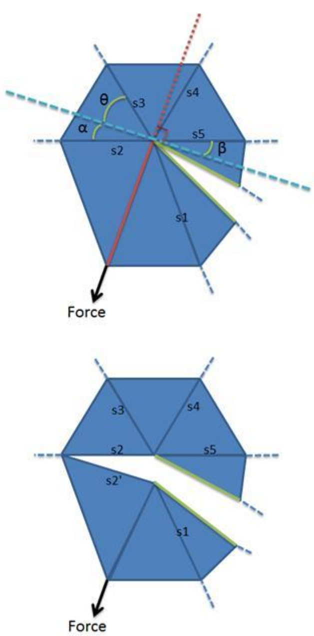

Figure 9: The angles between the spring forces and the axis perpendicular to tension axis.

After detecting the right springs to control for tearing, we found that they are s2 and s3. Our aim is to find the one which has the smaller angle between its force vector and the axis perpendicular to the tension axis, indicated as the blue dotted line in Figure 9. For this example, between s2 and s3, s2 has the smaller angle, so we choose it for the tear path. The angle between the force vector of s5 and the axis perpendicular to the tension axis is shown as here. Even though it could be smaller than this, we do not take it into consideration since it is on other side of the tension axis.

After a weak point is ruptured and as the tear continues on a weak path, the cloth structure and weak points change. For Figure 6, after the triangles connected to the weak point, p2, are detached, p2 or p2' are not weak anymore. The change of weak points can be seen in Figure 10, and are shown as blue points on the wire-frame view.

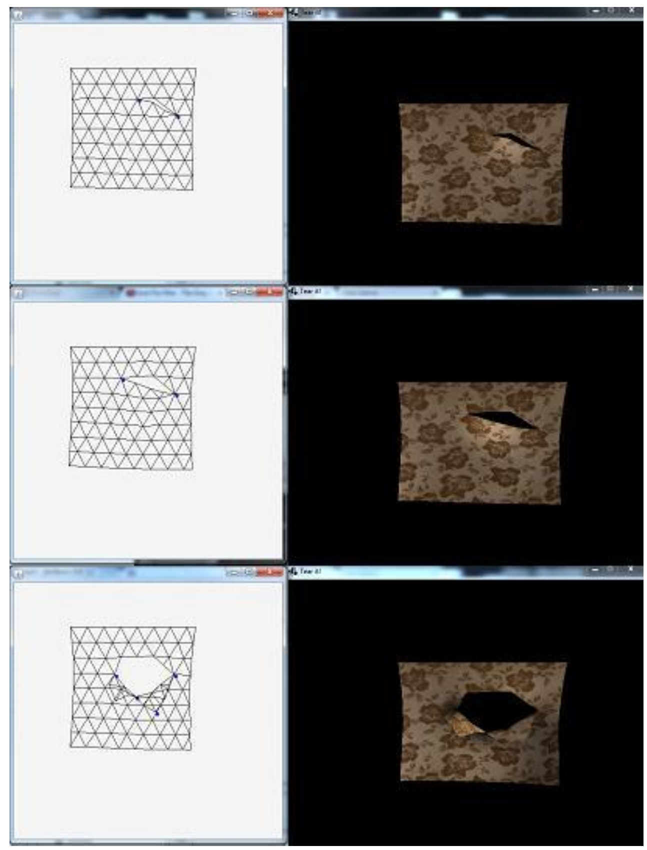

Figure 10: Weak point propagation.

3.2.3 One Point Connections

There is an exceptional case which needs to be addressed. In some cases, the connections between two different parts of the cloth may consist of only one point. The tear would have come to the edge of the cloth or two different tears may have come across at a point. Here that point is a weak point, but the algorithm we use for weak points searches for a spring to continue the tear. However, in this case the opposite parts of the cloth on the tension axis should have been detached directly without considering the springs.

We detect the situation by comparing the number of the springs which have a connection to the spring with the tension and the total number of springs connected to the point of interest. We create a new particle, p' instead of p, and apply a small displacement along

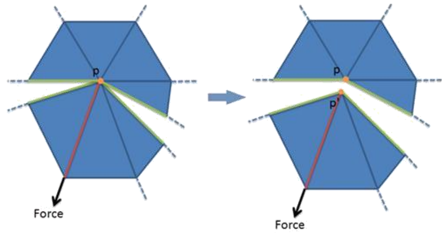

Figure 11: Tear on a one point connection.

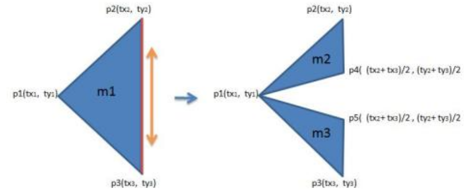

Figure 13: Texture position assignment after dividing a triangle.

Figure 14: Textured, torn cloth.

the axis of tension to be able to prevent the tear after the detachment by shortening the length of the spring with the tension. We use the same threshold as we use for weak points in one point connection cases, but it is possible to use a smaller threshold here. This is also a subject which could change according to the characteristics of the cloth material.

3.3 Texture Integrity

Another important part of the problem is to preserve texture integrity after tearing. In a continuum sheet model, particles are placed on the corners of a regular grid and cloth texture is applied to this plane. As the simulation runs, the positions of these particles change. These positions are used while drawing the polygons. In order to draw the correct texture portion on a polygon, we have to know the texture coordinates at the corners of that polygon. In a sheet model which uses a regular grid, it is possible to reach the texture coordinate on a particle using the indices. Since we use non-uniform structure we cannot use such an approach. Also, it is not suitable for a system in which the model is subject to structural change.

At this point, we take advantage of the data stored in triangle structure. At the initialization step, we store the texture positions on the vertices of the triangle in itself. This way we do not lose this data for a triangle during the simulation. When two adjacent triangles are separated, as in Figure 12, they do not share the same particle object on the separated side after the tearing. Since the texture position is stored in triangle objects, we still know the texture position on p3 and p3'.

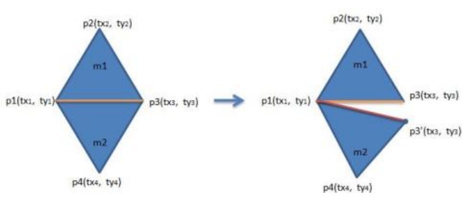

Figure 12: Texture position assignment after detaching one side of two triangles.

In the basic tear model, the triangle is divided into two. The old triangle is eliminated and two new triangles are created instead of the old one, as in Figure 13. We need to calculate the texture positions for these newly created triangles. The position value from the former triangle is used for the undamaged parts of the new triangles. For the ruptured edge, we find the texture position in the middle using the texture position on the tips of the ruptured spring.

In Figure 13, the middle of the texture positions of p2 and p3 is calculated and used at the edges p4 and p5 on the newly created triangles.

Here is an example from our implementation in Figure 14. After we pulled and tore the cloth, we dragged the red point back to the place it was originally connected to. We use a texture which enables us to identify the portions on the cloth, and we see that after the tearing, texture integrity is maintained correctly; no area is lost and texture positions are preserved correctly at the torn parts.

4 Experiments and Results

We developed a cloth simulation implementation and applied our method in this model. Here we used a simple rectangular cloth model (see Figure 14), but our proposed method is applicable to complex irregular cloth models. The cloth is attached from the upper edge. You can drag the cloth from any point with the mouse interactively, and the lengths of the springs on the cloth change. This way the tension on the cloth changes, and if it exceeds a certain threshold, those parts of the cloth are torn.

4.1 Physical Interaction with Outer Objects

We also tested the success of our method to see how it works when the cloth interacts with other objects. There are two key points,

(i) size of the impact area, and (ii) speed of the impact. The pressure is spread among a greater number of particles as the impact area increases and the pressure applied for each particle decreases. Thereby, an impact on a smaller area is more likely to cause a tear on the cloth, whereas an impact on a larger area tends to push the cloth without penetrating. In a similar way, faster impacts are more likely to tear the cloth while slower movements of the same object would not be able to tear it because of the difference in pressure applied.

Handling collision with complicated objects is another extensive research area and outside the scope of this study. In our implementation, we used spherical balls to collide with the cloth, but our model proposes a generic solution to the tearing problem regardless of the shape of the objects that collide with it, so long as the collision detection is handled properly, it is capable of working for an interaction with any kind of object.

In most of the images above, we used the images of an implementation with a low resolution cloth model to be able to explain the technique clearly. We used a high resolution cloth model for the implementation presented in this chapter where we tested how the tearing algorithm would behave in an interaction with a physical object. In addition, the high-resolution model provided a better collision response and showed that our method has no performance problems with detailed cloth models.

In this implementation, we throw balls at the cloth of different sizes and at different speeds. In the first example we observed that impacts applied on a small area can tear our cloth model. The thrown ball has a radius of 0.5 units, which can be considered as a small size for this example, and the speed of the ball is 3 units. In Figure 15 it is shown that there are three balls thrown at the cloth. Two of them tore the cloth and the last one is about to tear.

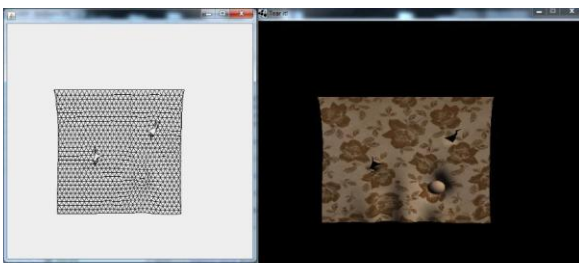

Figure 15: ball with a radius of 0.5 is thrown to the cloth with a speed of 3.

In the second example we increased the ball radius to 1. When we threw the ball at the lower part of the cloth, it passed beneath the cloth without tearing. Cloth slid on the ball as in Figure 16

If we throw the ball at the upper part of the cloth instead, the ball faces some resistance, resulting from strain that is caused by the weight of the lower part of the cloth. Here the size of the pressure area and the speed of the ball is at a level which can apply an impact that can tear the cloth before the cloth would be able to slide on the ball. This is shown in Figure 17.

As a third example, we threw a large ball with a radius of 2 and with a speed of 3. The cloth slid over the ball without being torn. The interaction surface is large, so the pressure is spread around. A greater number of springs responded to the impact so the tension at each spring decreases and they can resist the impact without being torn, as in Figure 18.

As a final example, we increased the speed and threw the same ball with the radius of 2 at a speed of 10 this time. Because of the speed of the ball, the tension in the springs increased so fast that the cloth cannot resist the impact and is torn, as in Figure 19.

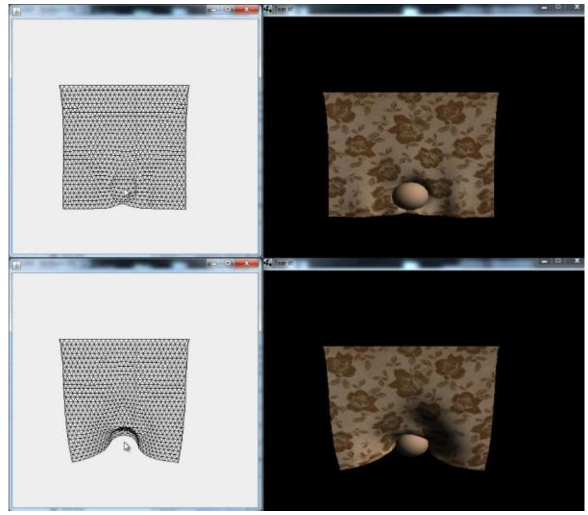

Figure 16: A ball with a radius of 1 is thrown to the lower part of the cloth with a speed of 3.

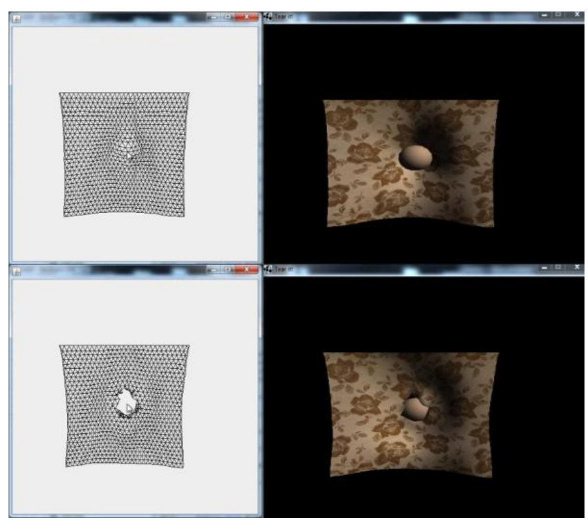

Figure 17: A ball with a radius of 1 is thrown to the upper part of the cloth with a speed of 3.

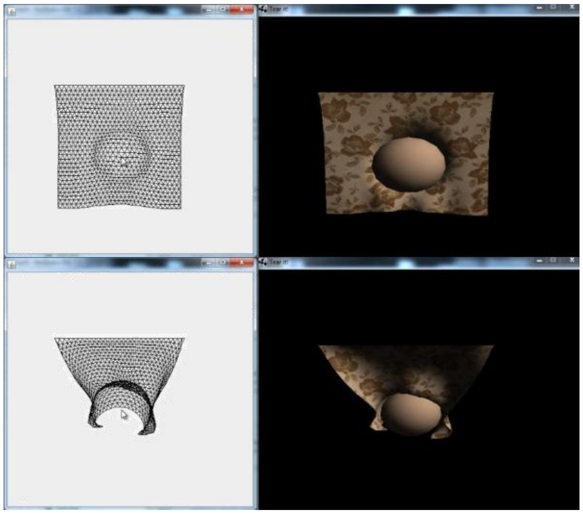

Figure 18: A ball with a radius of 2 is thrown to the cloth with a speed of 3.

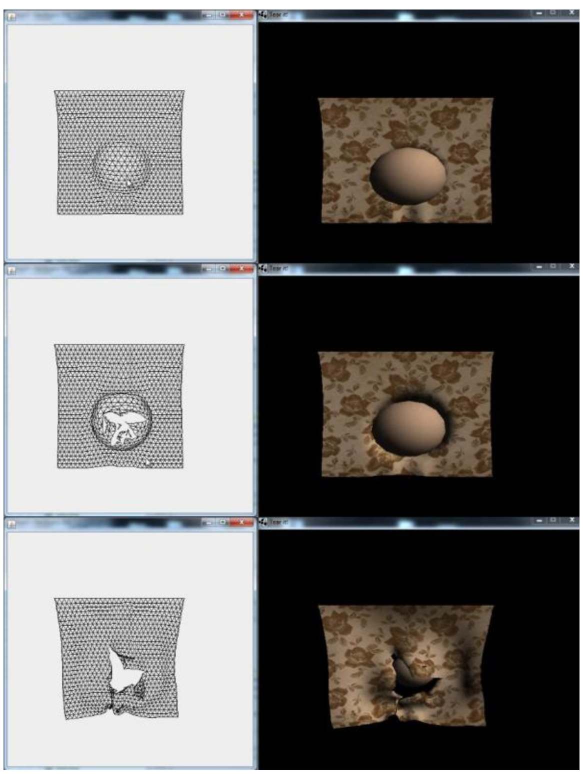

Figure 19: A ball with a radius of 2 is thrown to the cloth with a speed of 10.

4.2 Tear Response to Force Direction

Another experiment we carried out focused on the change of the tear propagation according to the direction of the force. According to the heuristic we developed for weak points, the tear path should be perpendicular to the direction of the force. It means that the tear should continue somewhat in the direction of the pull. We can see that the method could perform visually convincing results as in Figure 20.

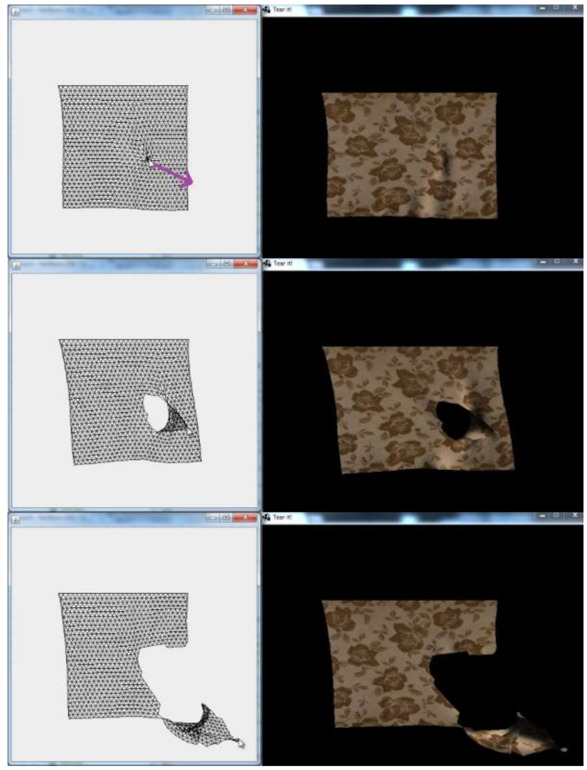

Figure 20: An example of the response of the tear to the direction of the force.

4.3 Profiling Results

One of the most important criteria for evaluating the success of a method used for modelling an interactive physical simulation is real time performance. Running cloth simulations in real time is not that much of a problem for the processing power of today's computers. We extend the current cloth simulation model with new capabilities without compromising the performance. Necessary calculations about tearing are processed only if a high tension is detected on a part of the cloth. Since this is a one-time process, it does not affect the overall performance of the simulation, so our method works without a problem on real time systems. In order to assess performance, we ran a simulation for 35 seconds. During the simulation, we tore different parts of the cloth constantly and observed the method durations. Initial parameters of the simulation:

Resolution of the cloth model used in simulation is 30x30, which makes 900 particles.

There are 2581 structural springs

There are 2465 bend-shear

springs. There are 1682 triangles.

These values change during the simulation as the tearing occurs. The forces that apply on each particle are the spring force, gravity, air friction, damping and wind. The specs of the computer that we run this simulation on are:

Operating System: 64-bit Windows 7 Ultimate

Processor: AMD Phenom II X2 555 3.20GHz

Ram: 4GB DDR3

GPU: NVIDIA GeForce GTX560 (Core Clock: 820MHz, Memory Size: 1GB GDDR5)

The profiling result obtained using Netbeans profiler is shown in Figure 21.

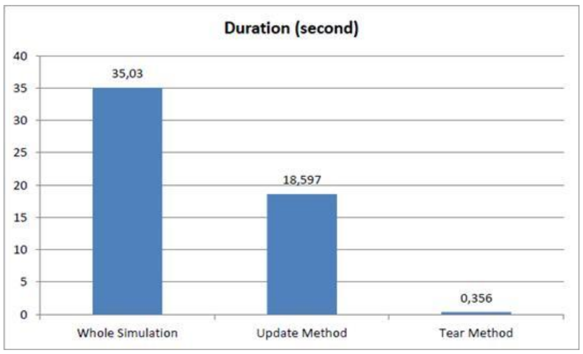

Figure 21: Textured, torn cloth.

The whole simulation takes 35 seconds, but our concern is regarding the Tearit.update() method. This is the place where all the required calculations for the simulation are done, and it takes about 18 seconds. On the highlighted line, we can see our tear() method. It takes 356 milliseconds. This is relatively low and does not have a significant effect on the performance of the simulation. There are 510 tearing process during this simulation.

Moreover, our model can reduce the computational cost since it works with non-uniform models. It is possible to model a cloth with fewer particles with a non-uniform model. Unnecessary triangles can be eliminated where a high level of detail is not necessary. This way the number of particles in the system is reduced, resulting in performance gain.

5 Conclusion

5.1 Contributions

There have not been many studies around the tearing of cloth. The existing ones have only dealt with some aspects of it, and they have not provided a general solution to this problem. Our aim in this paper was to fill this gap and provide a general solution to this problem, dealing with all aspects of it.

We tried to recreate a realistic tearing effect while partitioning the triangles on the cloth and generating the tear path by taking advantage of physical rules to achieve satisfactory results.

The presented method preserves the polygonal area consistency after the tearing by avoiding the elimination of any polygonal area at the process of tearing. At the same time, texture integrity is maintained successfully, regardless of any structural changes by the help of the triangle structure we used in our model.

Another important aspect of this study is that it is able to react successfully to external physical impacts. The reaction changes realistically according to the pressure area and the speed of impact. It is also capable of responding to direct user interactions successfully. The tear propagation changes according to the direction of force and this increases the sense of physical reality. We showed examples of these in the experiments and results section.

As much as the proposed method provides results with a good level of realism, it is also able to provide these results at an insignificant cost.

There is an important factor that makes the contributions mentioned above more valuable. The cloth model we used in our method is a generic model which could be adapted easily to many applications. It is based on spring physics, which is used widely used for cloth simulations, and also supports non-uniform cloth structure with the help of irregular triangular meshes.

5.2 Future Work

There are many studies about collision handling with cloth. A study which observes the interaction of different kinds of objects with this cloth model would be an interesting study, like using a knife for cutting a cloth.

Spring model is used for cloth simulations regularly, but it is also used for modelling some volumetric rigid objects [Criswell et al.]. Strict bending constraints are applied to the springs in some of those models to constitute volumetric shape. It may be possible to extend this model to support cracks on a volumetric object.

Another area of expertise, creating tearing sounds during the process of tearing the cloth, would be a nice complementary study to this study. However, the resolution of polygons determines the moment of tear and this would be a difficult problem for developing a successful algorithm for generating sounds in a proper way.

Another original addition to this study would be implementing a tessellation methodology to this algorithm. It is mentioned that the method presented in this study is applicable to multi-resolution cloth models. The parts modelled with coarser resolution could be torn in detail by subdivision and produce more realistic results even for a low resolution cloth model.

References

BAO, Z., HONG, J.-M., TERAN, J., AND FEDKIW, R. 2007. Fracturing rigid materials. *IEEE Transactions on Visualization and Computer Graphics 13, 2 (Mar.)*, 370–378.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH ’98, 43–54.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH ’02, 594–603.

CRISWELL, B., LENTINE, M., AND SAUERS, S. Avatar: Bending rigid bodies.

DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. In *ACM SIGGRAPH*

2010 papers, ACM, New York, NY, USA, SIGGRAPH ’10, 106:1–106:9.

H¨A GGSTR¨OM, O. 2009. Interactive real time cloth simulation with adaptive level of detail.

HELLRUNG, J., SELLE, A., SHEK, A., SIFAKIS, E., AND TERAN, J. 2009. Geometric fracture modeling in bolt. In *SIGGRAPH 2009: Talks,* ACM, New York, NY, USA, SIGGRAPH ’09, 7:1–7:1.

JAIN, N., KABUL, I., GOVINDARAJU, N. K., MANOCHA, D., AND LIN, M. 2005. Multi-resolution collision handling for cloth-like simulations. *Computer Animation and Virtual Worlds* 16, 3-4, 141–151.

KELAGER, M., NIEBE, S., AND ERLEBEN, K. 2010. A triangle bending constraint model for position-based dynamics. In *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS.*

METAAPHANON, N., BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2009. Simulation of tearing cloth with frayed edges. *Comput. Graph. Forum* 28, 7, 1837–1844.

NARAIN, R., SAMII, A., AND O’BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics* 31, 6 (Nov.), 147:1–10. Proceedings of ACM SIGGRAPH Asia 2012, Singapore.

NEALEN, A., MLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum* 25, 4, 809–836.

NG, H. N., AND GRIMSDALE, R. L. 1996. Computer graphics techniques for modeling cloth. *Computer Graphics and Applications, IEEE* 16, 5, 28–41.

O’BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH ’99, 137–146.

O’BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. Graph.* 21, 3 (July), 291–294.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface,* Canadian Information Processing Society, 147–147.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. *SIGGRAPH Comput. Graph*. 22, 4 (June), 269–278.