CLOTH TEARING SIMULATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE ÖNAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GAME TECHNOLOGIES

JANUARY 2013

CLOTH TEARING SIMULATION

submitted by **EMRE ÖNAL** in partial fulfillment of the requirements for the degree of **Master of Science in Game Technologies, Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Director, Informatics Institute ____________________

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu
Head of Department, MODSIM, METU ____________________

Assoc. Prof. Dr. Veysi İşler
Supervisor, Compter Engineering, METU ____________________

Dr. Çağatay Ündeğer
Co-Supervisor, SimBT Ltd. Sti. ____________________

Examining Committee Members:

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu
Modelling and Simulation Dept., METU ____________________

Assoc. Prof. Dr. Veysi İşler
Computer Engineering Dept., METU ____________________

Assist. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Dept., METU ____________________

Dr. Umut Durak
Roketsan Missiles Inc. ____________________

Dr Erdal Yılmaz
Sobee Studios ____________________

Date: **31.01.2013**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Emre, Önal

Signature:

ABSTRACT

CLOTH TEARING SIMULATION

Önal, Emre

M.Sc., Department of Modelling and Simulation
Supervisor: Assoc. Prof. Dr. Veysi İşler
Co-Supervisor: Dr. Çağatay Ündeğer

January 2013, 50 pages

Among different physical simulation topics, cloth simulation is one of the popular subjects in computer graphics. There are many different studies published on different aspects of cloth simulation, but there are not many studies which are focused on the tearing of cloth. Existing studies related to this topic have only dealt with some aspects of the problem and have not provided general solutions. In this thesis, 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, handling the tear properly for both outer physical impacts or inner manipulations like allowing a user to drag the cloth interactively.

The technique proposed in this thesis works with non-uniform cloth structure. 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 realtime simulation systems.

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

ÖZ

KUMAŞ YIRTILMA SİMÜLASYONU

Önal, Emre

Yüksek Lisans, Modelleme ve Simülasyon Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Veysi İşler
Ortak Tez Yöneticisi: Dr. Çağatay Ündeğer

Ocak 2013, 50 sayfa

Farklı fizik simülasyonları başlığı arasında, kumaş simülasyonu bilgisayar grafiği alanında popüler konular arasındadır. Kumaş simülasyonunun farklı yönleri hakkında birçok çalışma yayınlanmıştır yalnız kumaş yırtılması üzerine yoğunlaşmış pek fazla çalışma yoktur. Kumaş yırtılması ile ilgili olan az sayıda çalışmalar da problemin sadece belirli yönleriyle ilgilendiler ve genel çözüm sağlamamıştır. Bu çalışmada kumaş yırtılması probleminin farklı yönleri için genel bir çözüm sunulmaktadır.

Bu çalışmada üzerinde yoğunlaşılan noktalar şunlardır; gerçekçi yırtılma efekti sağlamak, yırtılma işleminden sonra yüzeysel alan bütünlüğünü ve doku bütünlüğünü korumak, dışarıdan fiziksel darbeler veya kullanıcının kumaşı sürüklemesi gibi içeriden etkileşimlere karşı kumaş yırtılmasını olması gerektiği gibi yürütmek.

Bu tezde sunulan teknik düzensiz kumaş yapısıyla çalışmaktadır. Bu birçok simülasyon sisteminin burada sunulan metodu kendine uyarlamasını

kolaylaştırmaktadır. Bu tekniğin işlem maliyeti çok düşüktür ve bu yüzden gerçek zamanlı simülasyon sistemleri için de uygundur.

Keywords: Kumaş Yırtılması, Kumaş Simülasyonu, Yay Fiziği, Maddelerin Kırılması

To My Family

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assoc. Prof. Dr. Veysi İşler for his support and thank all of my instructors in GATE who shared their experience with sincerity.

I am thankful to Kadir Yılmaz for his opinions and graphical support.

Finally I would thank to my family for their support and staying at late hours with me.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

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 on its own. There are many different studies on this topic [16] [13] [2], and today it is considered as an already solved problem[12]. However, in most of these studies, the tearing of cloth is not taken into consideration, therefore the methods presented in those studies are not appropriate for the tearing of a cloth.

Since the tearing of clothes has its own features, it also needs to be treated as a different problem in computer graphics. There are existing studies on fracturing materials, but since the nature of cloth physics differs from other kind of objects, such methods are not applicable to handle the tearing of cloth.

Sheet-based cloth model is the most common method used in cloth animation[16]. In this model, cloth is constructed using connected springs. Despite the fact that it does not represent the exact dynamics of cloth, the results are satisfactory in terms of their accordance with reality. Using spring physics produces these results without the need for using expensive exact physical rules that apply to real world. This makes it suitable for many systems used in computer animations, and it is widely used.

1.2 Scope of Thesis

In this study, we use a sheet-based cloth model and extend this model 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 our concern.

Another problem is to find a solution which could work on non-uniform cloth structures. In many cloth simulation applications, particles are placed on a regular grid and they are stored in two dimensional arrays in most of them[12][16]. This makes it easier for programmers to reach a point on the cloth using array indices, but in many situations this may not be an ideal way to model the cloth. According to the needs of the application, it would be better to construct the cloth model in a non-uniform way to avoid computational overhead or it would be necessary to model some parts of the cloth with a different resolution to reflect a better visual quality on parts which require more detailed movement capabilities, like the armpits of a sweater. The proposed method is applicable to non-uniform models, which makes it a general solution for a wide range of usage area.

We extend this sheet-based model using an object oriented approach, this makes it easier to maintain the integrity while making changes on the cloth structure. In addition to springs and particles, we use a data structure called a triangle. A triangle is composed of the springs, which represents a polygon on the cloth. While drawing cloth, required data is taken from its related triangle object for each polygon. We take advantage of these triangle objects while making structural changes on the cloth and maintaining the textural integrity.

While handling the tear in a cloth, we may encounter different cases which need to be handled seperately. We detect the state of the cloth and apply different solutions for these different cases.

Damaged and undamaged parts of the cloth would respond to an impact in different ways. While triggering a tear on an undamaged region, we face a 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 the 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.

1.3 Outline

The thesis is structured as follows:

- Chapter 2 includes some background information about cloth simulation and explains the sheet-based cloth model we use as a base model for our study. Then we provide an overview of some studies related to structural changes, like tearing or fracturing in different kind of objects.
- Chapter 3 is grouped into three parts:
    - In the first part we describe our cloth model, which is extended from the sheet-based model. Here we introduce the triangle structure, which is beneficial in handling many problems we face. Then we explain how we meet the shear and bending constraints using our modified cloth model.
    - Three different tearing models are introduced. Each of them is applied under different conditions of the cloth.
    - We explain how we preserve the texture integrity after the structural changes on the cloth that result from the tear and show an example.
- Chapter 4 we discuss the success of our method with two different examples and a profiling result of a sample simulation run. One of the examples show how our model responds to an outer physical impact. The other example shows how the tear path changes according to force direction. The profiling result gives the processing durations of different parts of the algorithm and shows that the time spent for tearing is insignificant.

- Chapter 5 summarizes and evaluates the contributions of this study, then mentions some possible future works.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Cloth Simulation

Among many different methods for cloth simulation[17] sheet based cloth model is the most commonly accepted method and used as a basis in many different studies. We will also use this model and extend it according to our needs.

2.1.1 Sheet Based Cloth Model

In sheet based cloth modelling;

- The cloth is treated as a system of particles interconnected with spring-dampers as in Figure 2.1.
- Each spring-damper connects two particles, and generates an attraction or repulsion force based on their positions according to the spring constant.
- Each particle is subjected to different kinds of internal and external forces. The internal force is a result of the attraction or repulsion forces of the connected springs. A viscous damping force would also be considered as an internal force, which is the inner friction of the springs[16]. There would be different kinds of outer forces according to the simulation conditions. This could be gravity, air friction, wind, etc.

  With these forces we form the foundation of the cloth system.

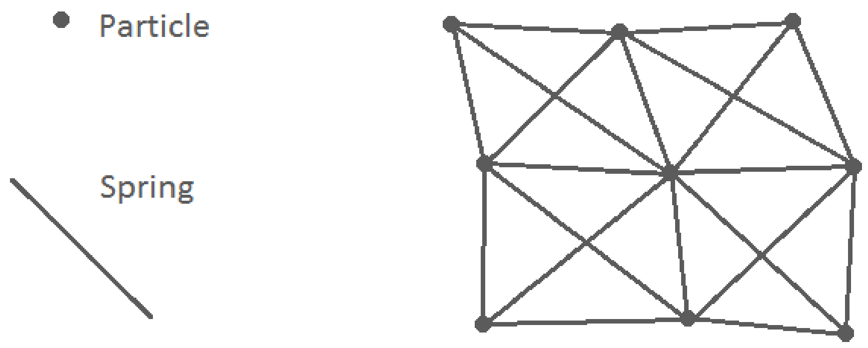

Figure 2.1: Continuum sheet model for cloth simulation

Once all of the total forces that apply to particles in the system are computed, we use Newton's second law (***f=mxa***) to compute the acceleration. Then we use the acceleration to forward the system by a time step Δt using the Euler integration.

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t$$
$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+1} \Delta t$$

These were the basics for a sheet-based model. It is possible to create a cloth model with these constraints, but it will have some deficiencies. There are some other constraints to overcome these problems and produce a more realistic cloth model.

One of the problems is the limiting of the deformation of the springs. Different forces act on particles, and as a result the length of the springs may increase and create a deformation effect that is unrealistic compared to a real fabric cloth. The main reason for this difference between this spring model and the real woven fabric is that woven fabrics are not as elastic as springs, and they do not elongate so easily. Increasing the spring constant makes the cloth stiffer and lowers the deformation level, but as we increase the spring constant the particles are subject to higher inner forces and this may result higher acceleration. When the displacement of the particles start to increase the whole system tends to lose its stability [16]. We need to decrease the timesteps but we are restricted by hardware limits at this point. Even at a level in which the system is stable, the cloth would oscillate at a high amplitute like a sheet of rubber, which is not the case for a fabric cloth. Increasing the damping coefficient of the spring would decrease this oscillation but this will cause it to move like a gummy.

A successful solution to this problem is to limit the deformation of a spring to a predetermined threshold [16]. If the distance between the particles of a spring exceeds a certain ratio, we move these particles towards each other.

Although we correct the distance between the particles connected with springs to a certain ratio, the distance between them may still exceed the limits we determined because the position of a particle is changed cumulatively for each spring connected to it. Nevertheless, this process makes the system stable, prevents the problems mentioned above and make the cloth look like woven fabric.

Other constraints are about the inner mechanics of the cloth. These constraints are maintained by three kinds of springs: structural, shear and bending springs, as seen in Figure 2.2.

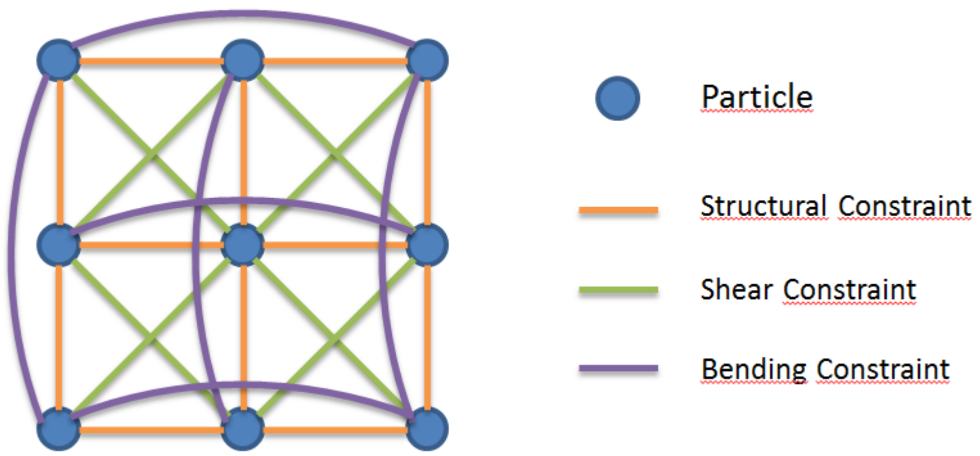

Figure 2.2: Structural, shear and bending constraints on a cloth.

Structural constraint is a must in this sheet based model. Each particle on the corners of the grids, which resides on a two dimensional surface, is connected to its vertical and horizontal neighbours. It is the basic constraint that enables a group of particles to behave like a sheet. Shear constraint helps the cloth to resist shearing and preserve its rectangular shape. The bending constraint prevents neighbour springs from folding onto each other and helps the cloth have nice looking and realistic, smooth bendings while moving and also preserve the drapery look.

This is the most common and a successful usage of the sheet based cloth model. This model gives realistic and efficient results for cloth animation, but in this model it is not taken into consideration to handle structural changes in the cloth, like tearing. Simply breaking the connections will not result in a proper tearing effect. The connections between particles are not designed to be seperated. There are many springs crossing each other, and there is not a direct relationship between the polygons and the springs, so it is unclear what will happen when you break these connections. The model will probably just start to animate in an unrealistic way. The polygonal area on the cloth will not be consistent anymore. Texture will not be partitioned in a proper way on the required regions.

2.2 Related Work

There are many studies on cloth animation with different techniques [17] [14]. The interaction of clothes with other objects is also covered in many studies [13][14], but in most of these studies, the tearing of cloth is not taken into consideration. Therefore the methods presented in those studies are not appropriate for the tearing of a cloth.

In an earlier study, Terzopoulos and Fleischer [6] proposed some methods on modelling inelastic deformation. They have studied viscoelasticity, plasticity and fracture. In one of the examples they showed fracture propagation on surfaces with a net falling over an obstacle. They used deformable model formulation, but the springs of the cloth were subject to fracture limits.

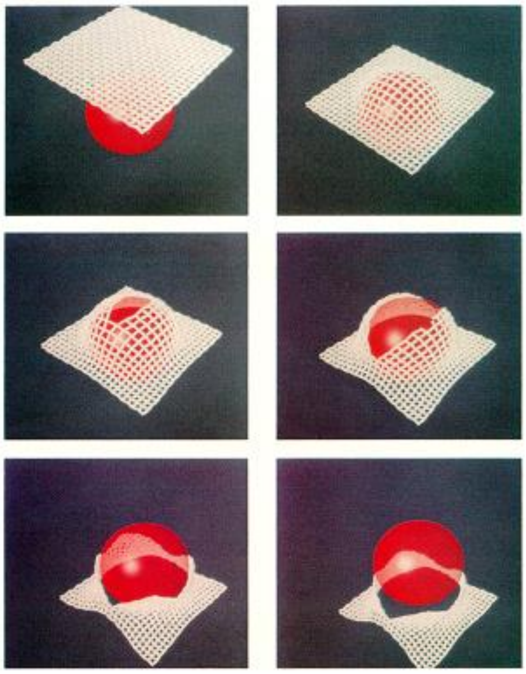

Figure 2.3: A net falling over a spherical obstacle. Fractures develop and propagate as the deformation exceeds the elastic limit (reprinted from [6])

They achieved a tearing effect as shown in Figure 2.3, but in this example the cloth object is a connected group of fibers. It does not have a polygonal surface. Also, since they used a deformable model to represent cloth, this system is not capable of providing a realistic cloth simulation system. Using a deformable model is not appropriate to simulate real cloth dynamics.

Metaaphanon et al. [1] 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 is modeled according to the yarn-level model, and tearing occurs in this yarn-level modeled part as in Figure 2.4.

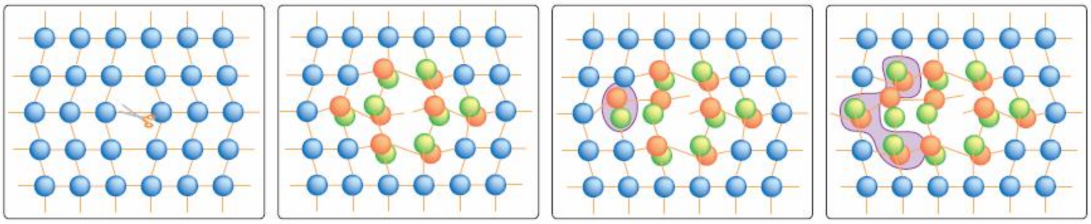

Figure 2.4: Transition from the sheet cloth model to the yarn-level model.(reprinted from [1])

This way they simulate the behavior of threads on the torn lines as in Figure 2.5, but the main focus of this yarn-level model is around this specific point only, frays at the edges. It does not propose a general model for the management of polygonal surface and the textures of the cloth.

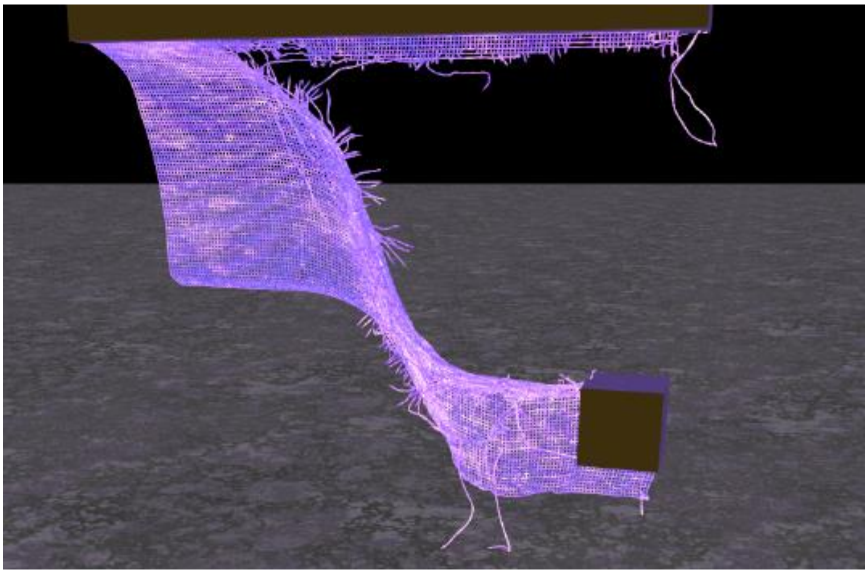

Figure 2.5: A piece of cloth is pulled and torn by a cube.(reprinted from [1])

Thomas made a study related to the tearing of cloth [22]. This study was a final project for a 'physically based animation' course at the University of Pennsylvania. We do not have detailed information about the technique, but there are some results so we can evaluate the success of the method. The cloth can be torn in two ways, (i) by stretching the cloth as in Figure 2.6 and (ii) by interacting with a physical object as in Figure 2.7. Although the deformation seems unrealistic, tearing by stretching the cloth seems usable in some applications that do not require a high level of realism.

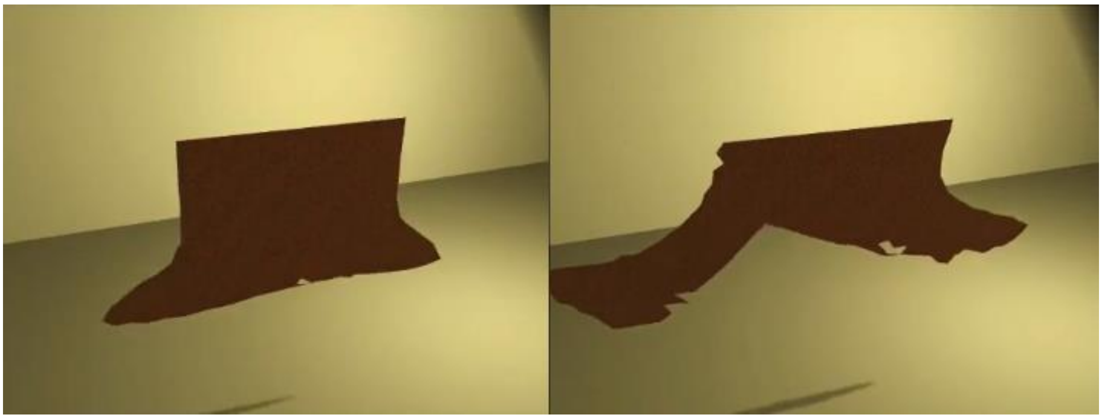

Figure 2.6: Tearing by stretching the cloth. (reprinted from [22])

However, the results for interacting with rigid objects does not seem reasonable. There is a ball thrown to the cloth in Figure 2.7. The ball penetrates the cloth but can not make a proper tear while penetrating.

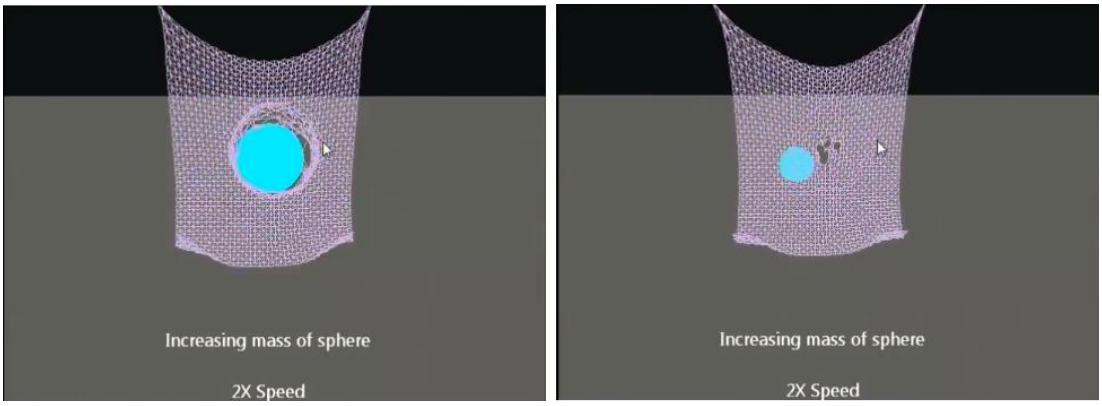

Figure 2.7: Tearing by throwing a ball. (reprinted from [22])

In many studies related to cloth, the tearing of cloth is mentioned as a future study [18] [19], but there is yet to appear a complete study that proposes a successful generic solution for tearing in a cloth.

Fracturing of rigid materials is also of relevance. Hellrung et al. [8] designed a system for cracking and shattering objects. This system aims to overcome the problem of tuning cracks and shattering objects manually, especially when there are a large number of fragments are to be produced. They propose a model to faciliate this process enabling an automatic generation of cracks that also allow the controlling of the complexity and density of the crack formulation. An example of their method is given in Figure 2.8. Since this model is only applicable to rigid objects, it is not appropriate for handling tears in cloth-like materials.

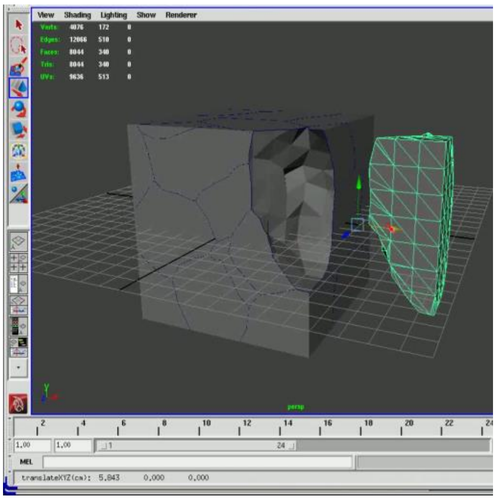

Figure 2.8: A fractured fragment seperated through the surfaces which are defined using Voronoi regions. (reprinted from [8])

Zhaosheng Bao and Jeong-Mo Hong [9] proposed an algorithm to handle the fracture of stiff and brittle materials in which the objects are treated as rigid bodies. Since that method is designed specifically for rigid bodies, it is not applicable for cloth-like materials either.

O'Brien and Hodgins had some studies on crack initiation and propagation [10], 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. 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 [15].

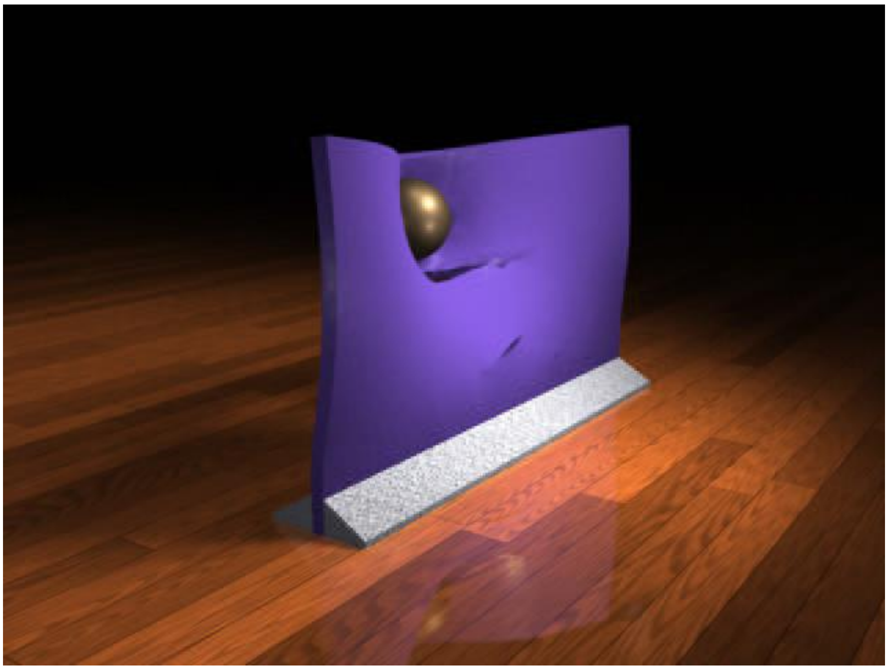

Figure 2.9: Example of a ductile fracture.

Fracturing a brittle material can not be used to represent a cloth tear, but it is possible to use a ductile object fracture to look like a tearing cloth by configuring the material properties, as in Figure 2.9. However the results were not satisfactory. The created effect would seem like a cloth tear but if we think about a whole simulation system still, this will not be an efficient solution since a ductile object model is not appropriate to represent normal cloth movements.

CHAPTER 3

METHODOLOGY

3.1 Cloth Model

In our study we use the basics in the sheet-based model mentioned above and extend it according to our needs.

First of all, since we want to support non-uniform cloth models, the springs and particles in our system are not constrained to be on a regular grid. There is a more strict relationship between polygons and springs in our model. Each vertex of a polygon on the cloth is a particle, and these particles are connected with springs so each of those springs also represent the edges of the polygons on the cloth.

Since we change the structure to a non-uniform model, we also can not use the shear and bending constraints as they are in a standard sheet-based 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. Springs crossing each other irregularly would make it harder to process the segmentation of regions on the cloth. If we do not want to sacrifice the visual quality maintained by those constraints for the sake of achieving a vulnerable cloth model, we need to find a way to meet those constraints.

Here we intoduce a new data type called triangle. We use this data type heavily in our method.

3.1.1. Triangle Structure

We use an additional data structure for triangles in the cloth model. It is used in different parts of the simulation steps. Each triangle represents a polygon on the cloth, like in figure 3.1.

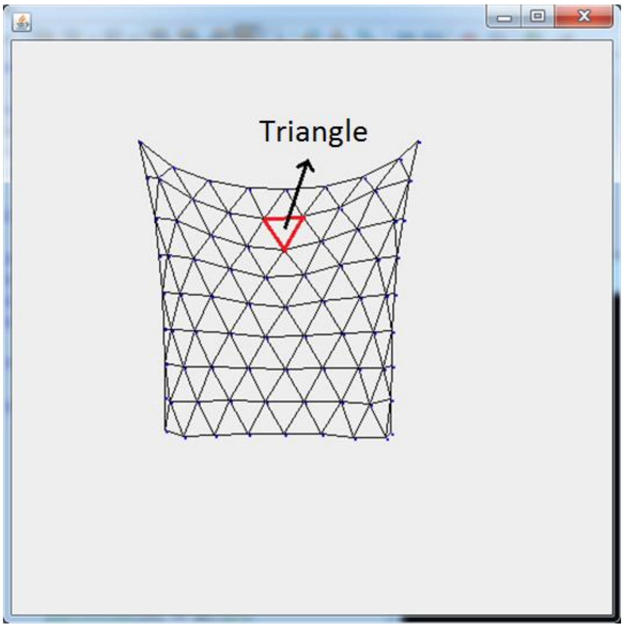

Figure 3.1: Triangle structure in the cloth model

In cloth simulations, particles and springs have been used as the building blocks of the cloth model. Triangle is like the third step in this object structure hierarchy. Springs connect particles and triangles connect springs. Two neighbour triangles share the same spring object at the intersection edge as two connected springs share the same particle object at the intersection point.

A triangle object contains the spring objects surrounding the polygon, but we also keep reference to particle objects at the corners for better control and faster access.

Since the triangle object is directly related to a polygon, it also holds the texture positions of the related polygon. We assign the texture positions at the initialization step and they are preserved correctly during the simulation even if that triangle is detached from its neighbour triangle because of a rupture.

In our implementation, 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. This data is used in different steps like, searching the tear path and reconstructing the cloth structure, calculating an interpolated normal vector for a particle which is used for lighting by searching the surrounding triangles, and detecting the related opposite particles which will be used in bending and shear constraints.

The example cloth model we used here in Figure 3.1 is like a uniform model, but it is designed that way just to clarify the structure and the algorithm. This triangle model presented here is not constrained to the uniform structure, as the method works in the same way for non-uniform models, like in Figure 3.2. However, an importer should be developed to import the vertices of the polygons of a model, which is typically modelled by an artist, and initialize the triangles of the cloth at the beginning of the simulation.

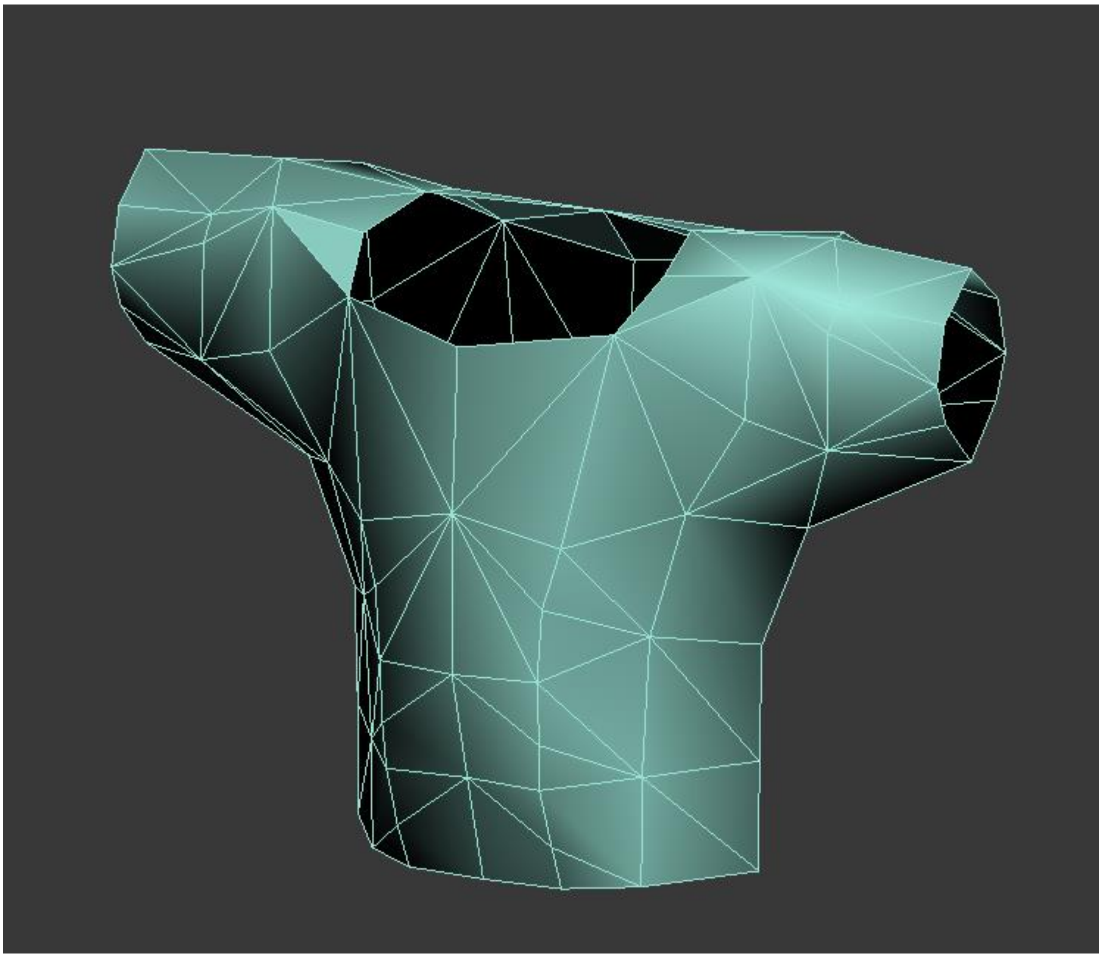

Figure 3.2: Non-uniform triangles on a cloth

3.1.2 Bending and Shear Constraints

We explained that the classic bending and shear constraints are not applicable to our triangle model. Mickey et al. published a study about a triangle bending constraint model [20] in which the difference between the normals of two adjacent triangles was 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. As a result, we have the normal data for the two adjacent triangles of a spring, if the spring is not on an edge of the cloth or on a torn line.

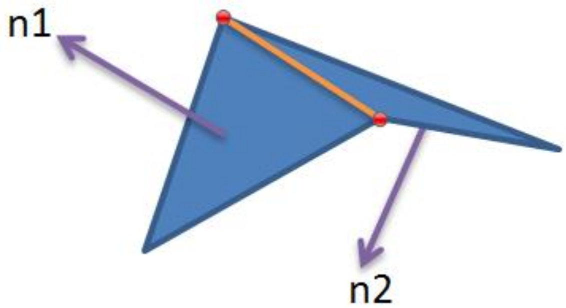

Figure 3.3: Normals of two adjacent triangles.

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 3.4. The repulsion force of this spring meets the bending constraint between the two triangles. In Figure 3.4, 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.

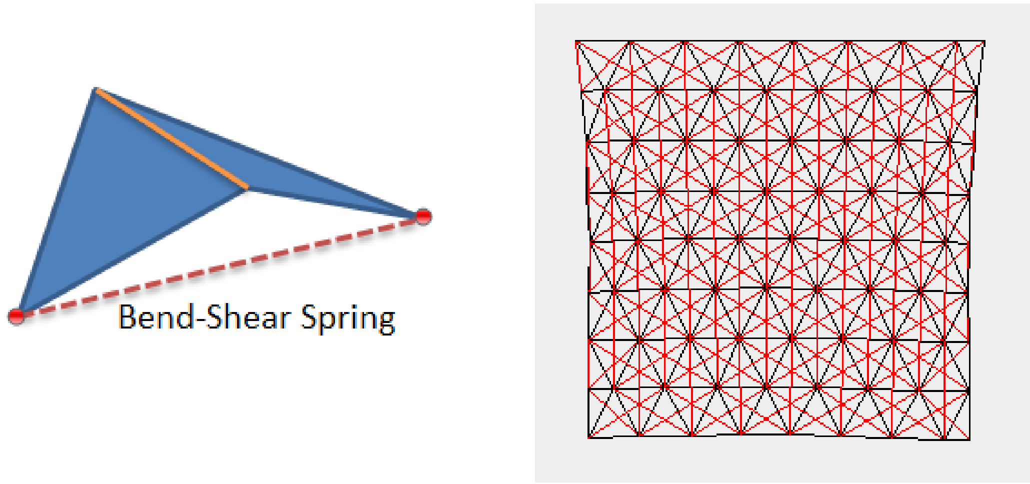

Figure 3.4: Bend-shear springs.

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 3.4 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.

We use the length of the surrounding springs of the two adjacent triangles to be able to calculate the length of this bend-shear spring we create between these two triangles.

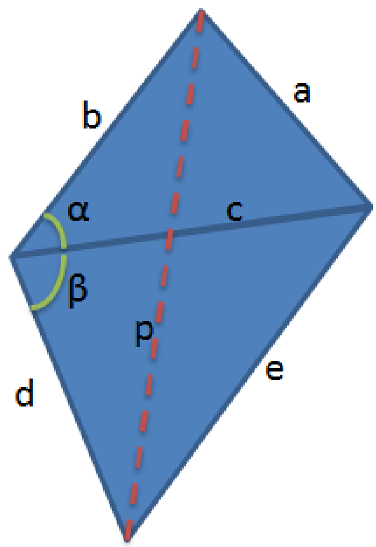

Figure 3.5: The length of a bend-shear spring.

In Figure 3.5, there are two triangles and a bend-shear spring is represented as the dotted red line. a, b, c, d, e and p are the lengths of the springs. We know a, b, c, d and e, and we use them to find p.

We use cosine rule in equation 1.

$$a^2 = b^2 + c^2 - 2bc\cos\alpha \qquad (Equation\ 1)$$

$$\alpha = \cos^{-1}\left(\frac{b^2 + c^2 - a^2}{2bc}\right) \qquad (Equation\ 2)$$

$$\theta = \alpha + \beta \qquad (Equation\ 3)$$

$$p = \sqrt{b^2 + d^2 - 2bd\cos\theta} \qquad (Equation\ 4)$$

We acquire the length p by equation 4. This is the length 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 lengths 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 3.6, 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.

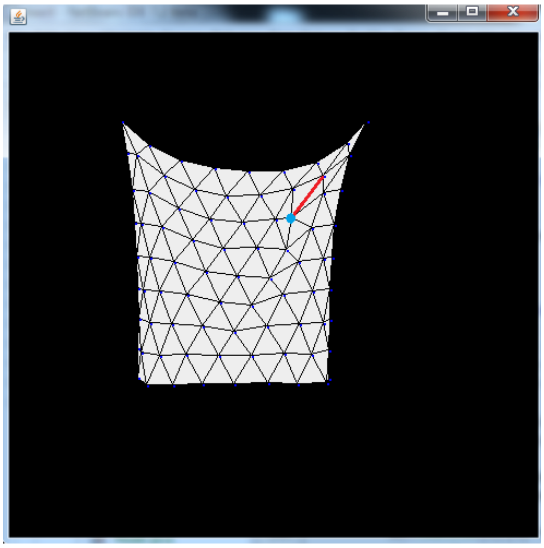

Figure 3.6: Tension on the cloth.

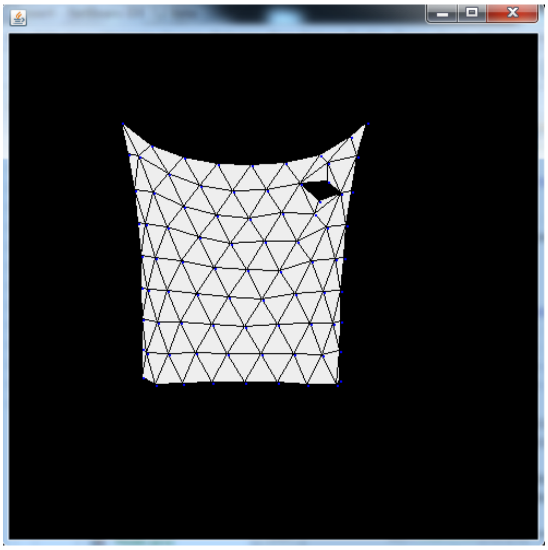

Figure 3.7: Basic tear.

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. In Figure 3.8, m1 is removed from the system, and we create m3 and m4 instead of m1. We do the same for m2.

We prepare new springs according to the new triangles and use them for the creation of those new triangles. The red springs are the remainder of the spring, which is subjected to high 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 3.8, since m3 and m4 do not share an adjacent spring, they can move independently. It is the same for m5 and m6.

On the other hand, since m3 and m5 share the same red spring, they are connected and they do not move independently. It is the same for m4 and m6.

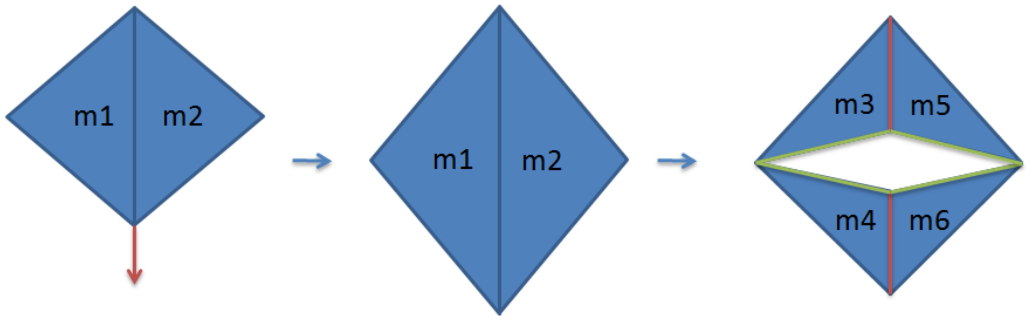

Figure 3.8: Basic tear.

There is a small calibration that needs to be mentioned. In this example, the spring on the intersection edge is elongated and is 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 3.9. 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.

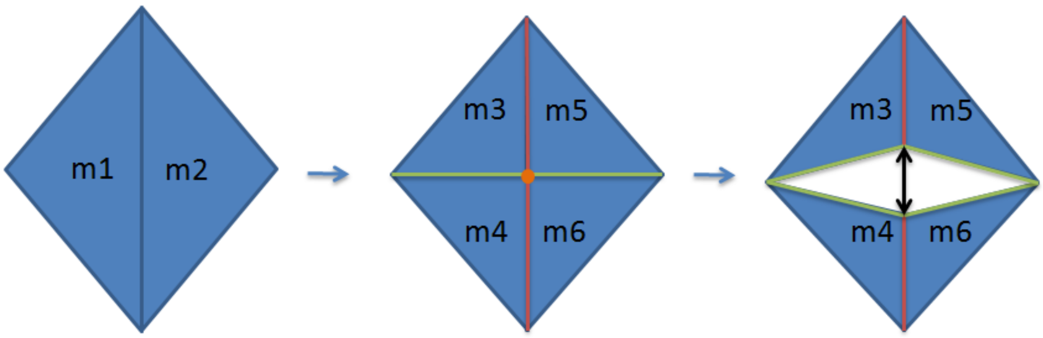

Figure 3.9: The particle displacement after a basic tear.

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. After the creation, the springs will reach the length they should have since there is no attraction force applied to the points on the torn edges.

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 makes the hole larger, rather than creating another hole next to the former one.

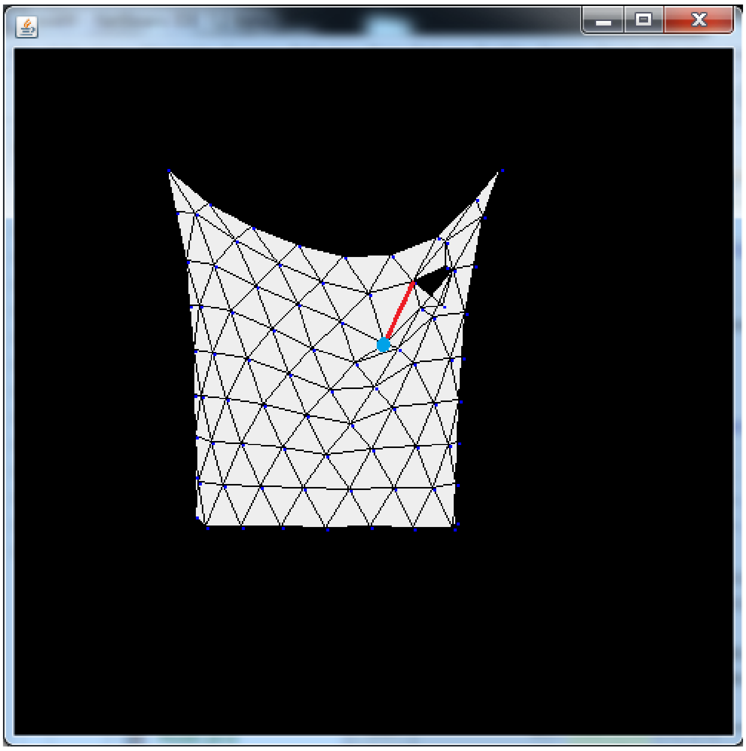

Figure 3.10: Tension around a weak point

For example, as in Figure 3.10 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 as in Figure 3.11. 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 are connected to weak points.

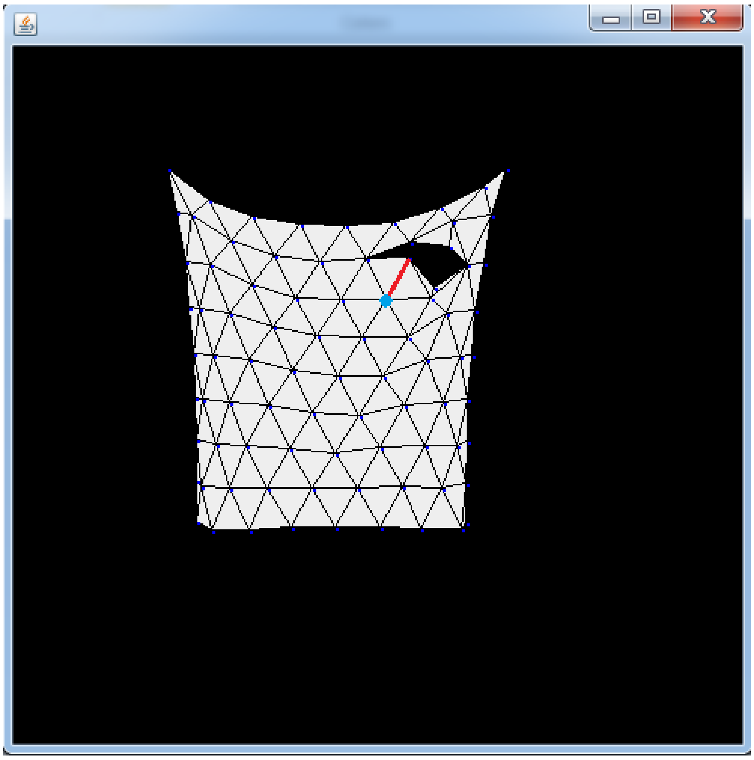

Figure 3.11: Tear around a weak point

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 3.8. Instead, we select one of springs connected to this weak point and detach the two neighbour triangles.

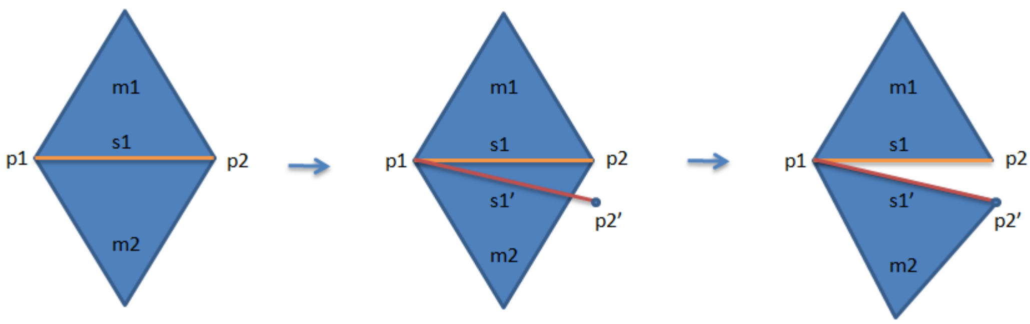

Figure 3.12: Detaching the triangles connected to the weak point p2.

When the tear continues on a spring connected to a weak point, we need to detach the neighbour triangles of that spring so that those triangles will not be connected and move together anymore. It means that they should not share the same spring after the tearing, this way the triangles can move independently from each other. There is a demonstration of this in Figure 3.12. Here we do not create a new triangle object as we do in the basic tear. We modify the existing structure by changing the connected spring and particles. We create a replacement particle p2' for the weak point p2 and connect the triangles on the other side of the tear to this particle. This way we get a result as in figure 3.12.

An important problem at this step is to determine the tear path. For example, in this case in Figure 3.13, 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.

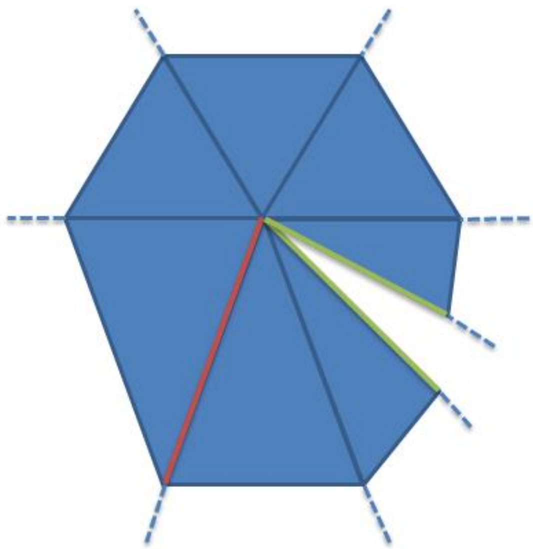

Figure 3.13: Possible paths for a tear on a weak point.

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. We can observe this in figure 3.14. 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.

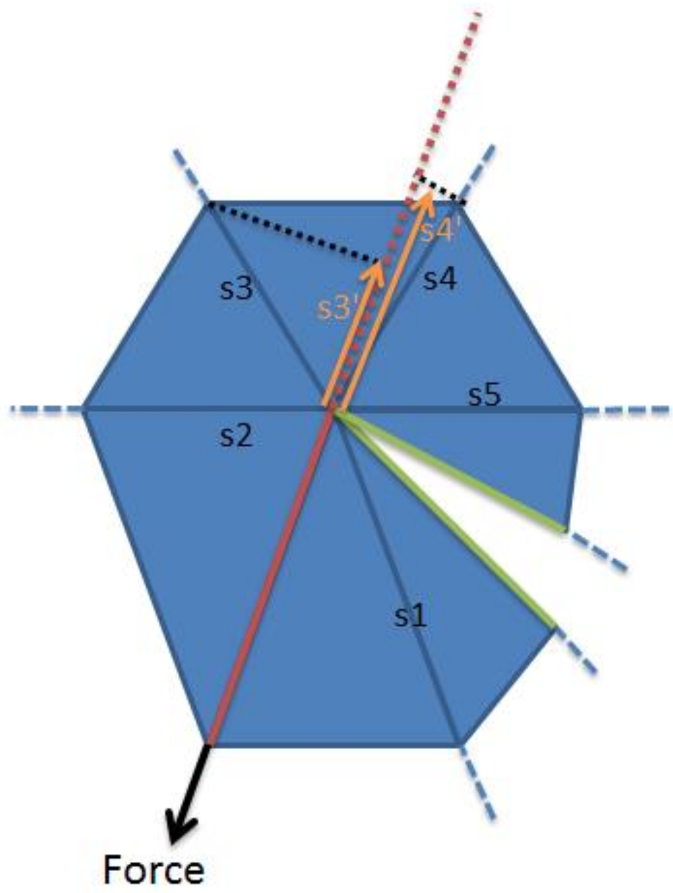

Figure 3.14: The projection of spring forces on the axis of tension.

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 group. 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 one of 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.

For this example, 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 3.15. According to this example, among s2 and s3, s2 has the smaller angle, so we choose it for the tear path.

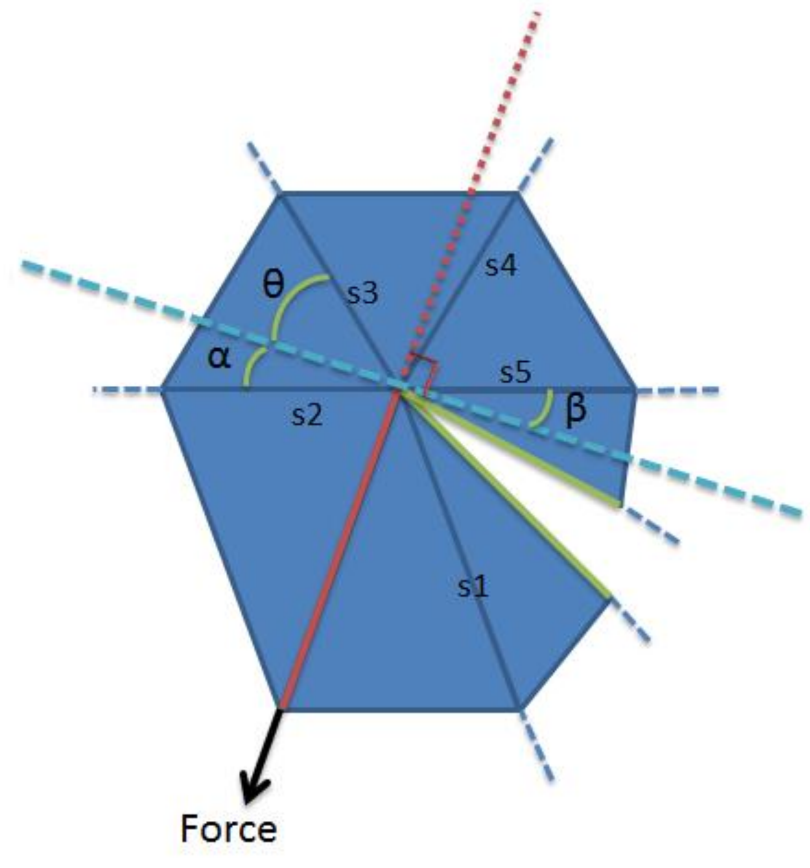

Figure 3.15: The angles between the spring forces and the axis perpendicular to tension axis.

The angle between the force vector of s5 and the axis perpendicular to the tension axis is shown as **β** here. Even though **β** could be smaller than **α** we do not take it into consideration since it is on other side of the tension axis. After we tear through s2, we get the result in Figure 3.16.

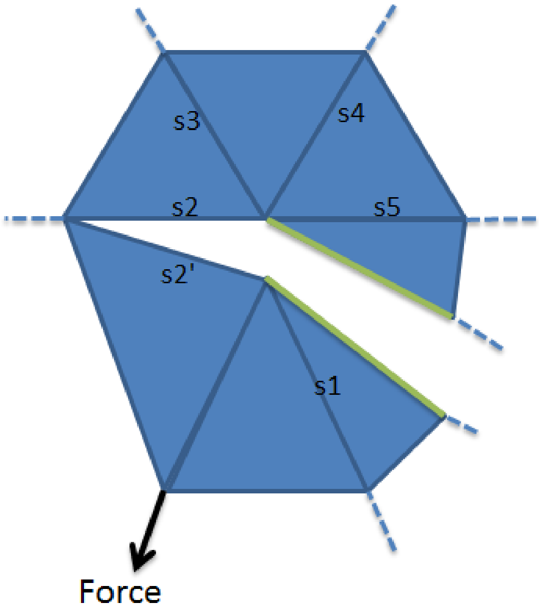

Figure 3.16: The structure change after tearing through the selected spring.

After a weak point is ruptured and as the tear continues on a weak path, the cloth structure and weak points change. For Figure 3.12, 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 the figures below, and are shown as blue points on the wireframe view.

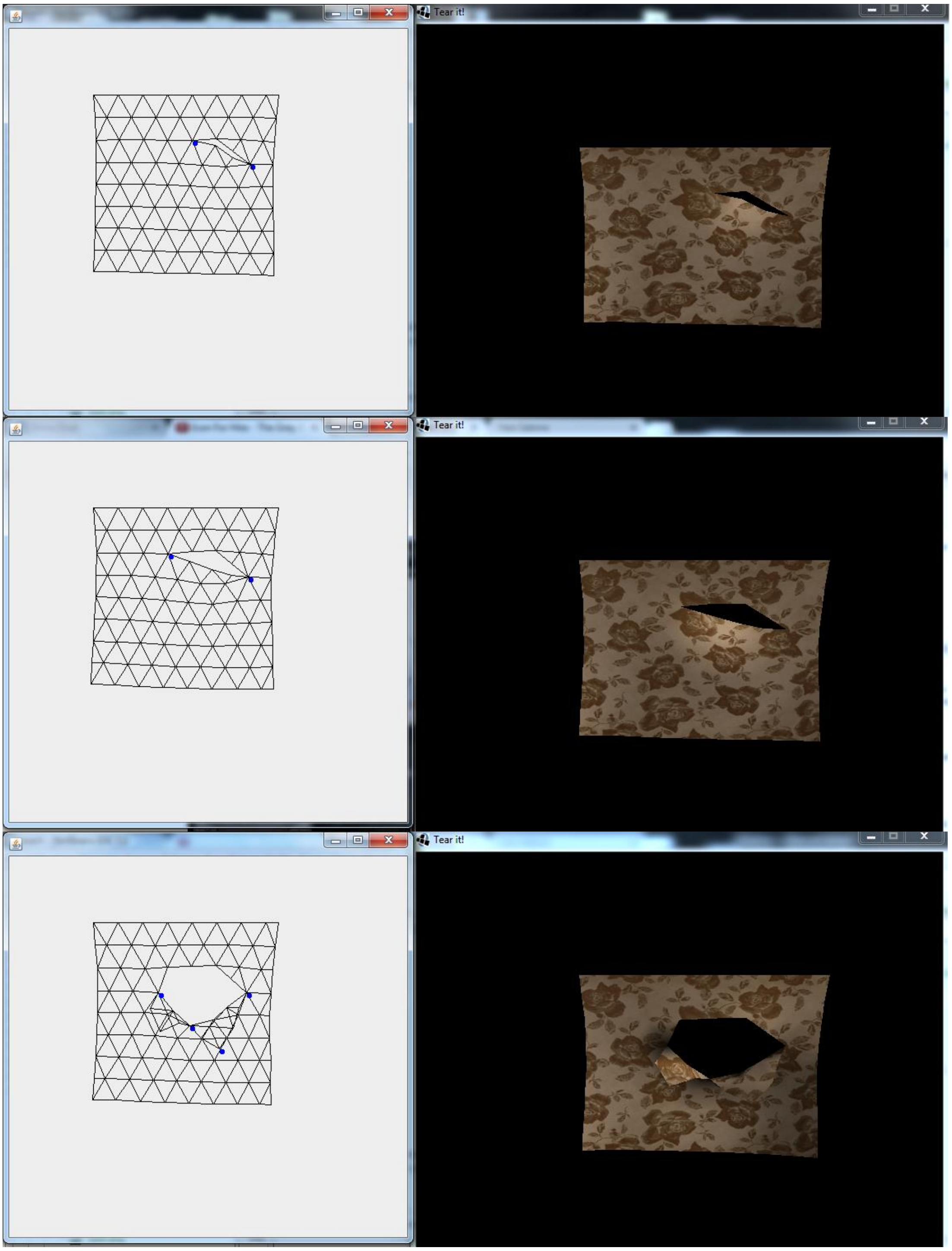

Figure 3.17: Weak point propagation.

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.

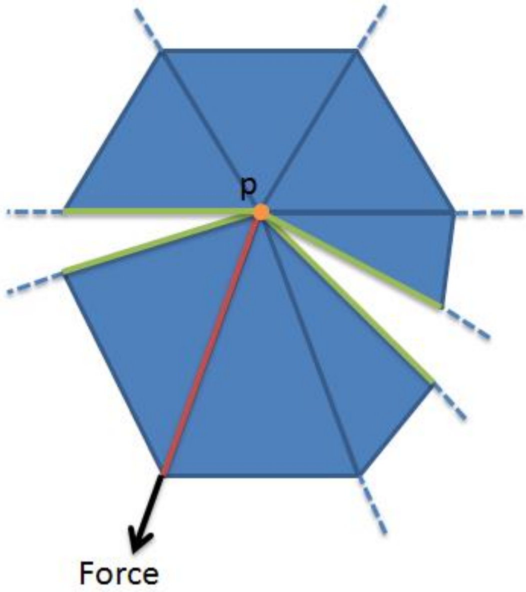

Figure 3.18: An example of one point connection.

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 the springs connected to the point of interest.

We create a new particle p' instead of p and apply a small displacement along the axis of tension to be able to prevent the tear after the detachment by shortening the length of the spring with the tension.

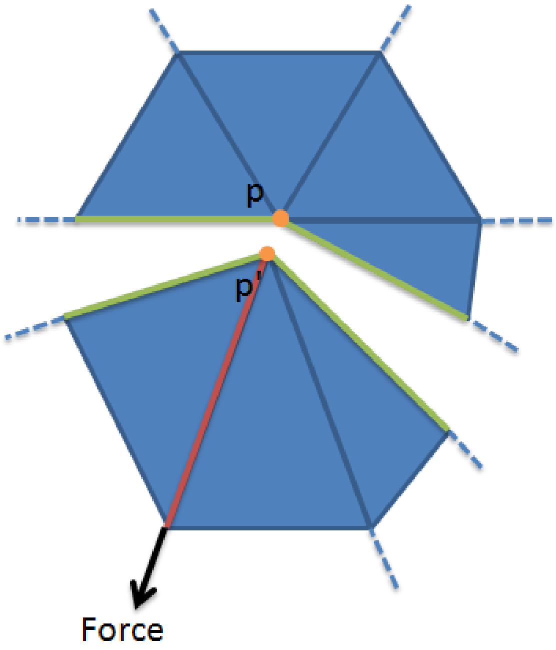

Figure 3.19: Detaching the segments connected with one point connection.

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 the texture integrity after tearing.

In a sheet-based model, particles are placed on the corners of a regular grid and cloth texture is applied onto 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 based model, this is handled by using the indices of the particles. Since that model uses a regular grid, it is possible to reach the texture coordinate on a particle.

We mentioned that the proposed model is not designed to handle structural changes. We need to find a different way to reach the texture positions on polygon vertices.

Also, we must be able able to handle the structural changes and preserve the texture positions on the vertices after the tearing.

At this point, we take advantage of the 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 seperated, as in Figure 3.20, they do not share the same particle object on the seperated side after the tearing. Since the texture position is stored in triangle objects, we still know the texture position on p3 and p3'. Here tx and ty values represent the texture position. They are preserved even after the tearing. While drawing the polygons for m1 and m2, we use those coordinates. As a result, there are not any areas lost or misplaced.

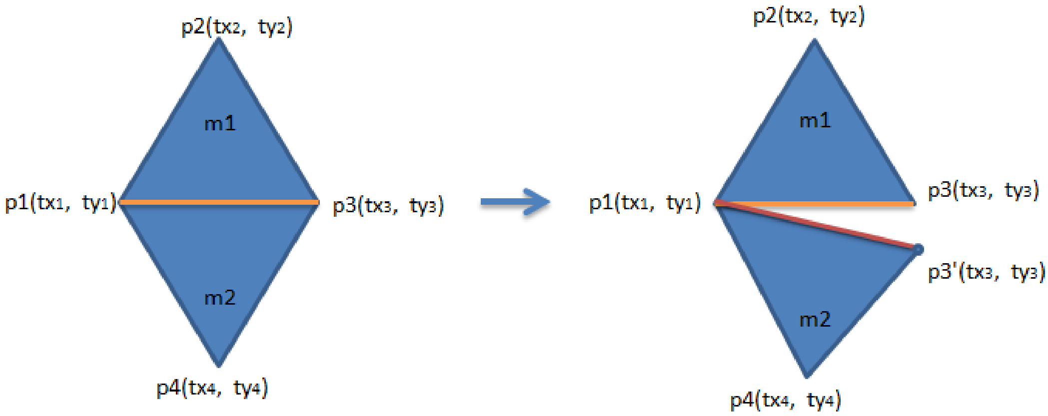

Figure 3.20: Texture position assignment after detaching one side of two triangles.

In the basic tear model explained in 3.2.1, the triangle is divided into two. Old triangle is eliminated and two new triangles are created instead of the old one, as in figure 3.21. 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.

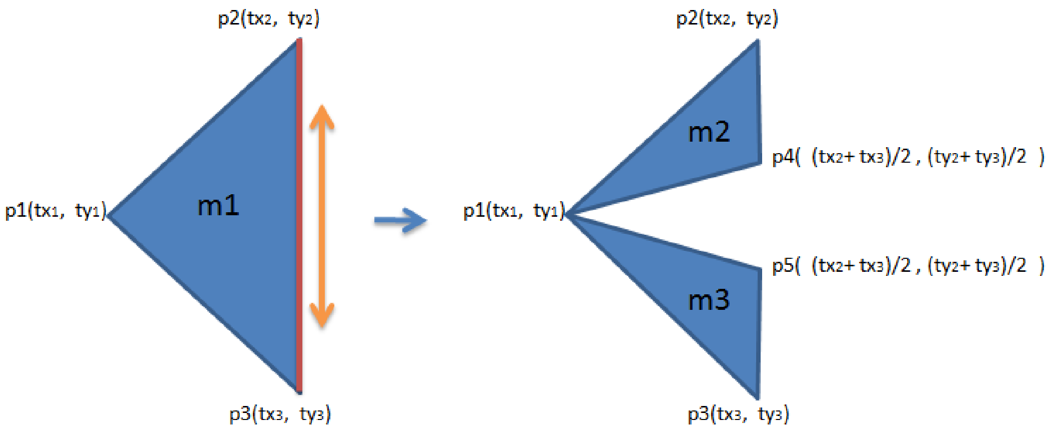

Figure 3.21: Texture position assignment after dividing a triangle.

In the figure above, 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.

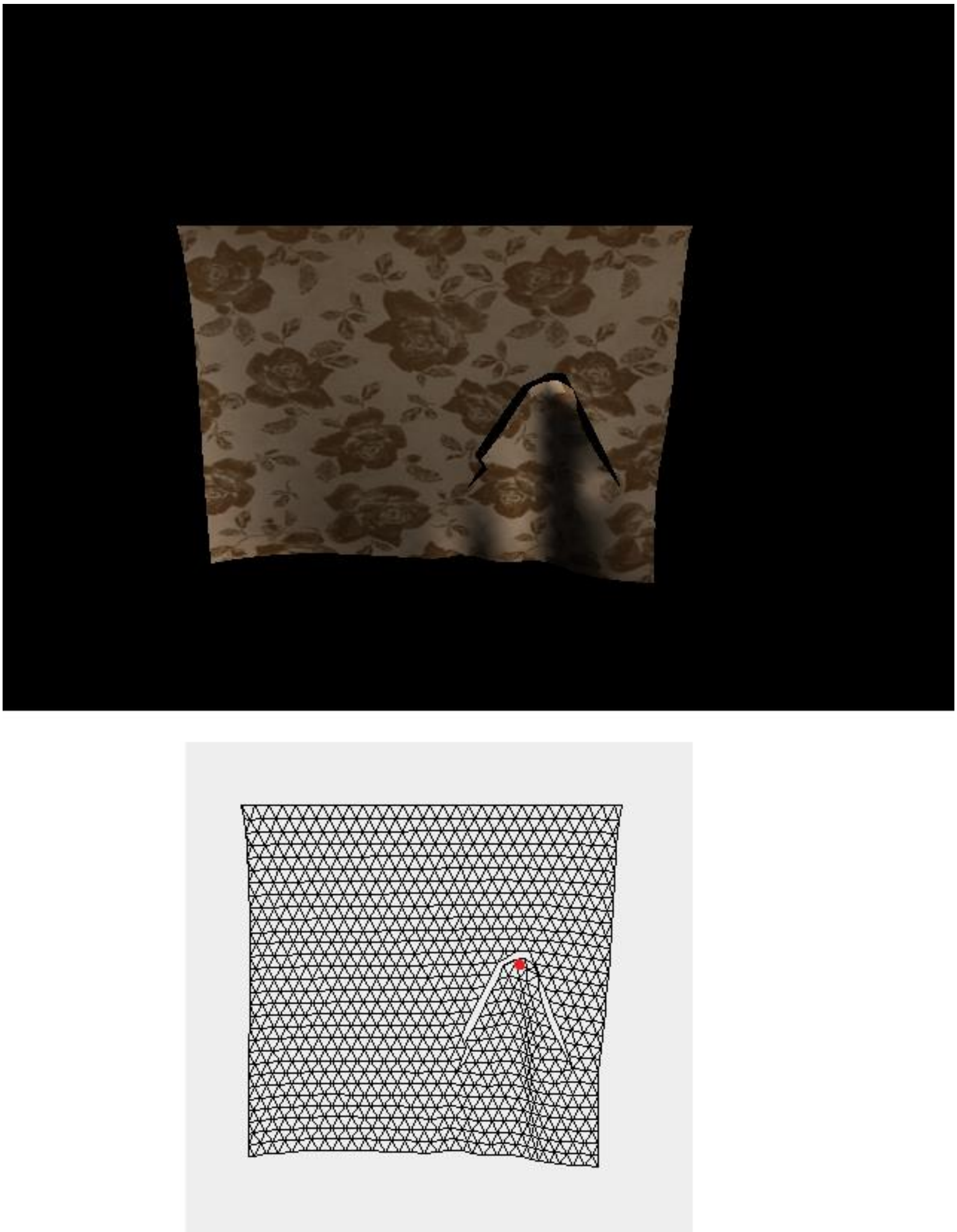

Figure 3.22: Textured, torn cloth

Here is an example from our implementation in figure 3.22. After we pulled and tore the cloth, we dragged the red point back to the place it was connected to. We can see texture positions are preserved correctly after the tearing.

CHAPTER 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 3.22), but our proposed method is also applicable to complex cloth models, like the example shown in Figure 3.2. The cloth is attached from the upper edge. You can drag cloth from any point with 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.

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 as in Figure 3.22.

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 of the pressure applied.

Handling collision with complicated objects is another extensive research area and outside the scope of this thesis. 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 as 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 to the cloth with different sizes and 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 consiered as a small size for this example, and the speed of the ball is 3. In Figure 4.1 it is shown that there are three balls thrown to the cloth. Two of them tore the cloth and the last one is about to tear.

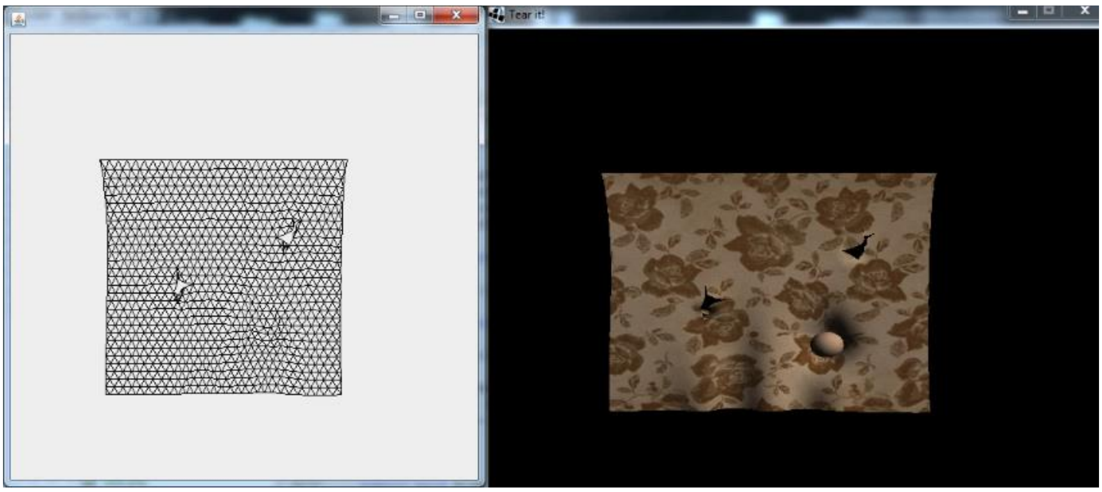

Figure 4.1: A ball with a radius of 0.5 is thrown to the cloth with a speed of 3.

In the second example we increased the ball radius to 1. When we threw the ball to the lower part of the cloth, it passed beneath the cloth without tearing. Cloth slid on the ball as in Figure 4.2.

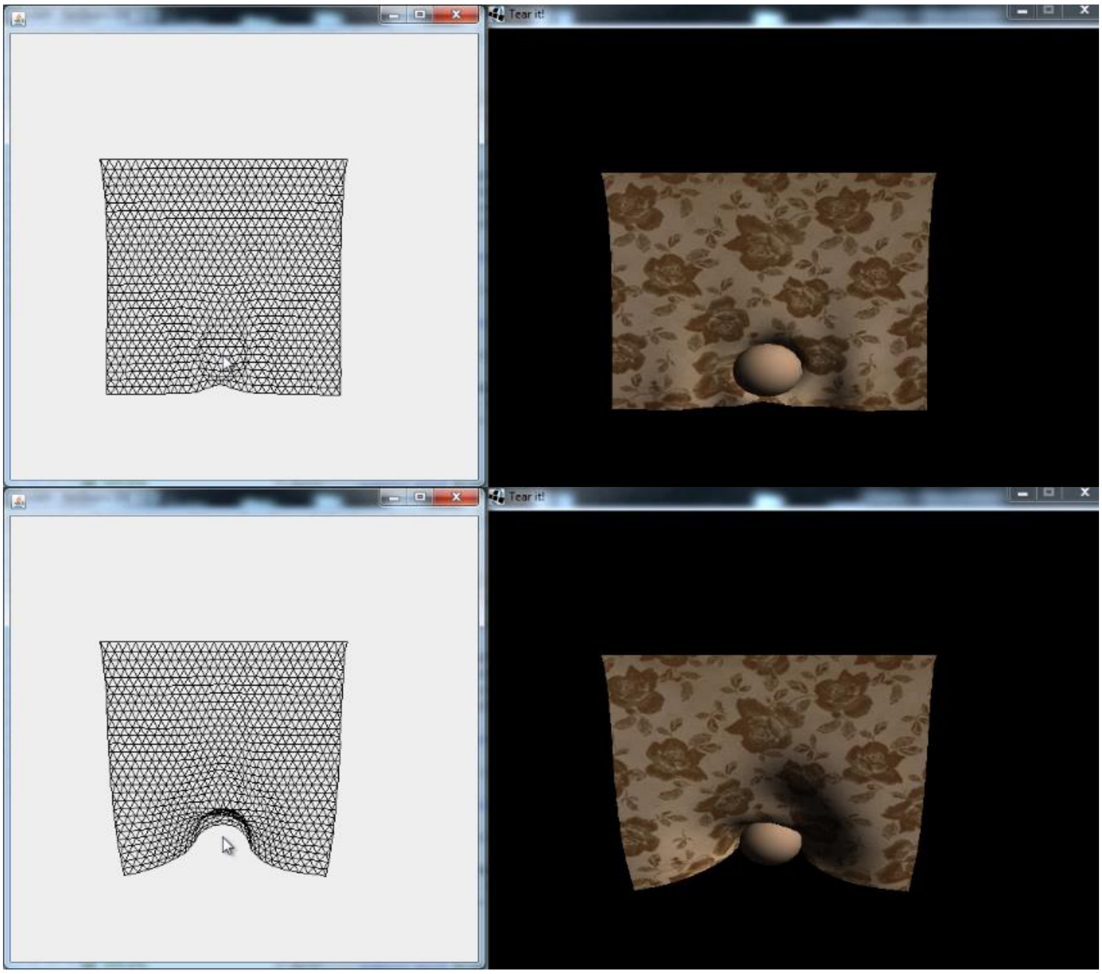

Figure 4.2: A ball with a radius of 1 is thrown to the lower part of the cloth with a speed of 3.

If we throw the ball to 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 4.3.

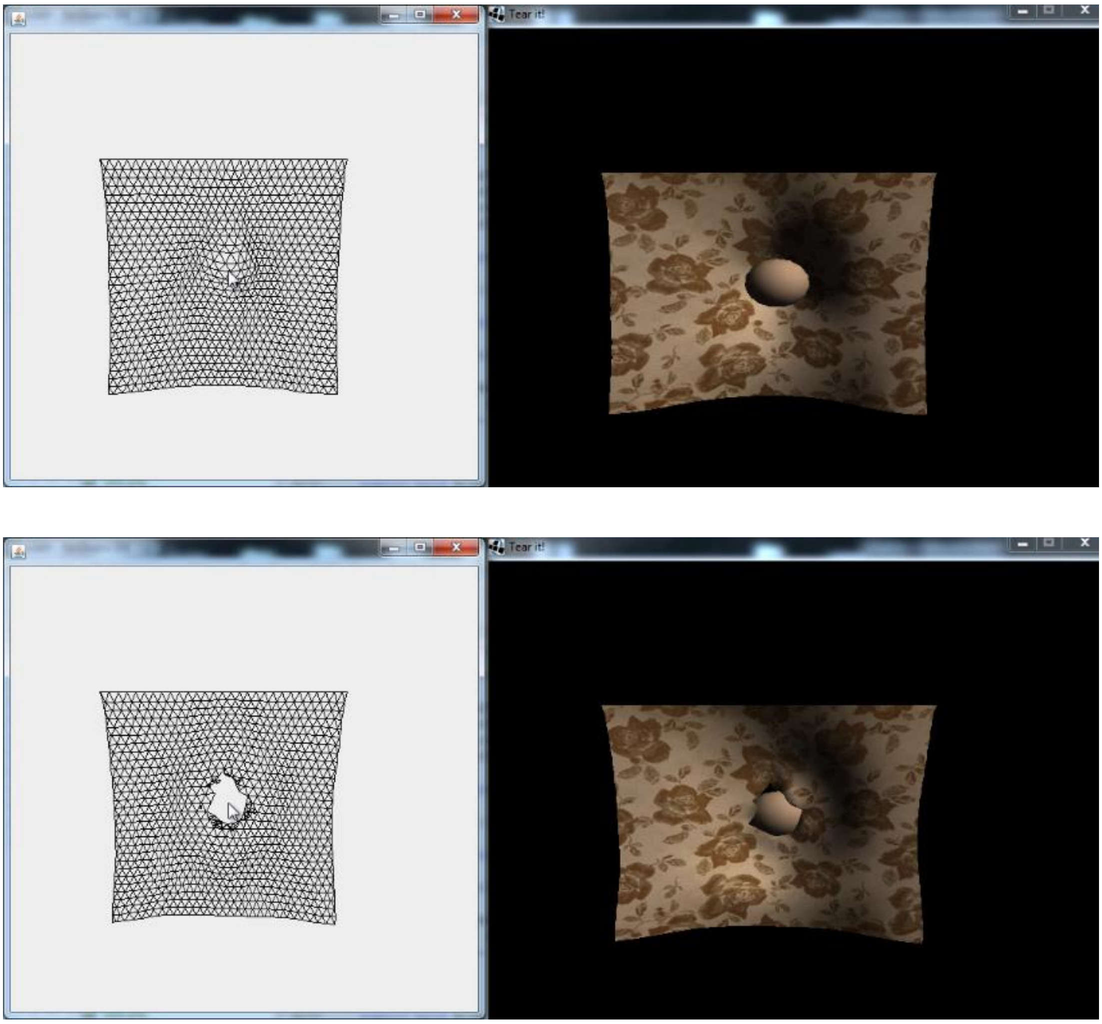

Figure 4.3: A ball with a radius of 1 is thrown to the upper part of the cloth with a speed of 3.

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 4.4.

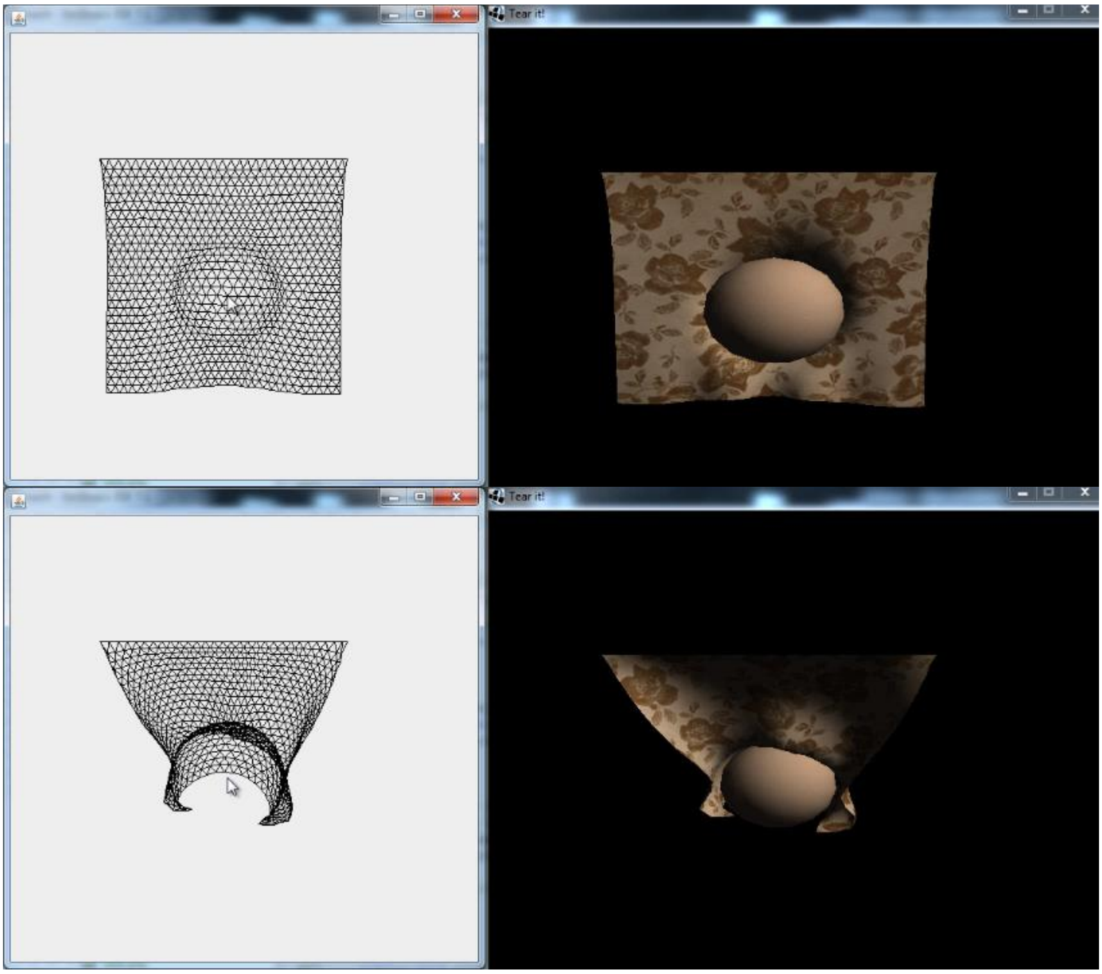

Figure 4.4: A ball with a radius of 2 is thrown to the cloth with a speed of 3.

As a final example, we increased the speed and throw 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 very fast that cloth can not resist the impact and is torn, as in Figure 4.5.

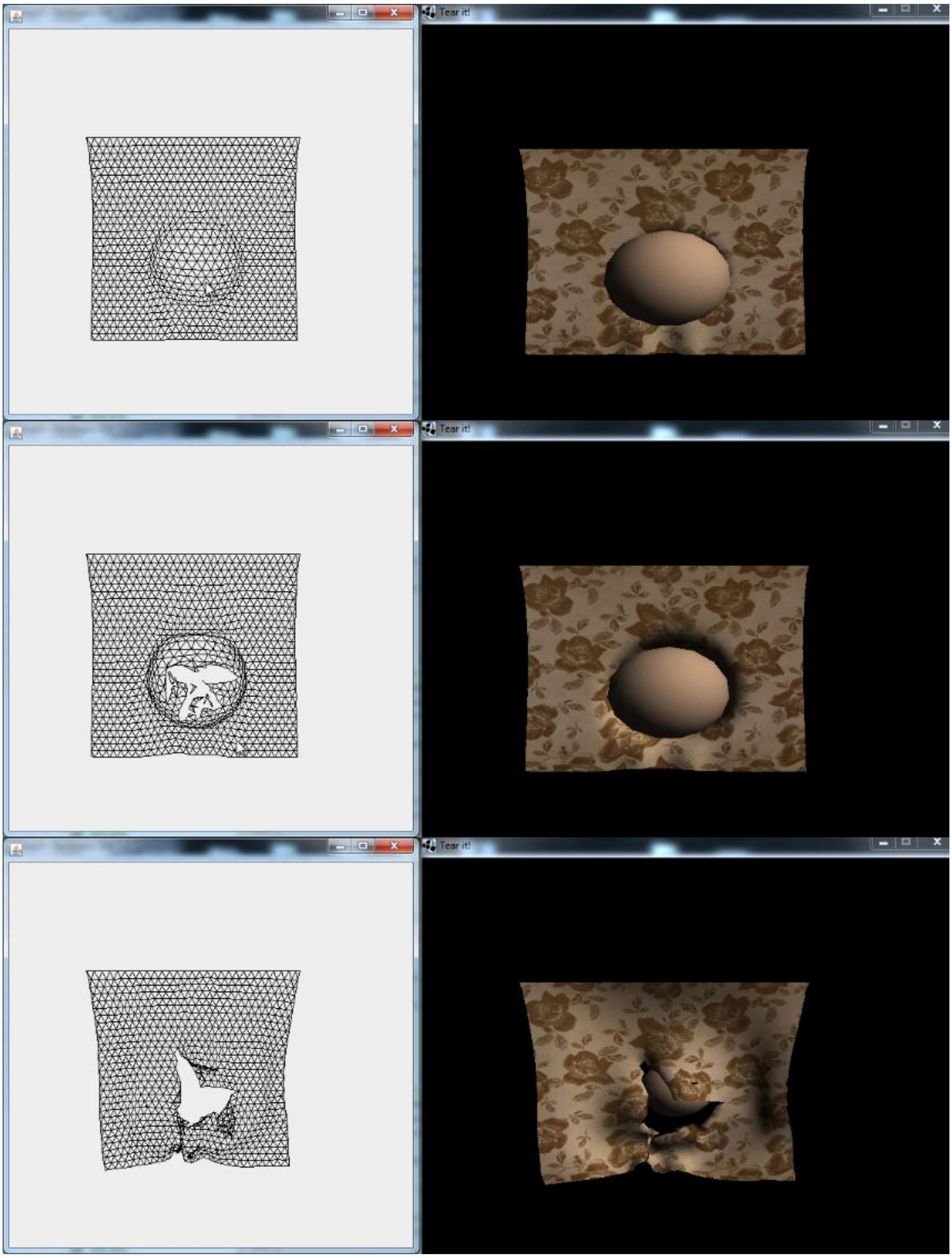

Figure 4.5: A ball with a radius of 2 is thrown to the cloth with a speed of 10.

4.2 Tear Response to Force Direction

Another experiment we made is about the change of the tear propagation according to the direction of the force. According to the heuristic we developed for weak points in Section 3.2.2, 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 in figure 4.6.

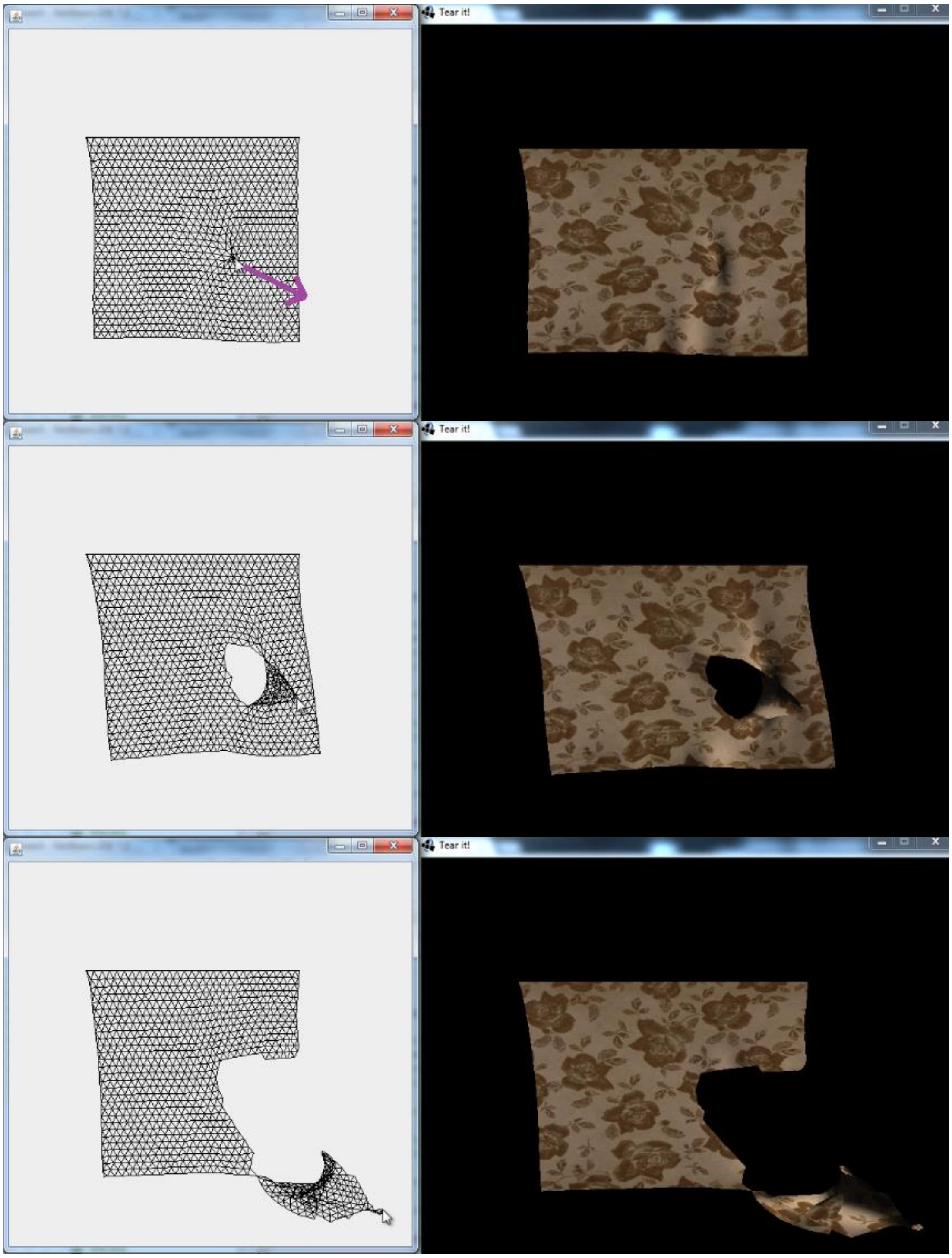

Figure 4.6: An example of the response of the tear to the direction of the force.

4.3 Profiling Results

One of the most important criteria for evaluating the success of a method used for modelling an interactive physical simulation its real-time performance.

Running cloth simulations in realtime 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 realtime systems.

In order to assess performance we ran a simulation for 34 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 onto 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 4.7.

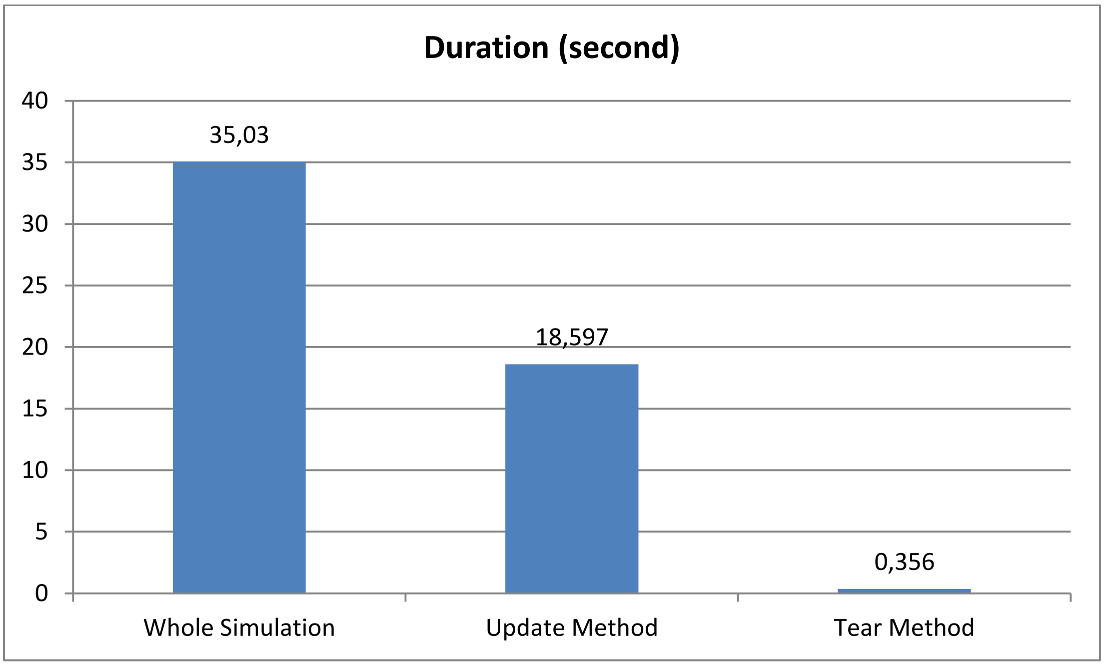

Figure 4.7: Profiling results for a sample simulation run.

The whole simulation takes 35 seconds but our concern is about 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.

CHAPTER 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 thesis was to fill this gap and provide a general solution to this problem, dealing with all the aspects of it.

We tried to provide 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 the 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 the well-known sheet-based model, which is the most common model for many cloth simulations and is also compatible with non-uniform cloth structure.

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 [21]. 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.

Although another area of expertise, creating tearing sounds during the process of tearing the cloth would be a nice complementary study to this thesis. 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 multiresolution cloth models. The parts modelled with higher resolution could be torn in detail and produce more realistic results. By implementing a tessellation algorithm, tearing the parts which have low resolution also can produce more realistic results.

REFERENCES

[1] Metaaphanon, Napaporn, et al. "Simulation of tearing cloth with frayed edges." Computer Graphics Forum. Vol. 28. No. 7. Blackwell Publishing Ltd, 2009.

[2] de Aguiar, Edilson, et al. "Stable spaces for real-time clothing." ACM Transactions on Graphics (TOG) 29.4 (2010): 106.

[3] Kaldor, Jonathan M., Doug L. James, and Steve Marschner. "Efficient yarn-based cloth with adaptive contact linearization." ACM Transactions on Graphics (TOG) 29.4 (2010): 105.

[4] Curtis, Sean, Rasmus Tamstorf, and Dinesh Manocha. "Fast collision detection for deformable models using representative-triangles." Proceedings of the 2008 symposium on Interactive 3D graphics and games. ACM, 2008.

[5] Sifakis, E., S. Marino, and J. Teran. "Globally coupled impulse-based collision handling for cloth simulation." ACM SIGGRAPH/Eurographics Symp. on Comp. Anim. 2008.

[6] Terzopoulos, Demetri, and Kurt Fleischer. "Modeling inelastic deformation: viscolelasticity, plasticity, fracture." ACM Siggraph Computer Graphics. Vol. 22. No. 4. ACM, 1988.

[7] Terzopoulos, Demetri, and Kurt W. Fleischer. "Deformable models." The Visual Computer 4.6 (1988): 306-331.

[8] Hellrung, Jeffrey, et al. "Geometric fracture modeling in Bolt." SIGGRAPH 2009: Talks. ACM, 2009.

[9] Bao, Zhaosheng, et al. "Fracturing rigid materials." Visualization and Computer Graphics, IEEE Transactions on 13.2 (2007): 370-378.

[10] O'Brien, James F., and Jessica K. Hodgins. "Graphical modeling and animation of brittle fracture." Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1999.

[11] Norton, Alan, et al. "Animation of fracture by physical modeling." The visual computer 7.4 (1991): 210-219.

[12] Steve Rotenberg. UCSD, Winter 2005 CSE169 Computer Animation Course Slides.

[13] Bridson, Robert, Ronald Fedkiw, and John Anderson. "Robust treatment of collisions, contact and friction for cloth animation." ACM Transactions on Graphics (TOG). Vol. 21. No. 3. ACM, 2002.

[14] Baraff, David, and Andrew Witkin. "Large steps in cloth simulation." Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998.

[15] O'brien, James F., Adam W. Bargteil, and Jessica K. Hodgins. "Graphical modeling and animation of ductile fracture." ACM Transactions on Graphics (TOG). Vol. 21. No. 3. ACM, 2002.

[16] Provot, Xavier. "Deformation constraints in a mass-spring model to describe rigid cloth behaviour." Graphics interface. Canadian Information Processing Society, 1995.

[17] Ng, Hing N., and Richard L. Grimsdale. "Computer graphics techniques for modeling cloth." Computer Graphics and Applications, IEEE 16.5 (1996): 28-41.

[18] Jain, Nitin, et al. "Multi‐resolution collision handling for cloth‐like simulations." Computer Animation and Virtual Worlds 16.3‐4 (2005): 141-151.

[19] Häggström, Olof. "Interactive Real Time Cloth Simulation with Adaptive Level of Detail." (2009).

[20] Kelager, Micky, Sarah Niebe, and Kenny Erleben. "A Triangle Bending Constraint Model for Position-Based Dynamics." Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS. 2010.

[21] Criswell, Brice, Michael Lentine, and Steve Sauers. "Avatar: Bending Rigid Bodies." SIGGRAPH 2010 Talks.

[22] Thomas, T., Simulation of Cloth and Tearing in Cloth. Retrieved January 7, 2013, from http://clothtear.blogspot.com/

[23] Cloth texture used in sample implementation. Retrieved January 13, 2013, from http://www.keteks.com/Suet.aspx

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü **X**

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : Önal
Adı : Emre
Bölümü : Modelleme ve Simülasyon EABD Oyun Teknolojileri Yüksek Lisans

TEZİN ADI : Cloth Tearing Simulation

TEZİN TÜRÜ : Yüksek Lisans **X** Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. **X**

2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası Tarih