Implementation of Cloth Simulation that can be Used in Games and Animation

by

Navneeth Nagaraj, B.Eng.

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

(Interactive Entertainment Technology)

University of Dublin, Trinity College

September 2016

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Navneeth Nagaraj

August 25, 2016

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Navneeth Nagaraj

August 25, 2016

Acknowledgments

I am heartily thankful to my supervisor Dr. Michael Manzke for his guidance and support throughout my masters study which set up a strong understanding of the subject, which led to the completion of this thesis.

I am also grateful to Dr. M C Padma, HOD of CS&E at PESCE and Dr. K A Radhakrishna Rao, Professor of E&C at PESCE for their invaluable support.

I would like to show my gratitude to my friends Yuanqi Huang, Patrick O'Halloran, Daniel Walsh, Barbara De Kegel, Aneesh Roy, Manuela Vasconcelos, Eugene Kulabuhov for helping me throughout the masters.

Special appreciation to my mom, dad and my sister, who have been by my side throughout my journey.

Last but not the least, I would like to acknowledge and appreciate my friend Seema Suresh, Masters Student at Arizona State University, for spending her time to proofread this dissertation.

Navneeth Nagaraj

University of Dublin, Trinity College
September 2016

Implementation of Cloth Simulation that can be Used in Games and Animation

Navneeth Nagaraj
University of Dublin, Trinity College, 2016

Supervisor: Michael Manzke

Simulation of Cloth in Games and Animation plays a significant role. If the simulation can be taken to a stage where the cloth behaves as real as in the real world then it can be used in many applications. Simulating thin-shell materials like cloth consists of more technical challenges: rendering complex cloth structures, making rendering faster, simulating realistic cloth behavior. This paper includes overcoming the technical challenges, exploring different model techniques, analyzing the drawback of the models and making possible corrections which is then integrated to form a new approach. The simulation of the cloth is implemented [1] concentrating on the parameters that affects the behavior of the cloth. At the end evaluation is done proving that the new approach is promising in giving realistic behavior and can be used in Games and Animation.

[1] https://www.youtube.com/watch?v=rUVmZZtFCLw

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Decades of research led to the development of Computer Graphics which became popular to the mankind in the form of games and animation. The graphics in these field are getting more realistic with the latest technology and more researches are being done to improve the quality of the results. In Games and Animation simulation of Cloth plays an important role. Many researches have been done to make the simulation of cloth look more realistic. Despite all these researches there have not been one technique that has been considered ideal or did act as dominant technique. There are many technical challenges in cloth simulation such as to make the simulation look more realistic at the same time to render it more faster and developing algorithms to produce complex cloth with complex properties.

Adding different fashion to the characters will add a new dimension to the 3D animation. It can be seen that techniques that involve in giving high degree of details like fast rendering, realistic behavior and more stable do have high demands. Decades of research had been done on cloth simulation which has made it quite useful. Animation production companies like Pixar, Disney [Baraff and Witkin, 1998], [Baraff et al., 2003] have made lot of research on the same field to make the simulation look realistic. For example, wrinkles on the cloth and response of cloth to the direction of wind have been improved remarkably along with the advancement in quality of Animation. The algorithms have been optimized such that they provide better results and runs at reasonable speed.

Physically based simulation is an area of interest in Computer Graphics where the results are obtained based on approximations. The approximations are made upto a point which gives satisfactory results at the end of the simulation. All the models used to simulate cloth have used this approach and there are many problems unresolved. There is still wide space available for investigating the techniques to make it better and achieve more desirable results.

1.1 Motivation

Many companies incorporate their own method for the simulation of cloth. For example Square Enix's Rise of the Tomb Raider [Michels et al., 2015] where the character Lara Croft's cloth behavior changes with different circumstances like cloth moves in the direction of wind; the cloth is more stiff when it is in contact with water bodies; snow, dirt and blood being deposited on the cloth and so on. But few of the drawbacks were, the rendering was not same at different angles, few angular scenes had lower visual effects and certain behaviors like dirt or blood washing off the cloth while in water was not seen. It can be seen that there are many techniques to obtain a certain scene but the same technique have many drawbacks. To develop a method that makes the rendering of the scene look real without having many flaws are demanding and is challenging.

Ubisoft [Ubi, 2004] have been developing games with the cutting edge technologies [McAuley and Hill, 2016]. They explained their latest technology that they have used in Far Cry Primal in GDC talk 2015 [Nicholas, 2015]. These techniques are optimized by doing decades of research and frequent analysis of the latest technologies to get more detailed results. Two interesting recent researches [Kavan et al., 2011] and [Wilson et al., 2014] where they explain a mid range cloth simulation in first and a technique that is used in Frozen movie for cloth and hair simulation by Disney in the second are very effective techniques.

All these techniques are remarkable and have provided a base for future improvements with the evolution of technology. It is challenging, interesting and worth exploring the different techniques used for modeling of the cloth and to discover new techniques for the simulation to obtain better results that can be used in games and animation.

1.2 Objectives

The primary objective of this dissertation is to implement and evaluate a technique for simulation of cloth which provides a realistic cloth behavior concentrating mainly on parametric changes and external forces like wind and other external bodies. In order to accomplish this, a number of objectives should be met:

- Extract optimized techniques for cloth simulation.
- Integrate all the techniques and provide a faster rendering result.
- Analyze behavior of cloth with different parametric values and evaluate the system.

The dissertation also includes the current challenges that are being faced while simulating the cloth. The main idea of doing this project is to achieve a technique for simulation of cloth which can be used in Games and Animation.

1.3 Overview

The following sections contain different chapters each explaining different stages. The following Chapter 2 highlights the state of the existing system that is being used and the problems faced by the existing methods and possible solutions for the same. The next chapter which is Chapter 3 contains the models and techniques which are designed by modifying the existing techniques so that better rendering is obtained. Chapter 4 consists the process of implementing the whole model by integrating different methods, this implementation is further evaluated in Chapter 5 and finally the results are analyzed and conclusion is drawn in the last Chapter.

Chapter 2

State of the Art

Movement of the cloth can be modeled in three different approaches:

1. Geometry Approaches [Kunii and Gotoda, 1990], [Weil, 1986b], [Weil, 1986a] : This is the earliest modeling method where force or any energy factor were not considered rather the shape of the cloth was estimated to have an almost accurate value by considering arched primitives like catenaries.

2. Continuum-based Approaches [Feynman, 1986], [Volino et al., 1995] : Using this approach the most accurate simulation results are produced since the approach is based on continuum mechanics; for example if we consider a small portion of cloth, we assign a function which calculates energy for that portion of cloth. The energy function is assigned by considering the behavior of fabrics. The energy function's minimum value is calculated and then the value of fabrics is predicted from it.

3. Spring Connected Particle System [Ascher and Boxerman, 2003], [Baraff and Witkin, 1998], [Baraff et al., 2003], [Baraff and Witkin, 2000], [Bridson et al., 2003], [Desbrun et al., 2000] : In this method some assumptions are made while calculating the forces acting on the cloth. This approach provides a satisfactory result by taking less time. Hence most of the current simulation uses this method.

In this paper more detail is provided for Spring Connected Particle System because that is the technique used for the new approach which is discussed in later chapters. Also, we will discuss about different model techniques used to achieve cloth simulation which is later modified for the new approach.

There exists many different models for the simulation of cloth. For better understanding of the models and to see the difference between the models, we will segregate the simulation of cloth into following five sections: equations to calculate motion, integrator, collisions in cloth, modifications & improvements and rendering of cloth.

2.1 Equations to Calculate Motion

2.1.1 Mass-Spring Damper Model

The motion of the cloth should be governed, the equations used to calculate motion does this work. Let us consider one of the first and simplest technique of cloth model where you take a portion of cloth and will divide it in such a way that it forms a pattern of equally spaced areas with their mass being M. Connect the neighboring areas with springs. Newton's Second Law of Motion and Hook's Law can be used as equations to calculate motion. Further improvements can be made by adding the damping or angular springs. The pattern from this form a grid like structure which plays a very important role to store the cloth. After storing the cloth, we can directly perform the simulation on it. The grid will have many vertices and each vertex will have their own local values such as Velocities, Forces, Positions etc. When the whole structure is formed, the information from each face of the mesh makes it possible to render the whole grid structure as a normal mesh.

$$F_{net}(v) = M * g + F_{wind} + F_{air\_resistance} - \sum_{Springs \in v} k(x_{change} - x_{initial}) = M * a \quad (2.1)$$

Where:

$$M \ - \ Mass \ of \ Vertex$$
$$g \ - \ Gravitational \ Vector \ = \ (0, -9.8, 0)$$

$$
\begin{aligned}
k &- Spring\ Constant\\
x_{initial} &- Initial\ Magnitude\ of\ Spring\\
F_{wind} &- Vector\ Representing\ Wind\\
F_{air\_resistance} &- (-a)\ *\ Velocity(v)^2
\end{aligned}
$$

M can be calculated by assigning a simple value to all the vertices which includes unit value. To get more accurate results, the area of the triangles are calculated and one third of the value is assigned to each of the incident vertex. By doing this we can calculate full mass of the cloth by summing all the areas of the triangle and then multiplying by mass density of the cloth. The gravitational vector can act as arbitrary vector; if time is measured in seconds and distance in meters and the object is on the surface of the Earth, then we can consider Gravitational vector which acts as "up/down" vector to be $(0, -9.8, 0)$. If the magnitude of the spring's length at initial(rest) position is represented as $x_{initial}$ and the magnitude of spring's length when it changes is represented as x_{change} then each spring structure should have these two values stored in them. F_{wind} is a constant function that changes its value periodically. The functions can be $\sin(a * t)$, $\cos(a * b * t)$, $\cos(\sin(7 * a * b * t))$. The value of a is determined by the environment(Usually air) in which it is incorporated and it will have a constant value. Sometimes, this parameter can be used to provide stability to the cloth or to gain certain appearance of the cloth. k is a spring constant whose value plays a very important role. Suppose if the value is chosen to be too low the cloth sags which looks unrealistic [Figure 2.1].

Contrary to that if the value is too high then the cloth will be too tight which again is unrealistic. Furthermore, if the value of k is made too large then then the cloth will have too much stiffness which is worse; for this situation there is a term in mathematics called explosion. So to get more realistic cloth behavior k value should be chosen carefully else more energy will be gained by the system with a function dependent on time. When the system reaches a state with high energy, then there will be an explosion in the system created with all the vertices moving towards infinity.

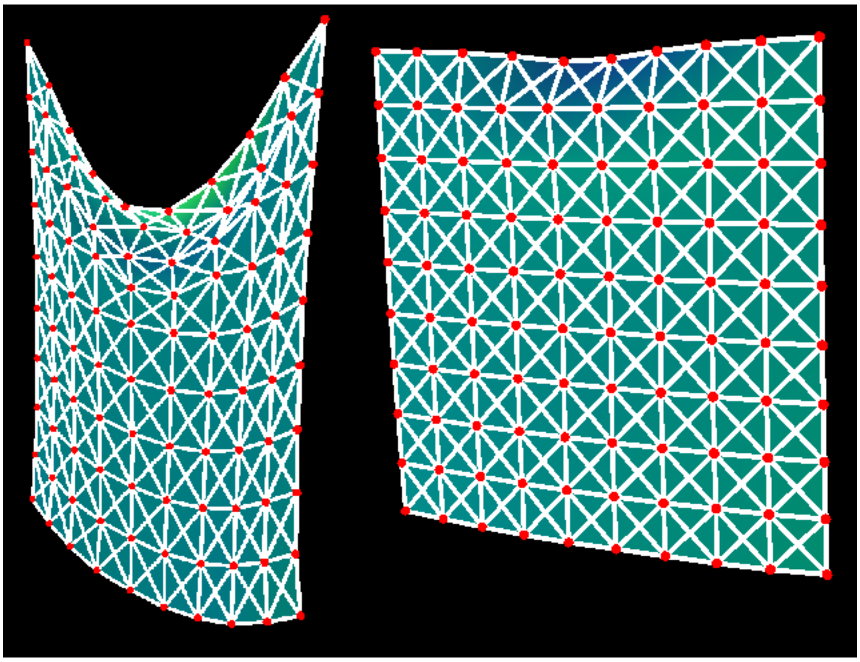


Figure 2.1: Simulation of Cloth based on Different Values of “k”

For the simulation of cloth we will be using Mass-Spring model. Why because in most of the real time applications this model is widely used and also it gives more accurate results. Here is a list of advantages and dis-advantages of Mass Spring Model that have been encountered:

Advantages of Mass Spring Model

1. It is derived from natural tendency and is easier to implement.

2. Calculations are not intensive because of which it is more faster and provides efficient results.

Dis-advantages of Mass Spring Model

1. The model is not derived from theory or observations which makes it less accurate.

2. It has too much of coupling among the springs.

3. Setting spring constant values to obtain desired effect is difficult.

4. Topology of spring network decides the behavior of cloth.

Physical Model of Cloth using Mass-Spring

Mass-Spring method uses a network of nodes of point mass which are connected internally with springs which are mass-less. Each node will experience two forces, one is an internal force which is from mass-less spring and other one is an external force which can be wind, gravitational force and so on. The main idea for the use of spring is that it defines the behavior of the cloth and also dampening forces to maintain stability. Springs can be divided into three different types based on the definition; They are: Structural Springs, Shear Springs, Bend Springs. Each of the spring follows the staggered principle of Hooke's law. By assigning different values to each type of the springs we can simulate different materials such as cotton, wool, etc.

1. The amount of stretch that a cloth can stretch is governed by Structural Spring.
2. The Shear Spring's work is to take care that the cloth does not tear.
3. The cloth should not collapse as soon as it is simulated, to avoid this behavior we use Bend Springs and an interesting property of this spring is that it can be placed in any fashion, since the sole work of it is to make sure that the cloth does not collapse quickly.

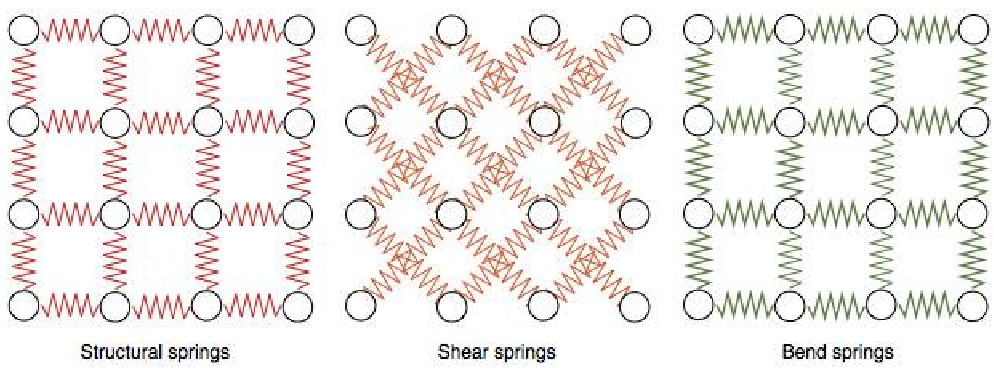


Figure 2.2: Massless Spring Types

2.1.2 Elasticity-Model

The use of Mass Spring Model has certain defects. One important defect is that it is not physically accurate, many attempts were made to correct the values but were unsuccessful. Adding to it, it needs prediction of arbitrary values which determines which of the vertex should be connected to which vertex by the spring and not to mention choosing the value of k. By considering the surface of the cloth and integrating all the energy functions a more accurate model was designed called Elasticity Model which includes the following properties:

- There exists a resistance in triangles and edges while making changes to their size. So energy is required if we want to compress or expand them.
- Bending is resisted by the edges. So energy is required if two adjacent faces needs to be bent.
- Deformation is resisted by triangle. So energy is required in deforming or shearing a triangle.

Let us consider a big vector 'V' which represents all the necessary variables in the system which includes location, velocity of all the vertices present in the system and available degrees of freedom. If the existing state of the system is taken as an energy function '$E(V)$' we can calculate the equation of motion for a vertex at position (x, y, z) as:

$$F_{net} = \left\{ \frac{\partial E(V)}{\partial x}, \frac{\partial E(V)}{\partial y}, \frac{\partial E(V)}{\partial z} \right\} \tag{2.2}$$

Evaluation of the above equation[Equation 2.2] is complicated. The derivative in the equation should be calculated analytically. If any attempt is made to calculate it in numerical way, then we will be reducing the energy function from $E(V)$ to $E(a)$. Which in turn gives:

$$\frac{\partial E(a)}{\partial a} = \frac{E(a + \delta a) - E(a)}{\delta a} \tag{2.3}$$

It can be seen that to calculate the value of energy function $E(V)$ a longer time is needed; because iteration over all the edges, vertices and faces takes place and also the summation of individual energy. But when we consider the energy function with a, it can be observed that the local area (all the incident edges and faces, called the one-ring of the vertex) affects the energy. Thus, to maintain our technique to be $O(n)$ when we compute the derivation numerically, the calculation of $E(a)$ should be done by taking into account the one-ring vertex's energy.

To summarize, there are few challenges during the implementation of this technique, even though it gives accurate results in theory, in real time the results obtained from this model is some times better and some times worse when compared to mass-spring-damper model.

2.2 Integrator

After deciding the cloth model that we have to implement(Mass-Spring Model), the next step is to integrate all the equation that can calculate motion. Assuming that our model follows Newton Laws of Motion, at every vertex we will have a velocity and position defined for each time frame t. Further, the derivative dv/dt gives us acceleration of individual vertex at a time t. What we want now is the velocity and position in the next time frame.

2.2.1 Explicit Integration Method

Euler's Method

There are many methods for Integrating. Euler's method is one of the simplest.

$$V_{t+\Delta t} = V_t + \Delta t \left(\frac{dv}{dt} \right)_t \tag{2.4}$$

$$X_{t+\Delta t} = X_t + \Delta t V_t \tag{2.5}$$

The use of subscript t indicates that dv/dt is evaluated at time interval t. The time-steps (The smaller the time-step, more accurate the results but the time to compute is slower.)that will be taken is referred as Δt. The above method can be derived as:

$$v_t = \frac{dx}{dt} = \frac{\Delta x}{\Delta t} \tag{2.6}$$

$$\Delta x = \Delta t v_t \tag{2.7}$$

$$X_{t+\Delta t} - X_t = \Delta t v_t \tag{2.8}$$

The velocity can also be implemented in the same way only that there is a disadvantage. It is observed that it tends to affect all the variables to increase expeditiously to ludicrous values by producing greater amount of positive feedback, even if the time-step value is small. All explicit integrators have this explosive behavior; the name explicit was derived from the fact that the state of the system at interval $(t+1)$ is computed by considering only the state at interval t. The explosive behavior is one of the key challenges and is a headache to solve easily, a huge amount of work is needed to deal with this problem. One of the way I believe this can be solved is by adding enough damping to the scene (damping of springs, resistance of air, acceleration damping, etc.) by doing this the amount of energy in a single time interval decreases and the feedback will never take place. There is another way in which this can be solved; by adding one vast array to the cloth model. This helps in preventing the explosive behavior. If one does not want to alter the model but still wants to solve the issue then one has to use an Implicit method.

Runge Kutta

Euler's method not only have explosive behavior, but also is not very accurate. After detailed analysis it was observed that as the value of time-step keeps reducing, the error reduces gradually. So use of higher order derivative reduces the percentage of error and gives more accurate results. There are many methods available, but one of the most commonly used method is Runge-Kutta method. By using the Runge-Kutta method one can derive upto N-terms. But for many reasons, it is considered that derivative of 4th order is optimal, since the error at this time-step is very small and also it guarantees that the integrating error at this time frame is proportional to the fourth power of the time frame (it should be noted that the fifth derivative is not proportional to fifth power). This algorithm does not need too much of data, all it needs is the first derivative, from this the consecutive derivatives are obtained by doing calculations. The dis-advantage here is that, irrespective of the process used to compute the derivatives of higher order, it is expensive; though we get accurate results, the system still explodes, only that it takes time to explode.

Verlet Algorithm

The beauty of Verlet algorithm is that, without knowing anything about the velocity terms, it does compute the velocity by taking into account the position at previous and existing time-step.

$$X_{t+\Delta t} = 2X_t - X_{t-\Delta t} + \left(\frac{dv}{dt}\right)(\Delta t)^2 \tag{2.9}$$

Another important property of this algorithm is that, similar to Runge-Kutta's fourth order derivative, this method is also accurate at fourth order. This method is more accurate, implementation is easy and does not need to know velocity.

2.2.2 Implicit Integration Method

Euler's Method

Euler's Implicit integration method takes existing state's variable and the derivative of the next time frame and calculates the state variable of the next time frame. The analogy for Euler's implicit method is impeccable.

$$V_{t+\Delta t} = V_t + \Delta t \left(\frac{dv}{dt}\right)_{t+\Delta t} \tag{2.10}$$

$$X_{t+\Delta t} = X_t + \Delta t V_{t+\Delta t} \tag{2.11}$$

If observed carefully, it can be noticed that t is changed to $t + dt$ in the subscript when compared to Euler's explicit method, from this we can say that all we need is $v(t+1)$; using this new value of velocity the term $x(t+1)$ can be calculated. The catch is, computing the new velocity. In either Elastic or Mass-Spring model the following is necessary: considering a giant vector 'V' which holds all the positions and velocities as $6m \times 1$ matrix (m being the number of vertices). Linearization of the equations should be done so that dv/dt at interval t can be represented as:

$$\left(\frac{dv}{dt}\right)_t = QV_t \tag{2.12}$$

Here 'Q' is massive $3m \times 6m$ matrix which represents the linear relationship between the changes in state and velocities of the system. This value will be substituted into Euler's implicit equation dv/dt from which velocity for the next time frame will be calculated. Nonetheless, inverting the giant matrix 'Q' is required, to achieve this linear conjugate gradient descent comes into play. Globally, this technique is very hard to implement, since it produces negative feedback there will be reduction in energy which in turn reduces explosion behavior. Further, this helps to use time frames with higher value without adding much dampers. It is observed that this technique is an expensive technique since the time frames with bigger values requires inversion of matrix which again is time consuming technique. When bigger time frames are considered the relationship between the faults and time frame is linear, the approximation of the algorithm is very bad for integrating cloth models.

Higher Order Implicit Algorithm

Is there an analogy that uses Runge-Kutta derivative in implicit methods? The answer is there exists, and it is seen that it produces a stable system and produces good errors, but I have never seen any model using this system because the first order derivative is very difficult to implement for anyone.

2.2.3 Semi-Implicit Integration Method

Symplectic Euler's Method

Many algorithms were brought up by settling between explicit and implicit methods. One of which is Symplectic Euler's Method. The equations for this technique are:

$$V_{t+\Delta t} = V_t + \Delta t\left(\frac{dv}{dt}\right) \tag{2.13}$$

$$X_{t+\Delta t} = X_t + \Delta t V_{t+\Delta t} \tag{2.14}$$

The reason it is called semi-implicit is because the velocity is calculated using explicit technique, but the implicit technique is used to calculate the new position. Doing this, we have decrease in either positive or negative feedback which results in better stability, and less cost in time complexity. On average this technique conserves more energy which is a greater advantage. The phase and the motion that occur because of oscillations are not conserved: as a result of this in cloth, out-of-phase circulation occurs in all regions of the mesh, overall this is not a preferred technique for modeling of cloth. Higher orders can be used only that there exists more accuracy with the same defects.

2.3 Collisions

The one thing that cannot be avoided are collisions; It is very hard to deal with especially in the cloth. To simulate the collision accurately to that of a real behavior is a nightmare. Many researches have been done on cloth and none of them have said that they have implemented an ideal collision technique. There are many challenges faced here such as hacking a solution with a reasonable plausible results or guaranteeing that the cloth does not intersect with itself or with other objects. [Baciu and Wong, 2004], [Baraff et al., 2003], [Mezger et al., 2003], [Bridson et al., 2005], [Eischen and Bigliani, 2000], [Bridson et al., 2002] These papers have proposed a complex technique which claims to have solved the challenges. They have the satisfactory results for visual appearances. To approach the problem of collision let us split it into two possible cases cloth-and-cloth and cloth-and-object. Before we proceed to the detailed discussion about them, there was a significant work done by others which are the backbone of the new techniques, for better understanding we will have a look at the prior techniques.

Objects should be represented in a particular manner so that collision is detected, or we can say that the technique used to detect collisions mainly depend upon the way in which the objects are represented. One of the common technique used by many is representing the cloth and object(human skin, chairs..)as triangle meshes; by doing this, collisions can be detected by identifying the vertex of the triangle which encounters collisions and detecting the edge to edge collisions between the moving meshes. This technique is expensive. To shun the $O(n)^2$ comparisons, researches started using bounding volumes. There are many available bounding volume techniques. Some of the most commonly and widely used techniques include OBB (Oriented-Bounding-Boxes), AABB (Axis-Aligned-Bounding-Boxes), Bounding Spheres. Further, to reduce the number of calculations the bounding volumes are organized in a hierarchical fashion. If we consider a two-dimensional clothing system which contains a mesh structure and if we repeatedly segregate the mesh by examining the respective co-ordinates then we will have a BSP (Binary-Space-Partition) tree for that cloth model.

One technique for handling cloth-cloth collision by [Bridson et al., 2002] was successful for rendering images and it was a non-real approach. Even complex structures like wrinkles and folds were handled by considering the collision, friction between them. In each time-step there will be a linear movement of cloth vertices was the assumption. The collision detection between the edges and the vertex of a triangle was solved by a cubic equation [Appendix A]. When there is a contact between the meshes in the cloth there will be repulsion forces along with static and kinematic frictional forces which can be calculated with simplified techniques. This technique is not incorporated in real time applications because the time complexity for this method is expensive.

2.3.1 Cloth-Object Collisions

Of the two collisions this is the easier one to implement. During implementation certain assumptions were made which are: all objects in the scene are solid objects, are independent of their motions, their position and time are fixed and is known. Geometric primitives are used to represent objects in the scene which may be cube, sphere, cylinder, etc. or the objects can be inner portion of the mesh.

Physically Correct Model

The following conditions are to be met for cloth-and-object collision if the model is to be called physically correct:

- Calculating the recommended $x(t+dt)$, starting at interval t and position of cloth being $x(t)$.
- Considering the positions of every face in its initial and final state; If in this time interval i.e t and $t + dt$, the face has encountered any external object, the actual time when the two bodies collide should be computed. If there was any prior contact with a surface and is not in contact now, the connection with the surface should be cut-off.
- Finding the time when the first similar collision occurred. Taking the scene back to this interval, the face should be marked when it has encountered contact with the surface. When this is done, all the frictional forces (static and kinetic) should be computed - along with carefully administering the normal vectors of the surface with which it is in contact with.
- Restart the simulation from this recent time frame, and rerun.

Implementation of this technique is very bothersome and is indeed slow, because it needs constant attention of observing collisions, backing-up and restarting the process in order to proceed. The system does not support flexible time skip, which means we have no option other than to change the time-step in small intervals. There are some researchers who claim to have solved the problem with the frictional forces, but most of these techniques fail in backing up, alternately they make predictions for the appropriate area of the face.

Easy to Implement Model

In real, all the methods discussed to handle collisions are troublesome; if we try to detect the collisions in-between the time-steps then we have a very large volume in which we have to detect the points of collision based on the path taken which is annoying. I have a model which I find easy to implement. Firstly, we compromise with the intermediate time-steps and also we do not consider edges or faces. By doing this my model does not detect collisions in two situations [Figure 2.3]. Red being the initial position, blue being the new position.

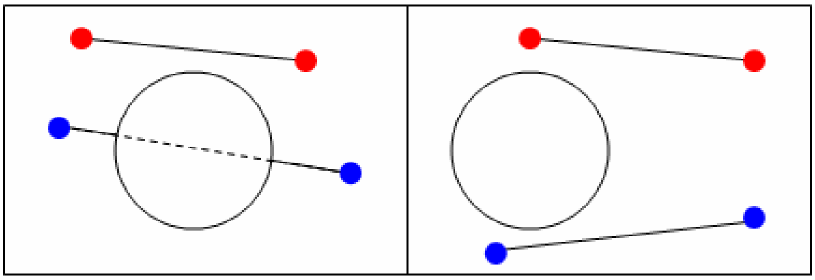


Figure 2.3: The Two Compromised Situations

My logic in the algorithm is that if the position of the vertex in the later time-step that is computed appears to be inside one of the body then the collision is detected. The response can be either of the two: repelling the object in any fashion or to change the position of the vertex to a value obtained in previous time-step. If we take an arbitrary position in the scene, then each point on the exterior face of the body is associated to that point in its own way. For example, if we consider a ball and a known position, then the cross-point of the ball and the line from the center of the ball to the known position appears to be the point closest to the surface which is associated to the known position. These known points are easy to calculate in geometric objects like ball. Here, instead of shifting the positions back to its initial point, we move it to repelling point. By doing this the simulation will look like a silky surface, and more refined; vertices will be growing in number and faces will be reducing, so the point of considering only vertices works very well.

2.3.2 Cloth-Cloth Collisions

It is not the nature of the cloth to pass through itself. One of the easiest way to avoid this is to add abhorrent force when the areas of the cloth are about to come in contact, making them separate with the force. This method is awful and is not the correct technique to do in practice. We will look at more powerful techniques.

Computations are made when there is change in positions at the same time it is determined if there were any intersection. After this we have to retreat to the earlier time frame, where we change the values of the velocity such that there will be no occurrence of the collisions later on. By doing this we can assure that there will be no collisions. This is a very effective technique and is not hard to implement. The only dis-advantage I encountered during the implementation of this is that detecting all the collisions is expensive and it did slow down the algorithm, adding to it there were disturbance to the geometric-structures.

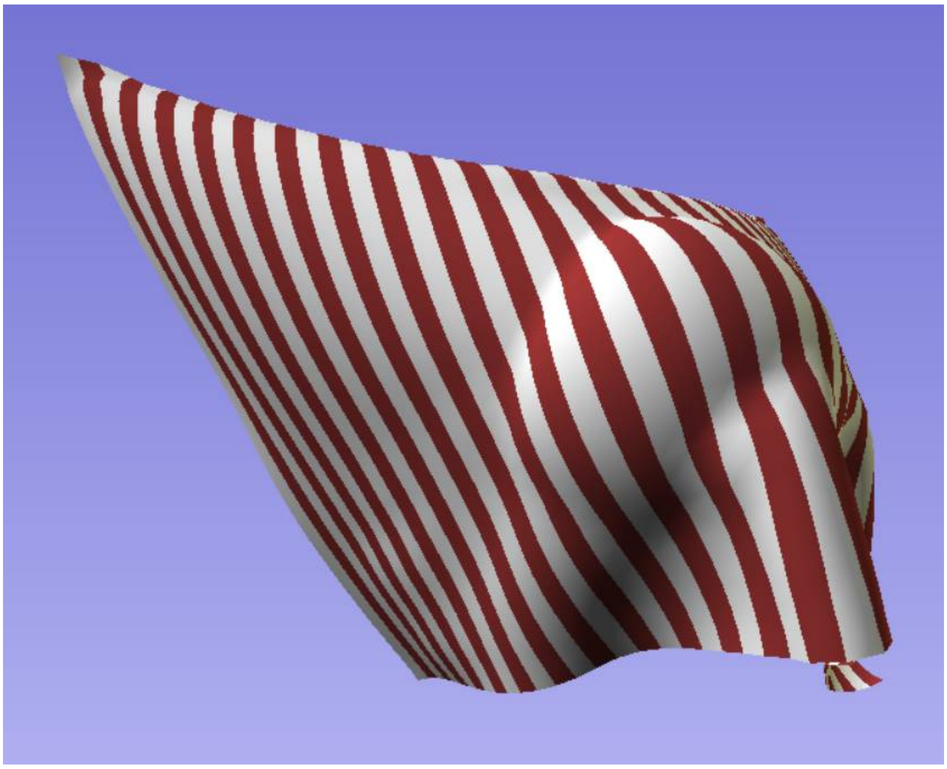


Figure 2.4: A Cloth with Fixed Two Corners Contacting with Geometric Object

Cloth is considered as a connection of marbles in my algorithm which forms a cluster. Imagining every vertex having a marble at the center, these marbles are not in contact if a spring connects the related vertices; by doing this we will be able to set values of the radius to be greater than the length between 2 vertices. This process covers the entire cloth's surface, if in the time frame between t and $t+dt$ there is no intersection between the marbles then there is no cloth-and-cloth collision. Now for the update method which is easy to implement: if marbles of two vertices are in contact (magnitude of the distance between the vertices is less than two times the radius of the marble) we will change the vertices to previous position which results in occurrence of no collision (we do not change the time-step). Velocities can be given different values to get bouncing effect, but it is convenient to give zero value for new velocities. Overall this algorithm is promising and works very well and is not difficult to implement compared to other techniques.

2.4 Modifications and Improvements

2.4.1 Maximum Stretch

We have seen that there will be an explosion in the cloth system at certain situations, this can be controlled by the maximum stretch. The understanding is simple: if two connected vertices are stretched to a point of maximum ratio compared to their positions at rest (say 120% of which 20% is stretch) then moving the vertices so that the stretch is exactly 20% is the idea. Similar to that, if the vertices are to be compressed to a certain value (say 93%) then placing the vertices at a position so that the compression is exactly 93% is the key. Now that we know to goto the previous time-step where we update the positions, we should make sure that for each time-step we iterate all the associated vertices (stretched or compressed), edges so that we maintain the limits between 93% and 120%. The reason to iterate every edge is because moving the next edge causes previous edges to violate the stretch conditions we have set. The big advantage here is, even if we do not successfully update all the edges, the system does not explode and the cloth remains stable. Another plus point is that we can use an implicit integration method (verlet) without worrying about small time-steps and also we need not worry about small values of k; because the maximum stretch values act as infinite stretch and low stretch for high and low values of k respectively. This overall process results in producing realistic cloth behavior because cloth stretches easily for small changes and once the tension is reached it becomes difficult. Lastly, if we use an explicit cloth model then changes at certain node 'P' does not affect another node 'Q' until the next 'N' time intervals provided they are certain edges apart (N edges). If there is an extreme movement at a node, then it can cause changes to the entire cloth in a single time frame which can be used if any attempt is made to change the vertices repeatedly which provides better results.

2.4.2 Bending Springs

Till now we have seen that the vertices are connected to the one's closer to them. But if we connect two vertices that are far apart then the strength from these springs can be considered as bending force which helps the cloth to maintain more stability. By doing this we can eliminate the unnecessary wrinkles on the surface of the cloth when simulated at high resolution. More details about bending is discussed in Chapter 5.

2.5 Rendering of Cloth

2.5.1 Sub-Division Method

The mesh is divided to form triangular structures which is used to render cloth structure. I have used RGB coloring scheme to detect the collisions and to keep track of the orientation of the cloth. The surfaces is sub-divided using [Kazakov, 2007] which helps in rendering low-resolution cloth and can be modified to obtain a high detailed refined cloth.

2.5.2 Quilting

Different cloths have different properties, few cloths like cotton, wool have visible thickness. To obtain these fabric properties, we take output from a thin meshed cloth and create a mesh that produces the required thickness and the algorithm to do this is simple. Create a function $f(a, b, c)$, then we have:

$$f(a, b, c) = d(a, b, c) - T \tag{2.15}$$

Where:

$$\begin{aligned} d(a, b, c) \;&-\; Length\ of\ the\ Shortest\ Line\ fromthe\ Point\ (a, b, c) \\ &\quad to\ the\ Surface\ of\ the\ Cloth \\ T \;&-\; Quilt's\ Thickness \end{aligned}$$

When we have a condition $f(a, b, c) = 0$, then all the points in the system will have a distance T. After computing the values for all the points by using function (a,b,c), we use marching cubes [Lorensen and Cline, 1987], [Wang and Olano, 2015] technique that gives results based on triangular approximation. This technique divides the system into equally spaced cubes then places the function $f(a, b, c)$ at every corner and creates a triangle by considering values of $f(a, b, c) < or > 0$ as vertices.

The implementation of the techniques discussed till now can be seen here [1]. It includes the implementation of individual techniques, modifications added to the techniques wherever necessary and the resulting simulation. At the end there is a simulation of the integrated model, the techniques of which will be used later on.

[1] `https://www.youtube.com/watch?v=KEbnjVPttJ8`

Chapter 3

Experiments

3.1 Overview

In all the experiments carried out by different researchers few of them gave interesting results. Now that we have discussed the techniques and implemented them with necessary modifications and corrections wherever necessary, we know the advantages and dis-advantages of individual technique. Now we will integrate the advantages with some more new techniques and implement much more realistic cloth system that can be controlled by parameters. Before proceeding to the implementation we will discuss about the design of new techniques that can be incorporated in our system.

Techniques that include physics in it are used by those who wish to have more realistic behavior in the visualization and also physical characteristics, which is one of our objective. Here we will see the different techniques that can be used for the simulation of the cloth in which each model concentrates on the certain aspects of the cloth. We will have a look at the experiments that can be done in different models, so that we analyze and later add the necessary techniques for our cloth model.

3.2 Cloth Model Experiments

3.2.1 Draped Cloth Model

C. Feynamm [Feynman, 1986] developed a technique which can be used for the cloths which have loose folds. His technique pictured cloth in 3-D space using a 2-D grid structure. To achieve the property, the equation is as follows:

$$E(P_{i,j}) = k_s E_{elastic,j} + k_b E_{bend,j} + k_g E_{gravity,j} \tag{3.1}$$

Where:

$$\begin{aligned} k_s &- elasticity \\ k_b &- bending \\ k_g &- density \end{aligned}$$

When the cloth's energy is at minimum state it is observed that, the cloth sets to a shape which can be called as final shape. The equation above was derived based on the same observation and was also influenced by the theory of elastic plates, the explanation of the theory is beyond the topic of discussion.

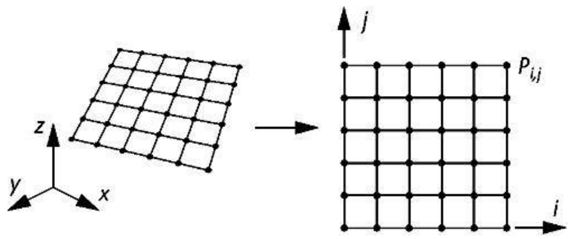


Figure 3.1: 2D Representation Grid Structure of 3D Cloth

The above image[Figure 3.1] shows the two dimensional representation of grid structure of three dimensional cloth. Here the energy at every point is calculated by considering the eight points which surrounds it. The position of the point which has minimum energy is calculated by a technique that takes less time to compute it. Multi-grid structure technique is used in order to make the execution of finding lowest energy state faster.

3.2.2 Provot's Model

Provot way of modeling the cloth [Provot, 1995] is unique. He used the the concept of spring-constrains. This limitation makes it not possible to cover up solid objects with cloth.

Newton Laws of Motion is used in this model:

$$F_{external}(x, y) + F_{internal}(x, y) = ma(x, y) \tag{3.2}$$

Where:

$$\begin{aligned} M &- Point\ Mass\ at\ (x, y) \\ a(x, y) &- acceleration \\ F_{internal}(x, y) &- Internal\ Forces\ at\ P(x, y) \end{aligned}$$

We must use internal forces here which will give more detail to the behavior of the cloth.

$$F_{internal}(x, y) = -\sum_{(m,n)\in R} K \left\{ \overline{P_{x,y}P_{m,n}} - ||\overline{P_{x,y}P_{m,n}}||_0 \frac{\overline{P_{x,y}P_{m,n}}}{||\overline{P_{x,y}P_{m,n}}||} \right\} \tag{3.3}$$

Where:

$$\begin{aligned} K &- Stiffness\ of\ the\ Springs \\ R &- Set\ of\ Neighboring\ Points \\ ||\overline{P_{x,y}P_{m,n}}||_0 &- Original\ Length\ Between\ P_{x,y}\ and\ P_{m,n} \end{aligned}$$

In order to get wrinkles on the cloth we should add the internal collision. Which is the addition of changes in point vector times the stiffness of the spring for all the neighboring point. The external forces can be broken down into three different forces:

1. Gravitational Force

$$\begin{aligned} F_{grav}(P_{x,y}) &= mg \\ Where &: \\ g &- Gravitaional\ acceleration\ Constant \end{aligned}$$

2. Damping Force

$$\begin{aligned} F_{damp}(P_{x,y}) &= -cv_{x,y} \\ Where &: \\ c &- Damping\ Constant \\ v_{x,y} &- Velocity\ of\ P_{x,y} \end{aligned}$$

3. Viscosity Force

$$\begin{aligned} F_{visc}(P_{x,y}) = k[N_{x,y}.(u_{fluid} - v_{x,y})]n_{x,y} \\ Where &: \\ k &- Viscosity\ Constant \\ n_{x,y} &- Unit\ Normal \\ u_{fluid} &- Velocity\ of\ Fluid \end{aligned}$$

Here we apply Euler's method that was discussed in chapter 2 in order to integrate the equation 3.3 through time so that we will be able to find out the position of $P_{x,y}$ at each time step. We apply provot's correction [Provot, 1997] to the classical model where un-realistic deformations occur. This is done by managing the constraint points, if the rate goes too high, the elongations of the spring is set to a limit of around 10 percent. The advantage of using this model is that the cloth can flow through any liquids.

3.2.3 Particle Model

Many people have created different particle models to obtain an ideal cloth model. One of the famous model among the many was from Davin Breen [Breen, 1993], [Breen et al., 1994], [Breen, 2000], [Breen et al., 1992]. Particle modeling is one of the best way to model the cloth. Here we will be discussing about the various research done by David Breen with other scientists and the corrections that will be made so that at the end we achieve a good cloth model.

We will be modeling cloth such that simulation depends on fabric. This is because, say we are modeling silk, we want the results to be different compared to that of other cloths. Continuum mechanics concept will be applied here to simulate a cloth model by taking advantage of finite elements(Finite Difference Methods)[Eischen et al., 1996], [Thomaszewski et al., 2006]. Only drawback of using this technique is that it does not take into account the property of integral particles. It is because it looks at the smallest possible level and it presumes that at this level only smallest interactions are present. In reality these smallest interactions makes an enormous impact to movements of the cloth and its structure. We will see how exactly these interactions can be captured by applying interacting-particle techniques.

Capture Micro-structure

Cloth is modeled as a cluster of particles which are placed at each intersecting points of the warp and weft thread. Warp being the vertical threads, horizontal threads are the weft. There will be a force i.e compression force which will act among the weaves that provides a clamp which in turn builds an axis so that the threads bend.

First step we should do is to get all the properties of the micro-structure of the cloth, this includes contact, bending, trellising. Overall energy of the system can be given as:

$$U_{toatl} = U_{repel} + U_{stretch} + U_{bend} + U_{trellis} + U_{gravity} \tag{3.4}$$

The origin of each of the forces can be found in the table below:

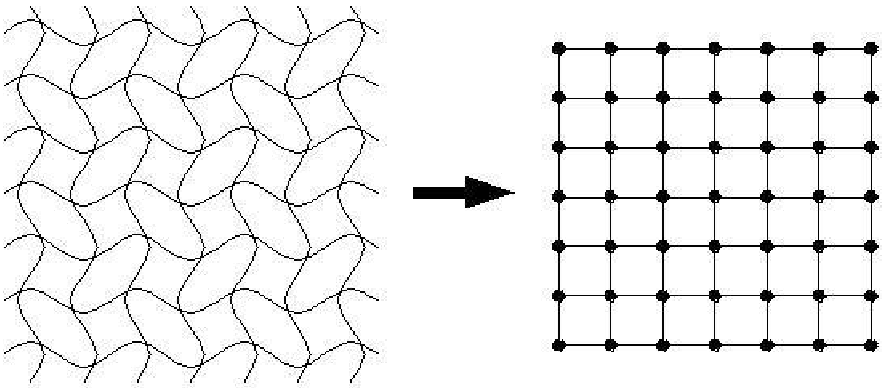


Figure 3.2: Blueprint of Initial Cloth and Portrayal of Same in Grid.

Table 3.1: Origin of Forces

U_{repel}	$U_{stretch}$	U_{bend}	$U_{trellis}$	$U_{gravity}$
Artificial energy of repulsion to keep minimum distance between particles	Energy that represents the tensile strain	Energy caused by threads bending out of the cloth plain	Energy caused by bending around a thread crossing in the plane. Represents shear.	Gravitational potential energy caused by the particle mass.

Repelling and Stretching

The next step is to find out each part of the force with different equations and we will start with repelling and stretching. The general method used is Kawabata system [Breen et al., 1994] in which the tensile strain data is used for all fabrics. But in simulation, it can be observed that the tensile strain is minimum when it is subjected to stress of its own load. This leads us to build a non-cloth definitive methods for repelling and stretching. The interaction between the two, constructs a sharp energy which will place the particles in a titular length 'w' with the neighboring particles. So when we draw the curves we can derive the following equations which can be used for the repelling and stretching energy:

$$R(r_{x,y}) = \begin{cases} C\left[\dfrac{(w - r_{x,y})^5}{r_{x,y}}\right] & r_{x,y} \leq w \\ & r_{x,y} > w \end{cases} \tag{3.5}$$

$$S(r_{x,y}) = \begin{cases} C\left(\dfrac{r_{x,y} - w}{w}\right)^5 & r_{x,y} \leq w \\ & r_{x,y} > w \end{cases} \tag{3.6}$$

Where:

$$\begin{aligned} R &- \textit{Repelling} \\ S &- \textit{Stretching} \\ C &- \textit{Scale Parameter} \\ x, y &- \textit{Successive Points} \end{aligned}$$

By calculating the energy between every two particles, we need to calculate the entire energy acting at a point say 'x', which can be obtained by summing up all the points on the cloth. We do not have to do that since the value of the function reduces to zero as the distance from that of 'i' increases.

$$U_{repel_x} = \sum_{y \neq x}^{n} R(r_{x,y}) \tag{3.7}$$

The neighboring points also produce stretching energy. The potential at each point 'x' is calculated by summation of individual stretching potential.

$$U_{stretch_x} = \sum_{y \epsilon N_x} S(r_{x,y}) \tag{3.8}$$

Where:

$$N_x \ - \ \textit{set of neighbors to } x$$

Gravity

Gravitational force acts on the cloth, for the behavior of the cloth to look more realistic calculating it is necessary. The gravitational force is calculated by multiplying point mass with that of elevation and acceleration due to gravity.

$$U_{gravity_x} = m_x g h_x \tag{3.9}$$

Where :

$$\begin{aligned} m_x &- Mass \\ h_x &- Elevation/Height \\ g &- acceleration\ due\ to\ Gravity \end{aligned}$$

Bending and Trellising

If we want to consider the properties of cloth when it is draped then bending and trellising energy plays crucial role. In the image shown below [Figure 3.3] the bending of the thread can be seen. The image is shown in the weft direction. The wrap will have a similar image only that there will be a rotation of 90 degrees. The angle between the directions of weft and warp decides value of the energy function. Suppose if the direction of the angles and the function is out of phase then we get the curve as shown in the figure[Figure 3.3 a].

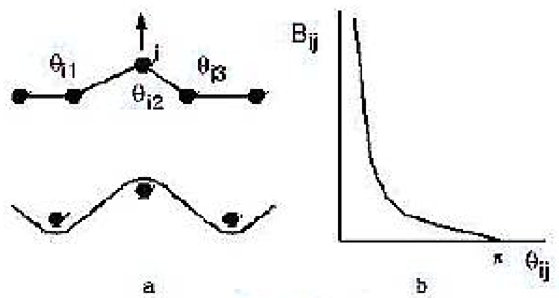


Figure 3.3: Bending

$$U_{bending_x} = \sum_{y \epsilon M_x} B(\Theta_{x,y}) \tag{3.10}$$

Where :

$$M_x \ - \ Collection\ of\ sin\ angles\ Obtained\ by\ Sections\ Connecting\ x\ to\ Neighbors$$

If we join the nearest neighbors horizontally and vertically in trellising we get two segments. There will be formation of S-shape if the cloth is draped this is because the neighboring particles will have an equilibrium angle of 90 degrees. The angle between the initial equilibrium line and the line segment between the new S shape points is the trellising angle. The overall trellising energy can be computed by summing up the entire energy of all the 4 points.

$$U_{trellis_x} = \sum_{y \epsilon Q_x} T(\Phi_{x,y}) \tag{3.11}$$

Where:

$$Q_x \;\; - \;\; Set\ of\ Four\ Trellising\ angles$$

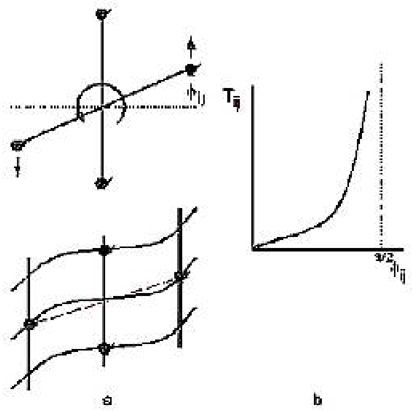


Figure 3.4: Trellising

Minimizing the Energy

After calculating the extent of energy each particle of the cloth has, we need to find out when these particle will have the lowest energy. There is an assumption that is to be made here which is, cloth comes to rest when the energy of the particle are at lowest energy state. Let us first write the energy equation in co-ordinates $\left(U = f(a, b, c)\right)$. U_{repel} and $U_{stretch}$ are the functions of $r_{x,y}$; where $r_{x,y}$ is the distance between the points x and y, it can be calculated as:

$$r_{x,y} = \sqrt{(a_x - a_y)^2 + (b_x - b_y)^2 + (c_x - c_y)^2} \tag{3.12}$$

Further, if we add the angle to the co-ordinates it can be used in the bending equations. This is done by cross product of vectors as shown below:

$$\Theta_{x,y} = \cos^{-1}(r_{x,y-1}.r_{x-1,y-2}) \tag{3.13}$$

Equations for different angles can be obtained by using the same method. This applies for both bending and trellising. Now that the energy that we have obtained is expressed in the form of co-ordinates, differentiation comes into picture, where we can meet the following conditions:

$\frac{\partial U}{\partial x} = 0$, $\frac{\partial U}{\partial y} = 0$, $\frac{\partial U}{\partial z} = 0$

Energy equations are to be differentiated because we get simultaneous equations which will be easy to solve. Here we will be differentiating with respect to x, y, z. The location of all the particles can be obtained by these simultaneous equations.

Cloth Specific Tuning

All the work that have been done till now are for general cloth model, we have not added any attributes to the cloth to make it different and to behave differently. By following the Kawabata Evaluation System (KES)[Breen et al., 1994] certain equations are developed. This system is followed because it measures both mechanical and physical properties of the cloth. KES explanation is not included in this paper. Structural Mechanics knowledge is necessary for better understanding of KES system which is beyond this paper.

The goal here is to develop functions that produces a realistic behavior for our cloth model. This can be done by two functions B (Bending Energy Function) and T (Trellising Energy Function). The amount of force required on a sample fabric so that there will be 3 types of deformations is measured by KES. In a graphical representation we get like the one shown in [Figure 3.5]:

The graph is represented using polynomials so that we can extract the required values from it. To do this we use B and T functions.

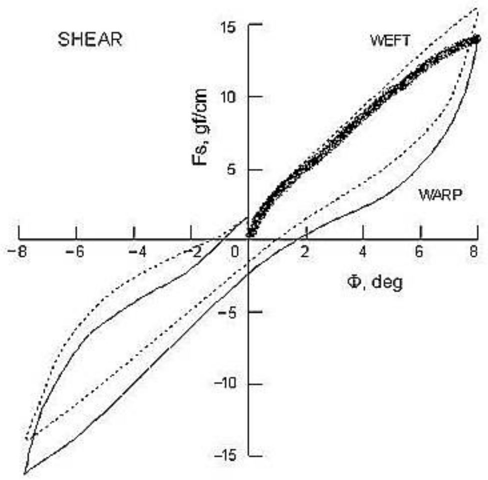


Figure 3.5: Kawabata Plot for 100% Cotton

Bending Energy Equations

We will apply theory of elastic bending [Wang et al., 2011], by applying we get straining energy dU which is generated due to the bend stored in the segments as dS. Both of which can be linked as:

$$dU = \frac{MdS}{2R} \tag{3.14}$$

Where:

$$M \; - \; Bending\ Movement\ on\ Segment$$
$$R \; - \; Radius\ of\ Arc\ Associated\ to\ Curvature\ by\ K = 1/R$$

Distance between the neighboring particles is w. Each one of these particle represents a part of the cloth. If we say these parts of the cloth are beams which run in parallel to one another, then the energy of a beam can be calculated as:

$$U = \int_{0}^{w} \frac{MK}{2} dS \tag{3.15}$$

Considering curvature and bending movement to be constant over a limit $w \times w$, the equation of energy then becomes:

$$U = \frac{MK}{2}w \tag{3.16}$$

Where:

$$M \quad - \quad Moment\ per\ unit\ Length$$

The energy for threads in cloth is given as:

$$B = \frac{MK}{2}w^2 \tag{3.17}$$

Since M is in terms of K, calculation of K(curvature) is necessary. Between two neighbors the curvature with respect to a point is considered constant. By doing this a circle can fit between the two neighboring points. The equation of K can be given as:

$$K(\Theta) = \frac{2}{w}\cos\left(\frac{\Theta}{2}\right) \tag{3.18}$$

If the angle tends to move towards 0, then the values of K becomes $\frac{2}{w}$. Physically this is wrong. When the angle tends to move towards 0 the tread will bend back on itself thereby creating curvature of infinity. In order to correct this problem, an assumption is made where the equations will have angular values between 45 and 180 degrees. If the simulation encounters other angles then the following equations can be used:

$$K(\Theta) = \begin{cases} -\left(\frac{\pi}{4}\right)^2 \frac{b}{\Theta} + a + \frac{\pi}{4}b & 0 \leq \Theta \leq \frac{\pi}{4} \\ \frac{2}{w}\cos\left(\frac{\Theta}{2}\right) & \frac{\pi}{4} \leq \Theta \leq \pi \end{cases} \tag{3.19}$$

$$a = \frac{2}{w}\cos\left(\frac{\pi}{8}\right), b = \frac{1}{w}\sin\left(\frac{\pi}{8}\right)$$

Trellising Energy Equations

With the force F and displacement dS the work dW is given by:

$$dW = FdS \tag{3.20}$$

If the width l is constant during shearing, then the particle travels through a path which is defined as $S = lA$. 'A' being the shear angle. Further differentiating, $dS = ldA$ is obtained. The force will act on a circular arc, in which case the direction will not be parallel to the force. For a situation like this, $F\cos(A)$ can be considered as the component of the force along the displacement. By this equation the total energy for shearing can be derived as:

$$T = \int F\cos(A)ldA \tag{3.21}$$

Analysis says that Kawabata curves can be used for any system if a shearing force being function of 'A' gives 'T' on integration.
The maximum shear is attained when the angle is 60 degrees, at this interval the forces tends to go-to infinity. In which case the following equation can be used:

$$\frac{a}{(60 - A)} + b \tag{3.22}$$

Chapter 4

Implementation

We have seen that simulation of cloth have been developed significantly well with time. We have seen different techniques, its advantages, dis-advantages, we have also made modifications to the techniques to improve the simulation to get better results. For further analysis, I would like to develop simulations with different scene using different approaches, the implementation of which is explained in this chapter. At the end, we will develop a library for simulation of cloth, so that we can analyze the behavior of the cloth for different parameters and at the same time evaluate our approach. Developing a library also helps for future work.

4.1 Interacting Forces

In a given environment, if we simulate a cloth, for its movement there has to be different forces acting on it. For more realistic results we need to consider the forces acting on the cloth which mainly are:

1. Mass-Spring
2. Gravitational Forces
3. Wind Forces
4. Collisions
5. Frictional Forces

All the computations are done by considering the Newton's 2^{nd} law of motion which states "The overall force 'F' acting on a body is equal to the product of the mass 'm' and acceleration 'a' of the object".

If a force is measured between two bodies, it should be considered as the basic force so that one understands how a particle accelerates and what positions are to be changed in the cloth model in each time step.

4.1.1 Mass-Spring Strength

It is necessary to maintain the connection between particles in the cloth, which is commonly called 'damping springs'. These springs cannot be seen, but helps hooking up each point of the cloth to make it look more real. It is explained that there will be need of three spring in [Provot, 1995].

By following Hooke's law, stiffness of a spring '$F_{stiffness}$' requires more than mass of the body and acceleration; the difference between the original position 'l_0' to that of the magnitude of distance between the positions of both particles '$|l|$' should be considered. The result is then multiplied with the direction 'd', '-1' and spring constant 'k_s'.

$$\begin{aligned} l &= P_0 - P_1 \\ d &= \frac{l}{|l|} \\ F_{stiffness} &= -1 * k_s * d * (|l| - l_0) \end{aligned} \tag{4.1}$$

Where:

$$\begin{aligned} P_0, P_1 &- Positions\ of\ the\ Particles \\ l &- Position's\ Difference \\ d &- Particles\ Direction \end{aligned}$$

$$\begin{aligned} l_0 &- Initial\ Length\ Between\ the\ Particles \\ k_s &- Spring\ Constant \\ F_{stiffness} &- Overall\ Force\ of\ Stiffness \end{aligned}$$

In order to avoid the oscillation, the two velocities i.e initial and current will have different values whose difference is computed '$v_{difference}$' and is multiplied with coefficient of damping 'k_d' and -1; the result is the damping force '$F_{damping}$'. If we add the damping force and spring's stiffness that were calculated, we get the resulting force of the spring 'F_{spring}'

$$\begin{aligned} v_{difference} &= v_0 - v_1 \\ F_{damping} &= -1 * k_d * v_{difference} \end{aligned} \tag{4.2}$$

$$\begin{aligned} v_0\ and\ v_1 &- Velocites\ of\ Two\ Particles \\ v_{difference} &- Difference\ in\ Velocity \\ k_d &- Damping\ Co-Efficient \\ F_{damping} &- Resulting\ Force\ of\ Damping \end{aligned}$$

$$F_{spring} = F_{stiffness} + F_{damping} \tag{4.3}$$

Structural Springs Connection

Structural spring, helps the particle to stay connected in the mesh by providing it with a force allowing it to compress and extend partnering with other springs, at the same time maintaining the integrity of the particles when it is extended or compressed.

In order to connect a point with its neighbor, the one near to it by a small space in all the four directions(top, down, left, right) are considered, doing this all the points will

be connected by structural springs. Together the whole cloth structure forms a grid, where the cloth can be stored and simulated.
By following Figure 4.1 if (i, j) acts as current position of a particle, then the neighboring co-ordinates will be:

$$([i-1], j);\ ([i+1], j);\ (i, [j-1]);\ (i, [j+1])$$

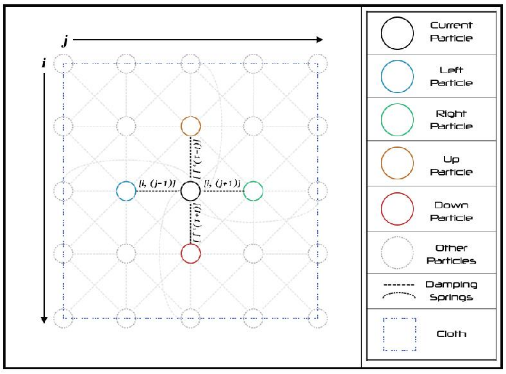


Figure 4.1: Particles Connected by Structural Springs

Bending Springs Connection

The particles that are at a distance of two spaces from its neighbors are connected by bending springs which helps to maintain stability. The connecting force manages the bending tension of the cloth. In brief it avoids the particles in the system, not to be too close or far from each other, which can seen in the form of wrinkles or strains.

In this case, the points which are linked together are away to each other by two spaces in all the four directions (top, down, left, right). These points are connected by bending springs.
By following Figure 4.2 if (i, j) is the current position of a particle, then the neighboring co-ordinates will be:

$$([i-2], j);\ ([i+2], j);\ (i, [j-2]);\ (i, [j+2])$$

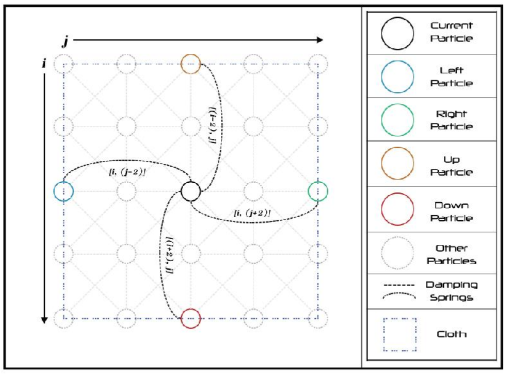


Figure 4.2: Particles Connected by Bending Springs

Shearing Springs Connection

For the cloth to remain stable it has to be connected between the diagonal points in the grid which is done by shearing stress. This is a junction from which cloth will have shearing springs. The deformation of the cloth takes place when two diagonal particles are separated or pushed close to each other. In this implementation we will not be discussing much about deformation rather we will be observing the behavior of the cloth as we change the shearing stress value.

We connect a particle to its neighbors who are at unit diagonal space in all the four directions (top, down, left, right) by shearing springs.
By following Figure 4.3 if (i, j) acts as current position of a particle, then the neighboring co-ordinates will be:

$$([i+1],[j+1]);\ ([i+1],[j-1]);\ ([i-1],[j+1]);\ ([i-1],[j-1])$$

4.1.2 Gravity

According to Issac Newton's law of universal gravitation "every particle on earth attracts other particles with a force which is proportional to the product of their masses and inversely proportional to the square of the distance between them".

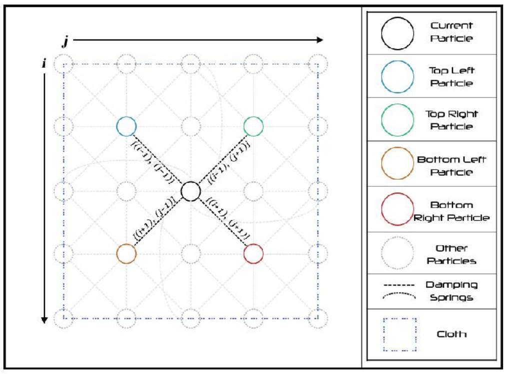


Figure 4.3: Particles Connected by Shearing Springs

$$F_{gravity} = \frac{G * m_1 * m_2}{d^2}$$
$$G = 6.67 * 10^{-11} N.m^2/kg^2 \quad (4.4)$$

Where:

$$m_1 \; and \; m_2 \; - \; Masses \; of \; the \; First \; and \; Second \; Bodies$$
$$d \; - \; Distance \; Between \; their \; Centers \; of \; Mass$$
$$G \; - \; Gravitational \; Constant$$
$$F_{gravity} \; - \; Resulting \; Gravitational \; Force$$

This equation is applied for universal purposes and when calculating they use objects that has big masses like the mass of planetary motion. By following Dunn and Parberry [Dunn and Parberry, 2011] for the calculation the Earth's surface which is a constant and is considered as the mass of the first object.

By considering this fact it can be shown that the objects interacting with the Earth's surface will be within a certain height whose value is not large, so the distance between the the center of masses between the first body and the object will not change drastically. Hence the distance is taken as constant.

To calculate gravity most of the unknowns are known and the acceleration due to gravity is a constant which we all know and the value is 9.80665 m/s^2.

4.1.3 Wind

In order to calculate the actual effect of the wind on the objects it needs more variable and the calculation is expensive. In the cloth model the wind is computed by considering cloth as a cluster of triangle with 3 particles. By following Mosegaar [Mosegaard, 2013] the wind forces are calculated for each triangle at a given time frame. In the natural environment the wind hits the object in a certain direction, the object reacts based on the magnitude that it is been hit by and is dependent on the triangle's normal direction and angle formed by it. If we represent the same in terms of equation, we get:

$$\begin{aligned} d &= \frac{n}{|n|} \\ F_{wind} &= n \times (d \cdot v_{wind}) \end{aligned} \tag{4.5}$$

Where:

$$\begin{aligned} n &- Normal\ of\ the\ Triangle\ Face \\ |n| &- Magnitude\ of\ the\ Normal \\ d &- Normalized\ Normal\ Direction \\ v_{wind} &- Wind\ Velocity \\ F_{wind} &- Resulting\ Wind\ Force \end{aligned}$$

4.1.4 Collision

In this implementation, two collisions are given more weight. The first one is how the cloth looks on the scene which we can call floor, here it is represented by a plane where the forces acting on the objects in the system are subjected to be on top without any interpretation. The next one is external object which collides with the cloth, in order to detect collisions from external objects, the position and direction of normal are considered for every point in the mesh so that we get more realistic results.

Planar Collision

To identify the collision of a point with an object, some important things are considered. Firstly there is a central point through which plane passes and its positions is taken as 'P'. Second thing is normal 'n' is perpendicular to 'P'. Now with all these set of points and following [Lengyel, 2002], an object 'Q' can be defined as '$n \cdot (Q - P) = 0$'.

There is a need of distance 'd' which is from the plane to any of the particle position, considering the plane being the center of the scene, dimensions being '$[0, 0, 0]$', we get offset 'D', this can be applied separately. Normalization is required for the plane's normal in order to get more accurate values. It is seen that, there exists three situations for different values of 'd':

1. '$d > 0$' the point 'P_P' can be considered to be on the positive section of the area. Here it will be occupying the area in which normal is pointing.
2. '$d < 0$' in this situation the point will be in the negative side of the plane.
3. '$d = 0$' in this situation the point will be filling the same area as that of the plane, so it can be said that it is in contact because it occupies the same area as that of plane.

$$
\begin{aligned}
P_0 &= [0, 0, 0] \\
d &= (n \cdot P_P + D) \\
(d < 0)? &: interpenetration \\
(d == 0)? &: contact \\
(d > 0)? &: seperated
\end{aligned}
\tag{4.6}
$$

Where:

$$\begin{aligned} P_0 &- Obstacle\ Position \\ n &- Plane's\ Normal \\ P_P &- Cloth's\ Particle\ Position \\ D &- Plane's\ Offset\ Position \\ d &- Magnitude\ of\ Resulting\ Singned\ Distance \end{aligned}$$

Thus, the point that fulfill condition three or zero condition, is not allowed to proceed in normal direction 'n', and each particle penetrating internally i.e '$d < 0$' should be lifted in the direction of normal 'n'. The distance moved should be equal to the value obtained by signed length 'd'.

Spherical Collision

To detect the collision of the cloth with Spherical objects, each particle of the cloth is considered and it is compared with the exact distance between the particle to that of sphere taking into consideration the center of the colliding object, in this case sphere has a distance of radius 'r_0' from the point it is calculated [Lengyel, 2002].

The contact point is determined by taking the difference '$diff$' between the center of the object in each interval. The difference is between the position of stationary object 'P_0' to the particle being referenced 'P_P' in the cloth. After this the emerging distance 'd' is checked whether to be identical or less than the radius of the sphere.

$$\begin{aligned} diff &= P_P - P_0 \\ d &= |diff| \\ (d < r_0)? &: interpenetration \\ (d == r_0)? &: contact \\ (d > r_0)? &: separated \end{aligned} \tag{4.7}$$

Where:

$$
\begin{aligned}
P_0 &- Obstacle\ Position \\
P_P &- Cloth's\ Particle\ Position \\
diff &- Difference\ in\ Position \\
d &- Magnitude\ of\ Resulting\ Distance \\
r_0 &- Radius\ of\ the\ Obstacle
\end{aligned}
$$

Now the point that matches the given comparisons will be diverted away from the sphere in the direction of normal 'n', by the value generated during inter-penetration which can be calculated by taking the difference between the resulting length and radius of sphere.

4.1.5 Friction

If the cloth is in resting position and an external body or external forces causes the cloth to move then there is a frictional force acting on the cloth which acts at an angle of 90 degrees to the surface of the cloth. It will drift across and there is no force to stop this. Opposite force is added to the object so that it obstructs the object force making the velocity to be decreased and neutralize the external force.

In cases where the object which blocks the motion of the surface is absent, then external force needs to be applied to decrease the acceleration else the movement of the body does not stop. Kinetic friction is the force that helps in decelerating the motion, this is one of the two frictional forces. It can be defined as a frictional force applied parallel to a body in motion which counteracts the movement of the body [Dunn and Parberry, 2011].

Static friction is the other frictional force. It is a frictional force applied on the surface of the body resting, which counteracts any external forces resulting in the force responsible for the movement of the body being the difference of frictional force and external force [Dunn and Parberry, 2011].

Static Friction

The amount of force that a static friction 'f_s' causes in the scene is calculated by using two values. The first is the co-efficient of static friction 'μ_s' and the normal force magnitude 'n'.

$$f_s = \mu_s \times n \tag{4.8}$$

Kinetic Friction

The kinetic friction comes into play when the body is in movement. So, in order to calculate kinetic friction, normal force 'n' is multiplied with co-efficient of kinetic friction 'μ_k'.

$$f_k = \mu_k \times n \tag{4.9}$$

Chapter 5

Results

After implementing the simulation of cloth, it is necessary to evaluate the results. In this chapter results are evaluated by conducting experiments on different scenes. The evaluation includes the analysis of our new methods and behavior of cloth with change in parametric values.

5.1 Experiment 1

The first simulation was implemented using Baraff and Witkin [Baraff and Witkin, 1998]. However, this approach was modified using the results obtained by other researchers, as well as improvements were incorporated to the simulation based on my insights. Ascher and Boxerman's [Ascher and Boxerman, 2003] correction was added in the implementation of the cloth model which included a gradient algorithm that gives the pre-conditioned gradient along with their pre-conditioner.

In Baraff and Witkin's model the parameters cannot be varied. If all the parameters are kept constant and the mesh is sub divided, it can be observed that the shape of the cloth changes significantly.

According to Baraff and Witkin, condition functions (C) is termed as forces, and they relate this to Energy (E) as:

$$E = \frac{k}{2} C^T C \tag{5.1}$$

Stretching, Shearing and Bending had separate energies. Bend condition function was defined by them by giving a correct argument which was not dependent on the area of the triangle. As a result, square of the area of the triangle controls the stretch and shear energies. Suppose if we split triangle into four equal triangles, then the area of one split will be equal to one fourth of initial energy. Tessellation values does not alter the values of bending and gravitation, from this it can be observed that the higher tessellation values tend to weaken stretch and shear forces, as a result causing the change in shape of the cloth making it look redefined.

Baraff and Witkin's conditions were modified to get more realistic results by making the condition functions dependent on the area of the triangle which in turn makes the energy function to be dependent on the area of the triangle. Baraff and Witkin's stretch conditioned needs to be altered to utilize the square root of area of the triangle.

$$F(X) = \sqrt{a} \begin{pmatrix} ||P_m(X) - b_m|| \\ ||P_n(X) - b_n|| \end{pmatrix} \tag{5.2}$$

and the case that they have given for shear in section 4.3 of their paper is changed to:

$$F(X) = \sqrt{a} P_m(X)^T P_n(X) \tag{5.3}$$

The changes will make the parameters of the cloth simulation not to depend on mesh tessellation. Finally, in Baraff and Witkin's paper equation (11) which is damping forces are defined as:

$$\dot{F}(X) = \left(\frac{\partial F(X)}{\partial X} \right)^T \dot{x} \tag{5.4}$$

The equation I used for implementation is :

$$\dot{F}(X) = \sum_n \left(\frac{\partial F(X_n)}{\partial X_n} \right)^T \dot{x_n} \tag{5.5}$$

Where:

$$n - Changes\ its\ Limit\ for\ all\ the\ Particles\ Influencing\ F$$

5.1.1 Performance

The simulation was executed in a system with following configuration:

Table 5.1: System Configuration

CPU	Intel Core i7-6820HK CPU 2.7GHz
OS	Windows 10
Main Memory	16GB
GPU	Nvidia GTX-980 4GB

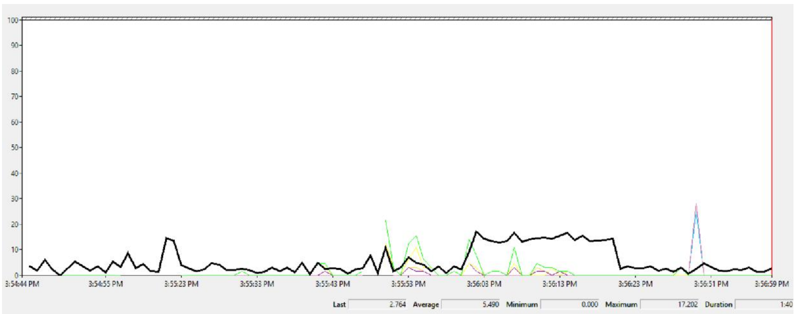


Figure 5.1: Performance of Cloth Simulation

In this experiment we focus mainly on the expensive techniques which includes calculation of collisions and forces. The simulation of cloth had 9 threads which are represented by different colors in Figure 5.1. The highlighted dark like represents the CPU performance in total. From this observation we can conclude that our integrated new technique is efficient and takes less time to produce remarkable results. The complete simulation can be found here [1].

[1] https://www.youtube.com/watch?v=rUVmZZtFCLw

5.2 Experiment 2

In this experiment the relationship between the parameters that have been used for simulation and the cloth behavior have been analyzed. The effect that individual parameters causes on the cloth is shown in the following section.

[Bhat et al., 2003] proposed few plausible parameters which we will be using in our new approach with some changes.

Table 5.2: Parameter Values for Modified Version of Cloth Model

Parameters	Symbolic Representation	Parameter-Values
Stretch Resistance	$K_{stretch}$	5.0
Shear Resistance	K_{shear}	0.5
Bend Resistance	K_{bend}	1.5×10^{-6}
Stretch Damping	$d_{stretch}$	1000.0
Shear Damping	d_{shear}	100.0
Bend Damping	d_{bend}	1.9×10^{-6}
Air Resistance	$K_{air\_resistance}$	0.09
Resistance(MPCG)		0.009
Stretch Range		4.9×10^{-5}

A series of simulations were performed in order to see the influence and evaluation of the parameters. In the simulation different settings are available, here we will see the draping of the cloth in circular table. The cloth in the simulation is made up of 66×66 square grid structure which has a split for each square to form triangles. For detailed analysis we make changes to only one parameter in each experiment and then observe the behavior of the cloth which includes appearance and spatial distribution of the energy functions of stretch, shear and bend energy. RGB colors are followed to represent stretch, shear and bend energies respectively.

Figure 5.2: Representation of Stretch, Shear, Bend and Combined Energy in Simulation of Cloth (from left to right).

5.2.1 Number of Blocks

First we will see how the number of blocks affect the performance of the cloth. To do this we will first discrete the surface in different combinations as shown in the Figure 5.3. It can be observed that there is not much change in the energy levels for higher and lower number of blocks. For further experiments we keep the block number to be 66×66.

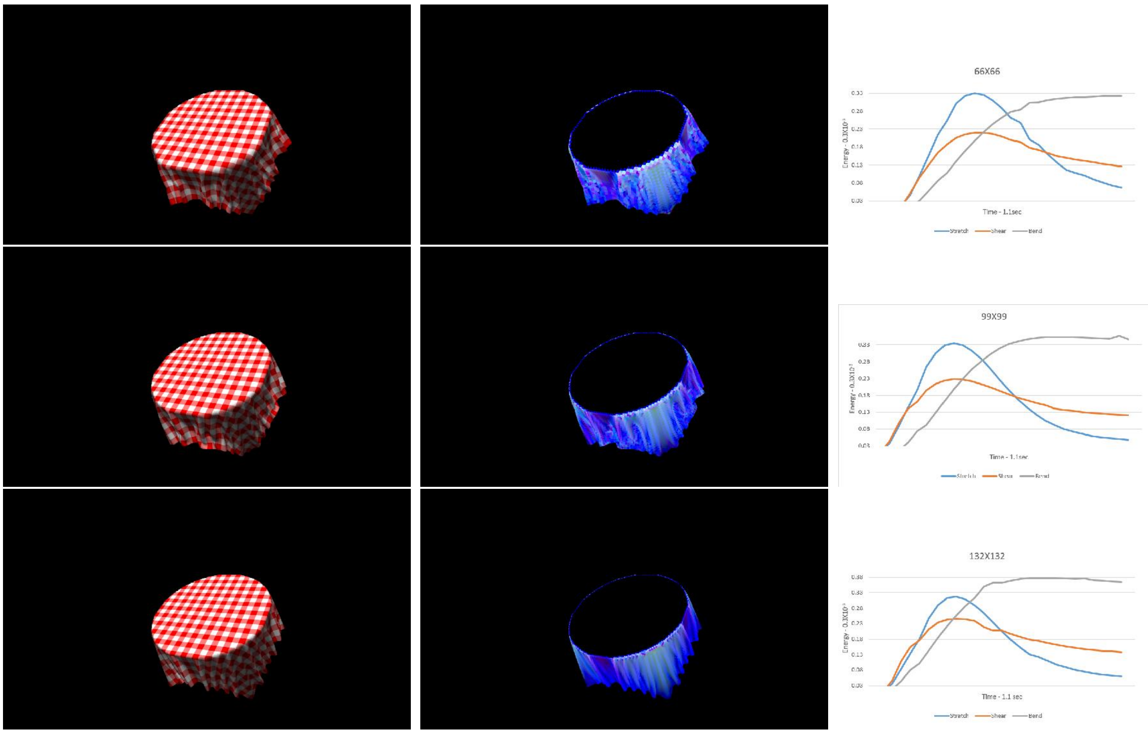


Figure 5.3: Simulation of Cloth with Different Block Values

For the lower discrete values the cloth failed to bend to the applicable extent, but when the blocks were increased, more realistic behavior was seen as the bending started rising slowly which can be observed in the graph above [Figure 5.3]. So the amount of discrete level is dependent on the application for which the cloth is used. For lesser details low values and for finer details higher value of block structuring can be done.

5.2.2 Time-steps

Baraff and Witkin [Baraff et al., 2003] used bigger time-steps which forced the motion of the cloth to have damping behavior. This is demonstrated by keeping the time-step constant in Figure 5.4. If the graph is observed, we can notice that there is sudden swing in the energy functions.

It is also observed that if the conditions for one simulation are to be used for other cloth simulation then it is recommended to have the same timescale. Only then the parameters can be used again. This behavior needs further experiments and studies.

For further experiments we keep the time-step constant because varying it and analyzing results would make it hard to correlate the outcome of different experiments as the system time values grow and drop down unnoticed.

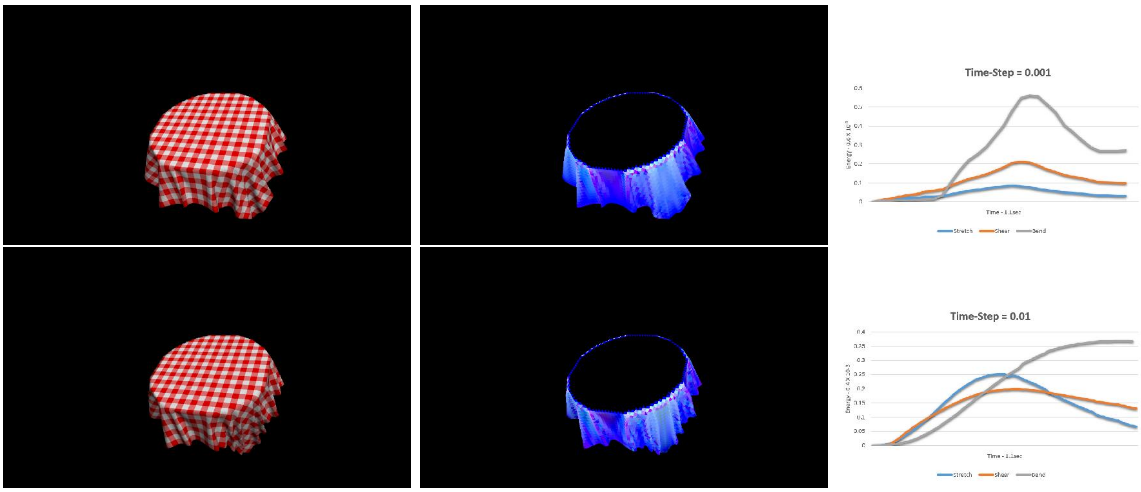


Figure 5.4: Simulation of Cloth with Different Time-Step Values

5.2.3 Resistances

Stretch Resistance

The different values to the stretch did not alter the actions of the cloth, which also did not affect the final draping of the cloth. But the overall energy from the stretch did change consistently and then remained constant. When the stretch values were small there was a state of equilibrium, and for stretch with bigger values there was short-term administration. The behavior can be seen in the Figure 5.5.

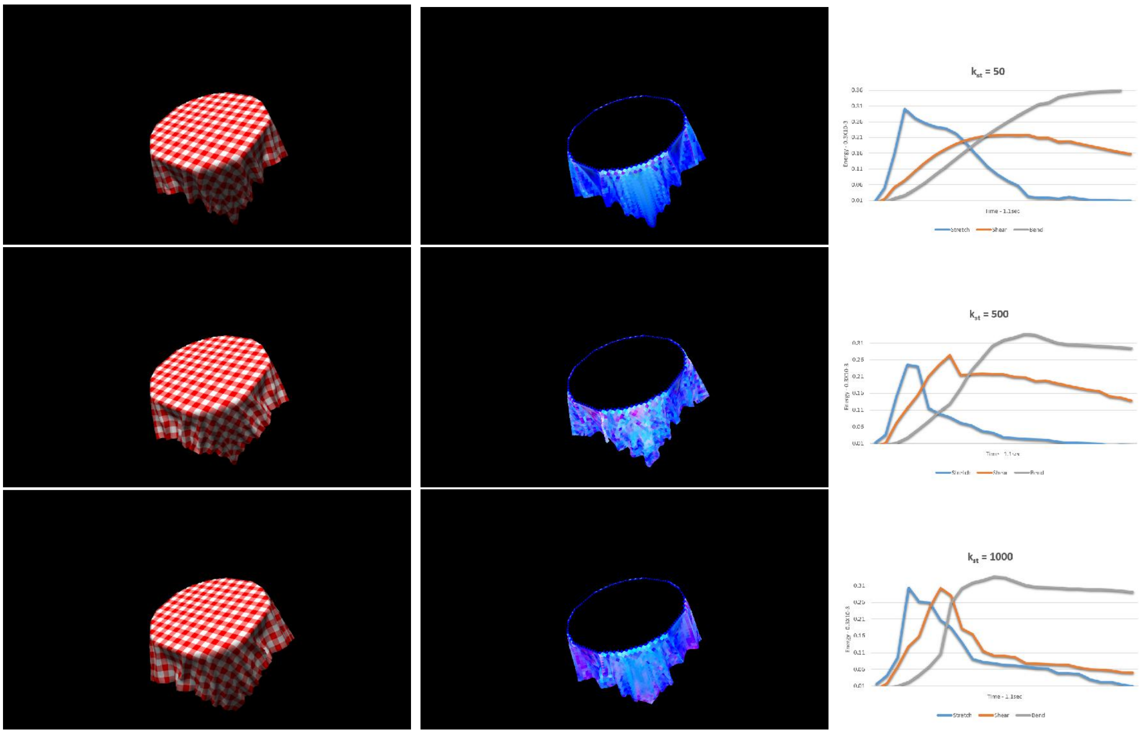


Figure 5.5: Simulation of Cloth with Different Stretch Resistance Values

Shear Resistance

Giving different values to shear resistance did change the appearance of the cloth. For values being bigger it was observed that there were more damping, while values being small the cloth appeared to be sagging, also the movement was free because more energy was gathered in the cloth. This behavior can be observed in the Figure 5.6.

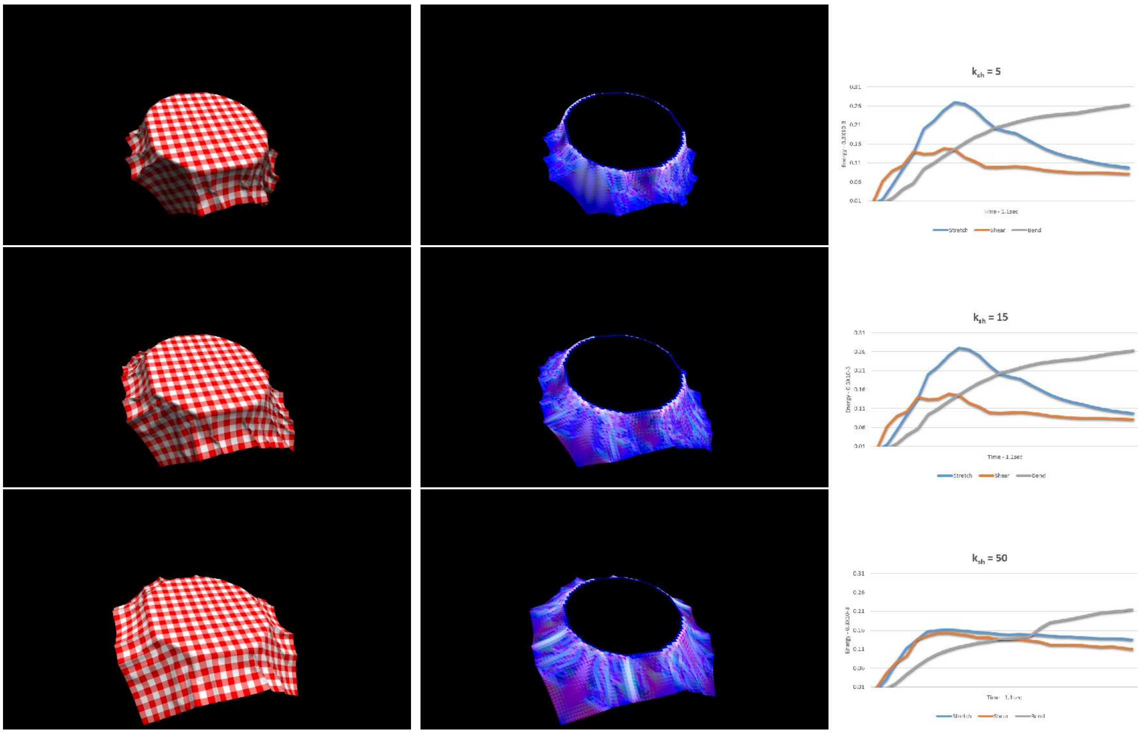
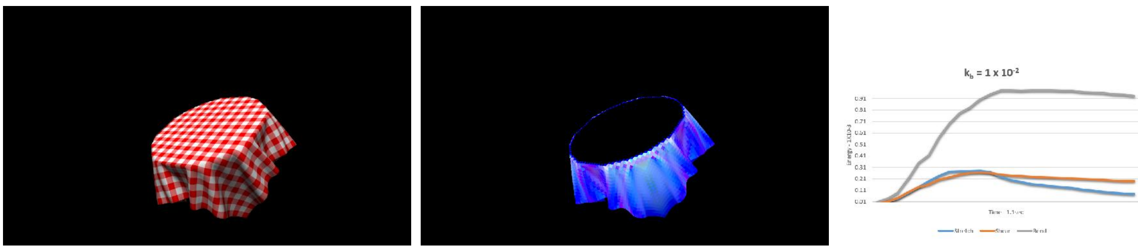


Figure 5.6: Simulation of Cloth with Different Shear Resistance Values

Bend Resistance

Among the three resistances, modifying the value of bend did cause major changes in the simulation of cloth. For higher values of bend more wrinkles were observed to appear on the cloth, at the same time more energy was produced in the system. This behavior can be observed in the Figure 5.7. It must be noted that different application needs different requirements, based on the circumstances and needs we can give suitable bend resistance values.



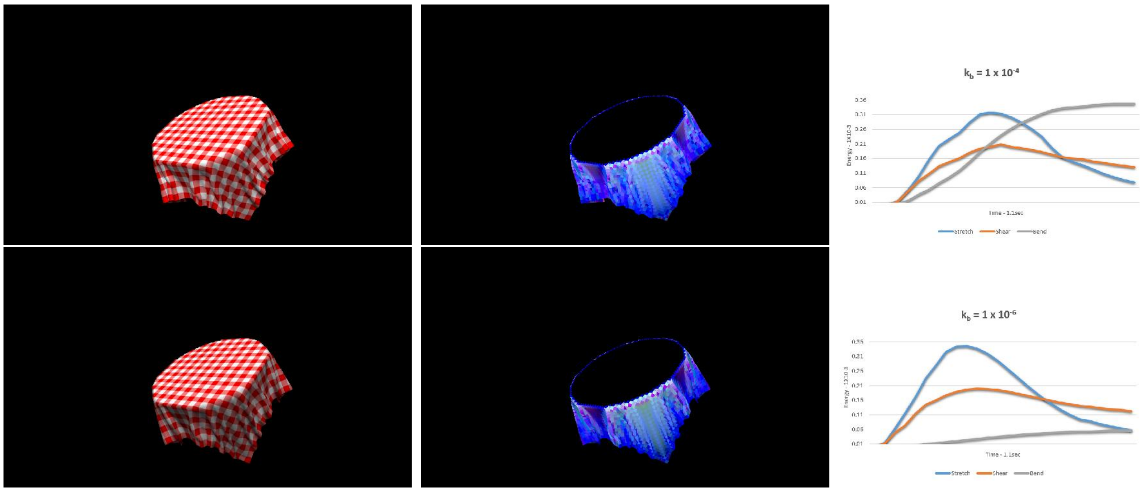


Figure 5.7: Simulation of Cloth with Different Bend Resistance Values

5.2.4 Damping Values

From my analysis on the research papers that I have read, it is said that the damping behavior of cloth can be observed by changing bending values where it is either over-damped or under-damped depending on the material used for simulation of the cloth, where as for stretch and shear it is always over-damped.

For better understanding of over-damped and under-damped, I will define it in terms of our experiment; if the cloth comes to rest in a slow pace then it is over-damped, suppose if the cloth falls quickly and then swings for some duration before it comes to rest then it is under-damped. We will look at this behavior in more detail in the following section and come to conclusion at the end.

Damping with Stretch

With the changes in damping stretch values there were few variations at the beginning of the simulation, but the overall effect on the cloth was minimum. Since we are changing the stretch values there was significant changes in stretch energy, which can be observed in the Figure 5.8, other than that the drape and other behavior did not have any effect. One interesting thing is, for values ≤ 2 i.e d_{st}, computation was not successful.

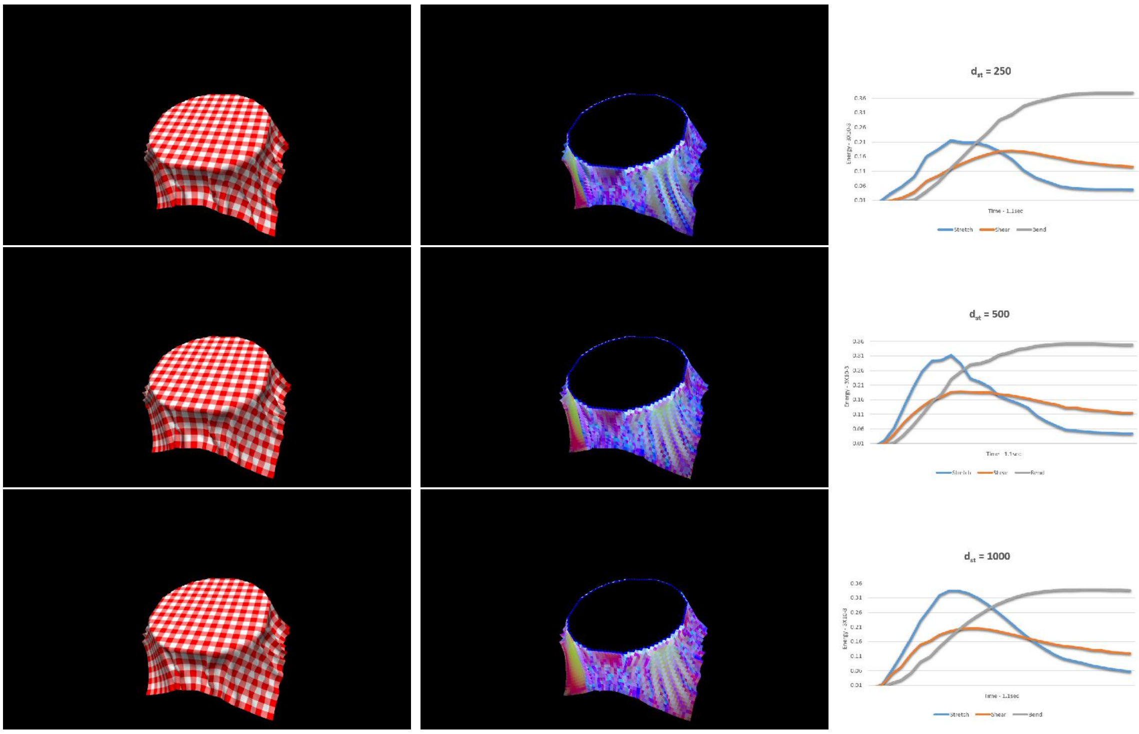
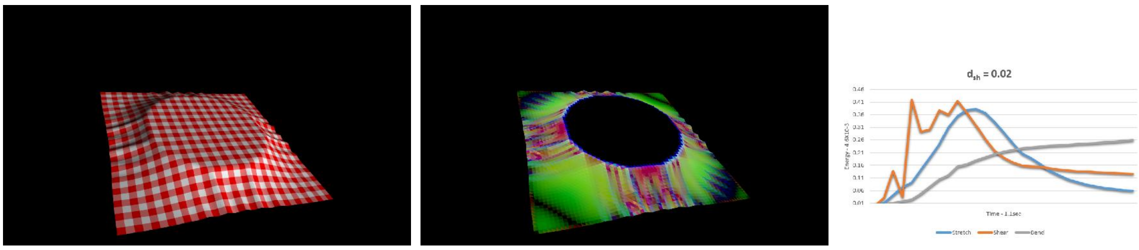


Figure 5.8: Simulation of Cloth with Different Stretch Damping Values

Damping with Shear

Similar to stretch damping, shear damping also did not affect the cloth. In the following graph Figure 5.9 we can observe the over-damped and under-damped behavior of the cloth for different values of shear damping. The final state of the cloth when it is draped had no adverse effect. Even though stretch damping and shear damping did not cause any adverse or major changes in simulation of cloth, it must be noted that these dampers support each other and helps in providing a stable cloth simulation.



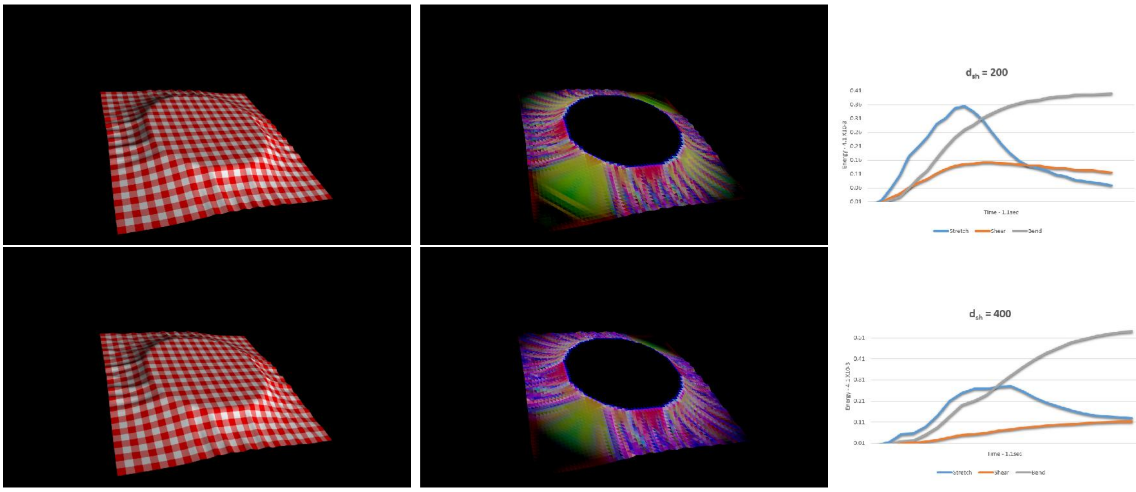
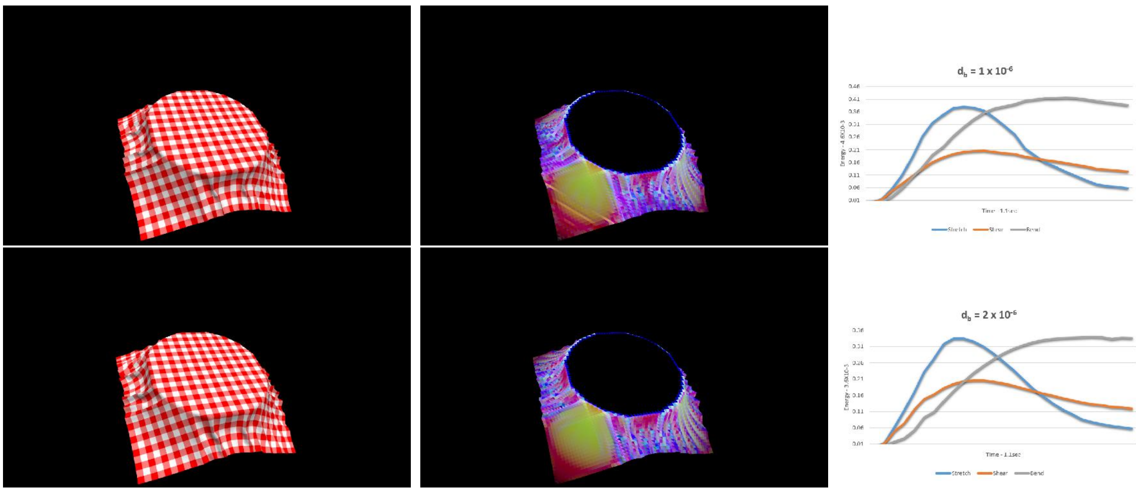


Figure 5.9: Simulation of Cloth with Different Shear Damping Values

Damping with Bend

Changes to the damping bend values resulted in significant changes in the cloth. Major changes can be observed around the corners which is shown in Figure 5.10. At times there was smooth movements in the simulation even though there existed over and under damped. For higher values of damping bend it can be observed that there are less wrinkles where as for smaller values of damping bend we have more wrinkles accumulating in the cloth. Despite these changes the final form of all the simulation was no different.



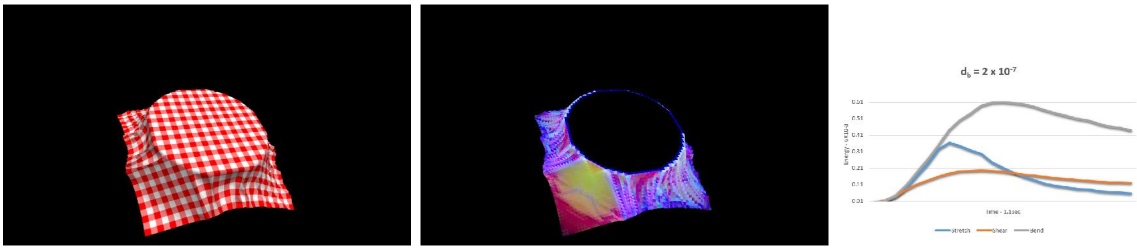


Figure 5.10: Simulation of Cloth with Different Bend Damping Values

5.2.5 Summary

We have seen the affect that is caused by parameters on the cloth, and can say that the behavior of the cloth can be controlled by giving different parameter values which results in simulating more realistic cloth behavior. Another importance of parameters is that, different cloth materials have different properties and the parameters can be given suitable values, which helps in simulating the cloth with a certain fabric properties. For example, say we want to simulate a cloth for a King, we know that king will have a robe and the robe is made up of different material, in such a case we can create a complex mesh structure and assign different values for the parameters in different meshes, which provides the intended result.

Finally, with the new approach where we control the behavior of cloth by changing parameter values, we can say that more interesting, realistic and remarkable results can be achieved for different fabrics without changing the techniques rather changing parametric values. Thus achieving the main goal of our project.

5.3 Experiment 3

Till now we have seen the behavior of the cloth and it should be noted that behavior of the cloth is different and appearance of the cloth is different; but both of these are important to make the simulation of cloth realistic. To measure the appearance of the cloth there is no particular method, which says the cloth's appearance is ideal to look realistic. This is because different characters have different attire and these attire will have different textures added to it, though the fabrics are same and the behavior is same, it may look different because of different textures added to it and also different situations require different cloth's simulation like a flag might need a different appearance when compared to the cloth put on a character. Hence it is not right to say that the cloth's appearance is ideal just by comparing. Unlike proving it in a theoretical way, we look more into practical application and then decide about the appearance.

[Kavan et al., 2011] developed an up-sampling method, which they say is better than low quality cloth rendering and less detailed than high quality cloth rendering, so the cloth obtained by their method is in between the two. But the paper [Kavan et al., 2011] does not have any particular method which they have used for comparisons. Even though our new approach provides high quality rendering images as we change parametric values; until further experiments and proper method for measuring the appearance of the cloth, our new approach can be placed in between the up-sampling and high quality rendering.

Figure 5.11: Appearance of Cloth in Different Scenes

Chapter 6

Conclusions

The main aim which was implementing a technique for simulation of cloth that provides realistic cloth behavior which can be used in games and animation have been achieved by conducting experiments with different scenes. A library was developed that can be used in any versions of OpenGL and also can run either with CPU or with GPU.

Different techniques that was used for the simulation of cloth was implemented and was analyzed to determine the advantages and dis-advantages of individual technique. The models used for simulation of cloth was experimented and modified to obtained improved results. After careful examination the efficient methods were incorporated with the new approach and then integrated; during the process the problems faced and the possible paths to overcome the problem was discussed and implemented, which resulted in a technique that provides realistic cloth simulation to an agreeable level.

Though few of the situations did not provide the expected results with the new approach discussed in Chapter 5, the new approach is promising in providing better results. For most of the situations including different collisions, behavior of the cloth under different circumstances, different clothes having different fabric properties were obtained by simulating cloth with the new approach. To prove that the new approach works very well many experiments were conducted for different scenes which includes, checking the efficiency of the new approach and analyzing cloth behavior for different parametric values. All these experiments gave remarkable output justifying that the new approach can be used to simulate realistic clothes in Games and Animation.

6.0.1 Future Work

Many further improvements can be done to obtain more realistic cloth behavior.

- Use of Spatial division method instead of sub-division which can provide more efficient results.
- The self-collision in the cloth can be improved further with hash technique.
- Implementing a Game and Animation with the new approach.
- Implementing Cloth's behavior when it comes in contact with fluids.

Appendix A

Appendix

A.1 Cubic Equation for Collision Detection

The Cubic equation techniques is used to solve cloth and object collision. For each time frame a trajectory path is imagined, which is linear in this case, then we obtain a cubic equation from the assumption, this cubic equation contains all the intersecting line segments and the swept volume of a triangle; solving it helps to identify the collision.

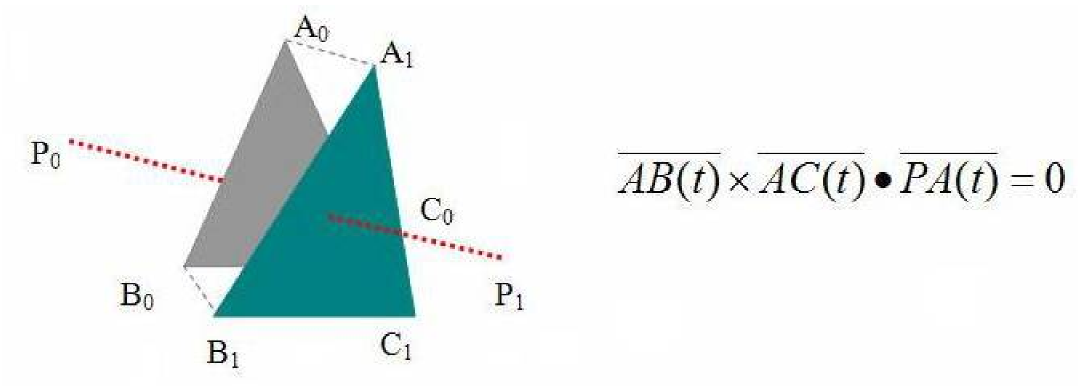


Figure A.1: Cubic Equation

The equation in the figure is the cubic equation, which helps in finding the intersection of the vertex of a triangle with the vertex that is moving. A, B, C being the sides of the triangle. P is the vertex that is moving, and they are represented as a function of time t.

Bibliography

Ubisoft entertainment, 2004. URL https://www.ubisoft.com/en-GB/.

U. Ascher and E. Boxerman. On the modified conjugate gradient method in cloth simulation. *The Visual Computer*, 19(7–8):523–531, December 2003. URL http://www.cs.ubc.ca/~ascher/papers/ab.pdf.

George Baciu and Wingo Sai-Keung Wong. Image-based collision detection for deformable cloth models. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):649–663, November 2004. ISSN 1077-2626. doi: 10.1109/TVCG.2004.44. URL http://dx.doi.org/10.1109/TVCG.2004.44.

D. Baraff and A. Witkin. Rapid dynamic simulation. In D. House and D. Breen, editors, *Cloth Modeling and Animation*, pages 145–173. A.K. Peters, 2000.

David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi: 10.1145/280814.280821. URL http://doi.acm.org/10.1145/280814.280821.

David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. *ACM Trans. Graph.*, 22(3):862–870, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882357. URL http://doi.acm.org/10.1145/882262.882357.

Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. Estimating cloth simulation parameters from video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer*

Animation, SCA '03, pages 37–51, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. ISBN 1-58113-659-5. URL http://dl.acm.org/citation.cfm?id=846276.846282.

D. Breen. *A particle-based model for simulating the draping behavior of woven cloth.* PhD thesis, Rensselaer Polytechnic Institute, 1993.

D. Breen. A survey of cloth modeling methods. In D. House and D. Breen, editors, *Cloth Modeling and Animation*, pages 19–53. A.K. Peters, 2000.

D. Breen, D. House, and P. Getto. A physically-based particle model of woven cloth. *The Visual Computer*, 8(5–6):264–277, June 1992.

David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 365–372, New York, NY, USA, 1994. ACM. ISBN 0-89791-667-0. doi: 10.1145/192161.192259. URL http://doi.acm.org/10.1145/192161.192259.

R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2003)*, pages 28–36. ACM Press, 2003. URL http://www.cs.ubc.ca/~rbridson/docs/cloth2003.pdf.

R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM. doi: 10.1145/1198555.1198573. URL http://doi.acm.org/10.1145/1198555.1198573.

Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566623. URL http://doi.acm.org/10.1145/566654.566623.

M. Desbrun, M. Meyer, and A. Barr. Interactive animation of cloth-like objects in virtual reality. In D. House and D. Breen, editors, *Cloth Modeling and Animation*, pages 219–239. A.K. Peters, 2000.

F. Dunn and I. Parberry. *3D Math Primer for Graphics and Game Development, 2nd Edition*. Taylor & Francis, 2011. ISBN 9781568817231. URL https://books.google.ie/books?id=X3hmuhBoFF0C.

J. Eischen and R. Bigliani. Continuum versus particle representations. In D. House and D. Breen, editors, *Cloth Modeling and Animation*, pages 79–122. A.K. Peters, 2000. URL http://www.mae.ncsu.edu/faculty/eischen/publications/AKPeters-Chap4.pdf.

Jeffrey W. Eischen, Shigan Deng, and Timothy G. Clapp. Finite-element modeling and control of flexible fabric parts. *IEEE Comput. Graph. Appl.*, 16(5):71–80, September 1996. ISSN 0272-1716. doi: 10.1109/38.536277. URL http://dx.doi.org/10.1109/38.536277.

C. Feynman. Modeling the appearance of cloth. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.

Ladislav Kavan, Dan Gerszewski, Adam Bargteil, and Peter-Pike Sloan. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph.*, 30(4): 93:1–93:9, 2011.

Maxim Kazakov. Catmull-clark subdivision for geometry shaders. In *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '07, pages 77–84, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-906-7. doi: 10.1145/1294685.1294699. URL http://doi.acm.org/10.1145/1294685.1294699.

Tosiyasu L. Kunii and Hironobu Gotoda. Singularity theoretical modeling and animation of garment wrinkle formation processes. *Vis. Comput.*, 6(6):326–336, November 1990. ISSN 0178-2789. doi: 10.1007/BF01901019. URL http://dx.doi.org/10.1007/BF01901019.

Eric Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Charles River Media, Inc., Rockland, MA, USA, 2002. ISBN 1-58450-037-9.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987.

ISSN 0097-8930. doi: 10.1145/37402.37422. URL http://doi.acm.org/10.1145/37402.37422.

Stephen McAuley and Stephen Hill. Physically based shading in theory and practice. In *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4289-6. doi: 10.1145/2897826.2927353. URL http://doi.acm.org/10.1145/2897826.2927353.

J. Mezger, S. Kimmerle, and O. Etzmuß. Hierarchical techniques in collision detection for cloth animation. In *Proceedings of Winter School of Computer Graphics (WSCG 2003)*, pages 322–329, 2003. URL http://www.gris.uni-tuebingen.de/publics/paper/Mezger-2003-Hierarchical.pdf.

Anton Kai Michels, Peter Sikachev, Samuel Delmont, Uriel Doyon, Francis Maheux, Jean-Normand Bucci, and David Gallardo. Labs r&d: Rendering techniques in rise of the tomb raider. In *ACM SIGGRAPH 2015 Talks*, SIGGRAPH '15, pages 81:1–81:1, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3636-9. doi: 10.1145/2775280.2792529. URL http://doi.acm.org/10.1145/2775280.2792529.

J. Mosegaard. *Mosegaards Cloth Simulation Coding Tutorial*. 2013. URL http://cg.alexandra.dk/?p=147.

Ross-Edwards Nicholas. Ubisoft at gdc 2015. GDC '15, 2015. URL http://blog.ubi.com/ubisoft-gdc-2015/.

X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings of Graphics Interface (GI 1995)*, pages 147–154. Canadian Computer-Human Communications Society, 1995.

X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation (CAS 1997)*, pages 177–189. Springer-Verlag, 1997.

Bernhard Thomaszewski, Markus Wacker, and Wolfgang Strasser. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 107–116, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 3-905673-34-7. URL `http://dl.acm.org/citation.cfm?id=1218064.1218079`.

Pascal Volino, Martin Courchesne, and Nadia Magnenat Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 137–144, New York, NY, USA, 1995. ACM. ISBN 0-89791-701-4. doi: 10.1145/218380.218432. URL `http://doi.acm.org/10.1145/218380.218432`.

Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph.*, 30(4):71:1–71:12, July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964966. URL `http://doi.acm.org/10.1145/2010324.1964966`.

Yu Wang and Marc Olano. Isosurface smoothing using marching cubes and pn-triangles. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 132–132, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3392-4. doi: 10.1145/2699276.2721400. URL `http://doi.acm.org/10.1145/2699276.2721400`.

Jerry Weil. The synthesis of cloth objects. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 49–54, New York, NY, USA, 1986a. ACM. ISBN 0-89791-196-2. doi: 10.1145/15922.15891. URL `http://doi.acm.org/10.1145/15922.15891`.

Jerry Weil. The synthesis of cloth objects. *SIGGRAPH Comput. Graph.*, 20(4):49–54, August 1986b. ISSN 0097-8930. doi: 10.1145/15886.15891. URL `http://doi.acm.org/10.1145/15886.15891`.

Keith Wilson, Aleka McAdams, Hubert Leo, and Maryann Simmons. Simulating wind effects on cloth and hair in disney's frozen. In *ACM SIGGRAPH 2014 Talks*, SIGGRAPH '14, pages 48:1–48:1, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2960-6. doi: 10.1145/2614106.2614120. URL `http://doi.acm.org/10.1145/2614106.2614120`.