# Techniques for Animating Cloth

M. Adil Yalçın, Cansın Yıldız

***Abstract*—Cloth simulation is a must for realistic computer animated scenes, because clothes are ubiquitous, and come up in the forms of human clothing, table clothing, and non-typical cloth-like objects such as curtains, flags, leaves and even human skin and paper. Therefore, cloth modeling and animation has been a topic of research since mid 80s in the field of computer graphics, which resulted in many methods that aim to solve the many problems within. This report is aimed to provide a brief survey of some of the techniques used for cloth animation.**

## I. OVERVIEW

This report is aimed to provide a brief survey of some of the techniques used for cloth animation. After giving the basic definition and properties of real-world cloth, generic deformable objects are discussed to provide insight to representations of cloth at computer graphics. Then, the fundamental physically-based cloth simulation techniques are presented. Collision detection, which is a major problem at simulation, is discussed next. Remaining sections of this survey discuss non-physical, geometric approaches of imitating cloth behavior and also parallelization approaches of physically based models as well. This report discusses some of these approaches, for a more detailed overview, you can consult to [14].

### *A. The Basics*

A textile/cloth/garment is a flexible material consisting of a network of natural or artificial fibers often referred to as thread or yarn. Yarn is produced by spinning raw wool fibers, linen, cotton, or other material on a spinning wheel to produce long strands Figure 1 on page 1. Textiles are formed by weaving, knitting, crocheting, knotting, or pressing fibers together (felt).

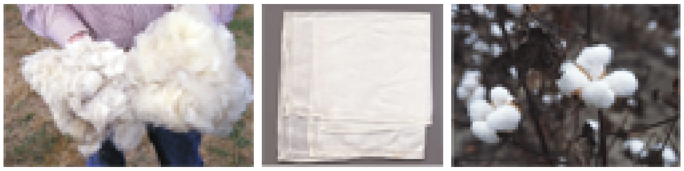


Figure 1: Some yarn materials: wool, linen and cotton

There are several ways to produce a cloth from yarns Figure 2 on page 2. Some basic production methods are:

- Weaving: The process of making woven material by interlacing yarns at right angles. The yarns that run length-ways of the cloth is called warp and that run across from side to side is called weft .
- Knitting: The process of making cloth by loops called stitches pulled through each other.
- Crochet: The needlework done by interlocking looped stitches with a crochet hook.
- Macramé: Another textile making method using knotting; a process of fastening yarns by tying.
- Felt: A fabric made of compressed matted animal fibers.

Although, above are the major types of cloth, in computer graphics perspective, there are even more varieties. Some materials like paper, flags, curtains, towels and even skin are the examples of non-typical cloth, and they can be simulated as a cloth. Yet, the methods presented below mostly focuses on woven clothes.

### *B. Properties of Cloth*

*1) Mechanical properties:* Cloth has three mechanical property; stretching, shearing, and bending Figure 3 on page 1. Stretching is the displacement of the cloth along the warps' and the wefts' direction. Shearing is the cloth displacement along two diagonal directions. Finally, bending represents the curvature of cloth surface. These three phenomena are very different from each other: a typical cloth cannot be compressed at all, and it can only be stretched to a limit of %10. But inversely, it can easily bend.

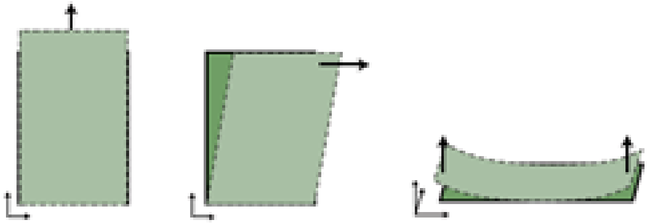


Figure 3: Stretching, shearing and bending respectively

*2) Visual Properties:* As implicitly stated above, cloth is a very flexible thin material without any elastic property. Therefore, it can easily draped onto an object and it has several wrinkles on its surface most of the time Figure 4 on page 1.

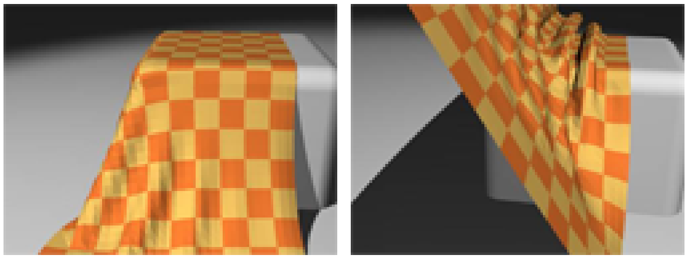
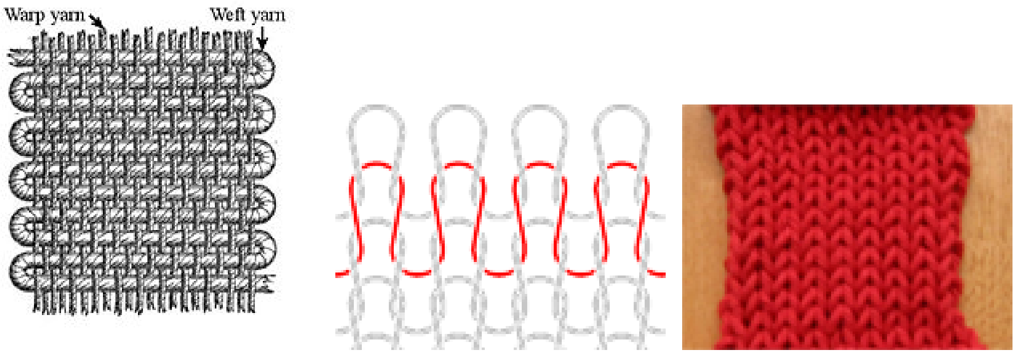


Figure 4: Draping and wrinkle patterns

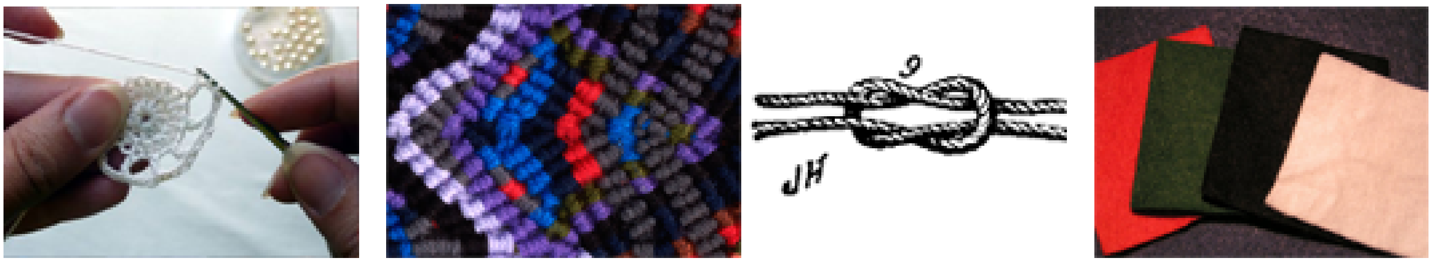
*3) Simulating Cloth Properties:* Cloth simulation is a difficult task because a cloth has;

1) Many primitives and/or nodes at model,



(a) Warp and weft in plain weaving

(b) Stockinette stitch schema and actual material for knitting



(c) A crochet and hook

(d) Macramé and Square Knotting

(e) A selection of felt clothes

Figure 2: Different types of cloths

2) High degree of freedom at those nodes,
3) Stiffness against stretch forces,
4) Variety of properties.

Also because of above same reasons, collision detection of a cloth is a major problem itself. Those difficulties lead people to a choice between realism vs. simplicity. A simple model will probably run faster but not be very realistic, inversely a complex model will probably run slower but be realistic.

## II. TRADITIONAL PHYSICAL TECHNIQUES

We will begin the cloth simulation survey with physical based techniques. The physically based methods represents cloth model as a finite number of mass points and/or some triangular/rectangular surface meshes. The resulting internal forces and/or energy functions are then computed according to each points/meshes environment. Using an *explicit* or *implicit* integration, the simulation is actuated. *Explicit integration* allows particles to be updated independently, whereas *implicit integration* couples neighboring particles, resulting in a system of equations.

“The formations or the number of neighboring points vary according to the technique. Energy-based techniques calculate the energy of the whole cloth from a set of equations and determine the shape of the cloth by moving the points to achieve a minimum energy state. Force-based techniques represent the forces among points as differential equations and perform a numerical integration to obtain the point positions at each time step. In general, energy-based techniques are used to produce sstatic simulations while force-based techniques are used in dynamic simulations”, as stated in [18].

### A. Continuum Models

Terzopoulos et al. [23] were among the first to model deformable objects using physics. In their work, they tried to give a general method for elastically deformable objects. They treated cloth as a deformable object with no thickness.

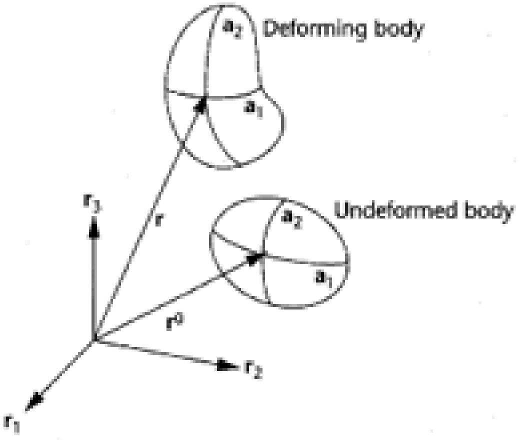


Figure 5: Deformable body representation [23]

They employ the theory of elasticity to animate their deformable objects as seen in Figure 5 on page 2. For any given object they have a set of points, whose time varying position is represented by

$$r(a,t) = [r_1(a,t), r_2(a,t), r_3(a,t)]$$

A body whose is in rest position is simply represented by

$$r^0(a) = [r^0_1(a), r^0_2(a), r^0_3(a)]$$

Given position functions, a deformable object's motion can be modeled using an equation in Lagrange's form

$$\mu\frac{\partial^2 r}{\partial t^2} + \gamma\frac{\partial r}{\partial t} + \delta_r\epsilon(r) = f(r,t)$$

In the above formulations, $r(a,t)$ is the position of the particle $a$ at time $t$, $\mu(a)$ is the mass density of the object at $a$, $\gamma(a)$ is the damping density, and $f(r,t)$ is the net external force. $\epsilon(r)$ is the net instantaneous potential energy of the elastically deformed object.

Figure 6: A flag, a soft object and a carpet from Terzopoulos' work

Each object has a potential energy of deformation $\epsilon(r)$ and its surface is discretisized by *finite-element method*, which results in a system of ordinary differential equations. Then those motion equations are numerically integrated using a semi-implicit method. Given results are very interesting and they have inspired several subsequent works (Figure 6 on page 3).

### *B. Energy-Based Particle Systems Model (Breen)*

Although continuum model led some interesting results for cloth simulation, it is not the most accurate way to deal with cloth. As Breen et al. [5], [6] stated, a cloth assembly is not held together by molecular bonds or welds like a deformable object, but rather it is bounded by friction between warps and wefts.

Breen et al. used particles to model the draping behavior of cloth. Their method treats the intersection points of warps and wefts as particles (see Figure 7 on page 3). Then, for each step of the simulation, they let the surface free fall: only gravity and collisions are considered. Resulting shape is a rough representation of the cloth. Afterwards, system energy is minimized in order to reorganize the cloth surface. The total energy $U_i$ of a particle $P_i$ is calculated as

$$U_i = U_{repel_i} + U_{stretch_i} + U_{shear_i} + U_{bend_i} + U_{gravity_i}$$

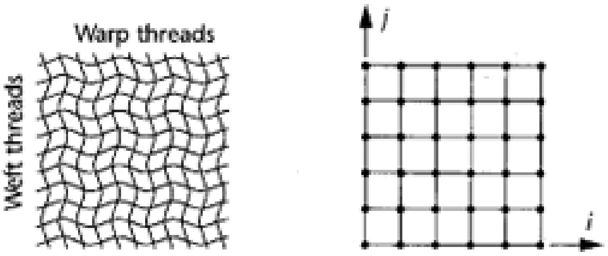
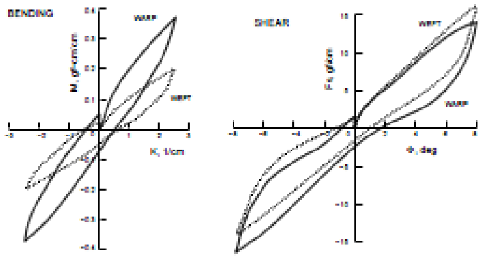


Figure 7: A woven cloth and its particle representation from Breen's work

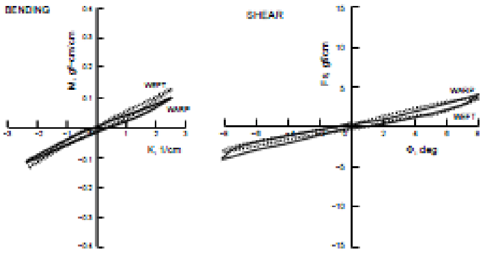
where $U_{repel_i}$ is the repulsion energy among particles, $U_{stretch_i}$ is the stretching energy, $U_{shear_i}$ is the energy due to shear (bending in the plane), $U_{bend_i}$ is the energy due to bend (bending out of the plane), and finally $U_{gravity_i}$ is the gravitational energy.

Using this formula, to minimize the total energy, they introduced an algorithm called *stochastic gradient descent*. It is a good algorithm to find local minima, but it takes lots of time to do so.

One important thing about Breen et al.'s approach is that they tried to simulate draping as realistic as possible. To do so, they measured bending, shearing and tensile properties of a cloth using a system called *Kawabata*. It is an actual mechanical system where a 20x1 cm sample cloth is tested using some machinery. A sample plot of the properties can be seen at Figure 8 on page 3. Then they derived the energy equations from those plots.



a) 100% cotton



b) 100% wool

Figure 8: Kawabata Bending and Shear Plots from Breen's work

Using particle system and Kawabata measuring, Breen et al. produced successful draping simulations of cloths (see Figure 9 on page 4).

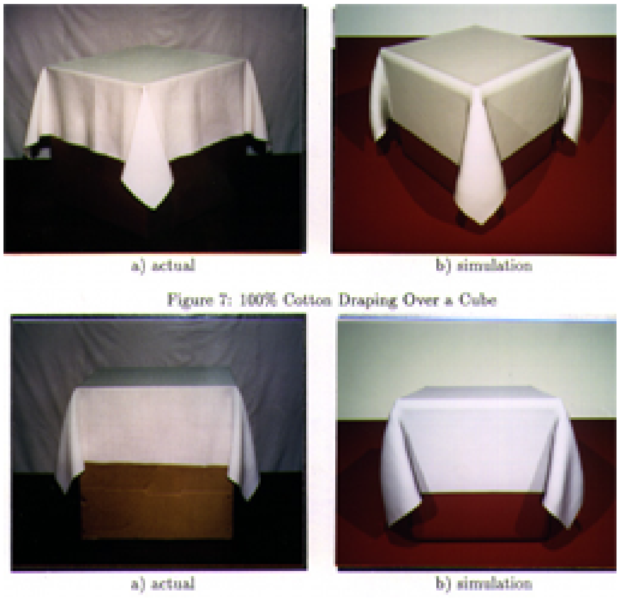


Figure 9: Actual vs. simulated cloth drapes from Breen's work

### C. Mass-Spring Model (Provot)

The continuum modeled cloth as an elastically deformable object. Later, Breen et al. suggested that it is not the best way to deal with cloth, but did not mention about reasons. In his work, Provot [20] also claims that elastically deformable objects are not an accurate representation of the cloth. The problem encountered in this modelization is that woven clothes are far from having ideal elastic properties. This is why, under some conditions, elastically represented clothes are behave more like sheets of rubber than textile. This behavior is visible especially when the model is subject to high constraints like a flag attached to a poll (The constraints will be concentrated on the attachment points).

Similar to Breen et al.'s work, Provot also uses particle systems to simulate cloth behavior. But what Provot produced is a cloth animation, whereas Breen et al. was only concerned with the static behavior of draped cloth. Unlike Breen et al., Provot uses mass-spring Forces between particles to represent the overall behavior of cloth and uses explicit Euler integration to simulate this model. He also introduced a stiffness adjustment to get rid of the super-elasticity behavior.

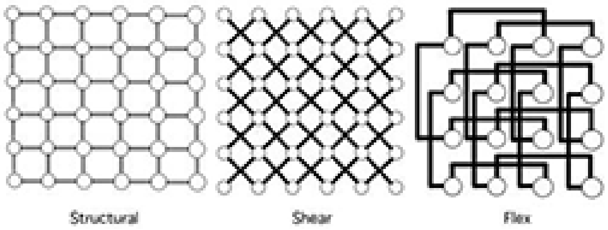


Figure 10: Structural, shear and flex springs [20]

In his work, a cloth is represented by a particle system which is held together by three different sets of springs; structural springs, shear springs, and flexion springs (see Figure 10 on page 4).

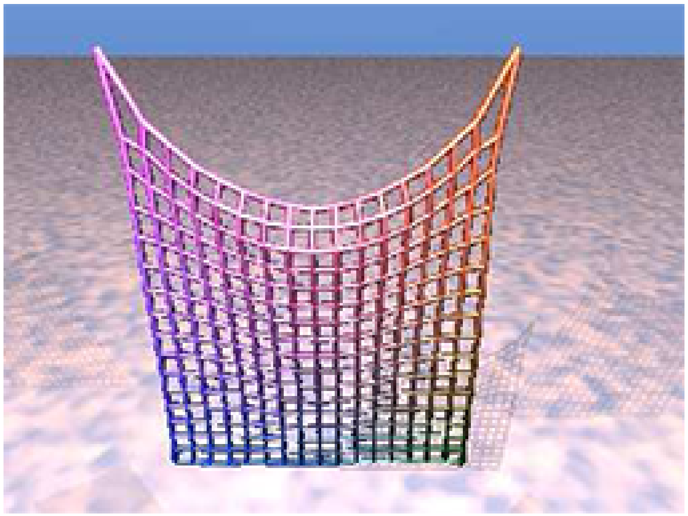
Then simply, for each point in the system, a force $F$ is calculated by adding internal spring forces $F = k * x$ and external forces (gravity, wind etc.). Using the basic explicit integration position $P_{i,j}(t+1)$ for a point $i, j$ is calculated as

$$a_{i,j}(t+\Delta t) = \frac{1}{m}F_{i,j}(t)$$

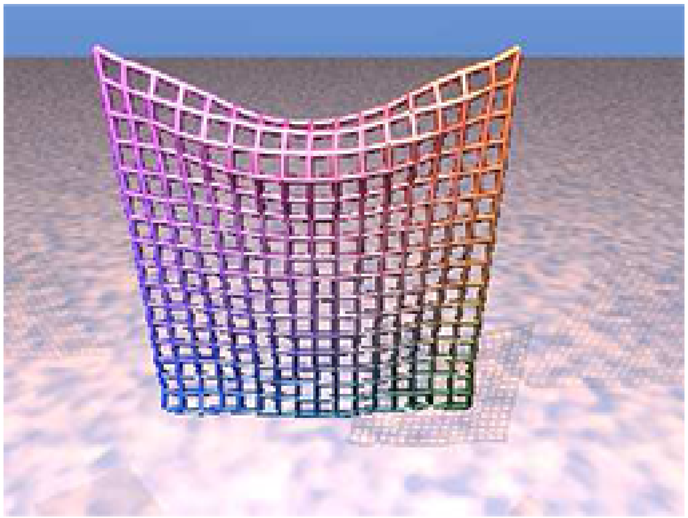
$$v_{i,j}(t+\Delta t) = v_{i.j} \times (t) + \Delta t \times a_{i,j}(t+\Delta t)$$

$$P_{i,j}(t+\Delta t) = P_{i,j}(t) + \Delta t \times v_{i.j}(t+\Delta t)$$

After implementing this algorithm, Provot observed that when a hanging cloth is simulated, unrealistic deformations occur at the pinning points. Provot solved this problem by thresholding the deformation rate in structural and shear springs around the pin points to a predetermined threshold (a limit of %10 elongation) (see Figure 11 on page 4).



(a) 200 iterations with the basic algorithm



(a) Stiffness algorithm is applied to structural and shear springs.

Figure 11: Applying stiffness to cloth [20]

### D. Large Steps in Cloth Simulation (Baraff-Witkin)

So far proposed algorithms share a common restriction. They need a small sized time steps to simulate cloth accurately. When the time steps between each simulation gets bigger, the cloth takes a chaotic shape, as demonstrated in Figure 12 on page 5. This notion makes it impossible to come up with a fast simulation of cloth.

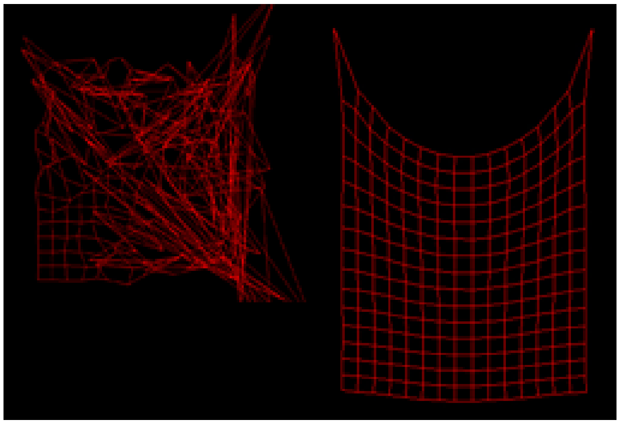


Figure 12: Large vs. Small Time Steps of a Cloth Simulation [15]

Luckily, Baraff and Witkin [3] has proposed a very interesting animation technique which does not suffer from this behavior. They represent cloth by a uniform triangular mesh. Unlike most of the researches on cloth simulation so far, they used an implicit integration method to solve the continuum formulation of the internal energy of cloth, which is similar to proposed method of Terzopoulos et al. Integration method generates, at each time step, a sparse matrix that is solved using a modified conjugated gradient. Furthermore, they have developed a technique that used an adaptive time step. Results are very interesting (as seen in Figure 13 on page 5) and computational time is very fast.

Figure 13: Results of Baraff and Witkin's work [3]

### E. Interactive animation of structured deformable objects (Barr)

The use of implicit integration, which can stably take large time steps, has been proposed by Baraff and Witkin in the context of cloth animation. As Barr et al. [11] stated, this method offers extremely low computational times, which indicates the possibility of real-time animation of simple objects. Inspired by this approach, they propose a fast and stable way to animate any mass-spring system (see Figure 14 on page 5).

Figure 14: Real-time results from Barr's work [11]

Their algorithm is somewhat a hybrid approach. They used mass-spring systems to model the cloth itself like Provot suggested, and then they borrowed the idea of an implicit integration from works of Baraff/Witkin and Terzopoulos as well. After that implicit integration, they had a post-processing process involving inverse kinematics which has the same objective as Provot's post-processing, to solve the stiffness problem.

## III. COLLISION HANDLING

### A. The Problems within Collision Detection

Collision detection and collision response in physical simulations require new ideas to be applied. A piece of cloth animated under only external forces (such as gravity) cannot react to the other physical quantities in the scene, such as other objects and even itself. Also, most collision detection schemes cannot guarantee collision-free simulations, and dealing with this imprecision can be critical if such methods can fail under some circumstances.

Many of the studies on collision detection separates collision detection and response from internal dynamics of the deformable body [8], [7], [21], and most of the discussions in this chapter follow these references. Three different criteria may exists in evaluation collision detection and resolution techniques: Plausible simulations, robustness and speed. [7] proposes to prioritize them in this given order.

The challenges involved with collision detection and response are many. Cloth is made up only a surface; it has a thin representation in the techniques. This results in the inter-penetrating cloth regions to be high visible during animation and even more, solvers may not be able to easily recover back from such errors without storing a history of the animation. The cloth consists of a high number of collide-able primitives (tens of thousand of primitives for high quality simulations), and all the primitives are in the surface and are candidates for collision. This high number of primitives result in a large degree-of-freedom and limitless configurations of cloth surface. Since number of primitives is large, number of collisions can also expected to be large in most cases, and the these collisions differ in speed, depth and physical surface parameters, such as friction, elastic or inelastic behaviors. Self intersections are harder to compute than intersection with external simpler geometries, because of $O(N^2)$ pairs of collide-able entities in a cloth with $N$ nodes. We also would like to stress that this $N$ value is large, and getting larger as the techniques and computer organization evolve, which increases the importance of fast and robust methods for dealing with self intersections.

### B. Internal Dynamics vs Contact Dynamics

As stated previously, separating internal dynamics from collision dynamics mostly simplifies many of the problems, while it can support robust techniques. A basic approach following this observation is to let internal dynamics control the simulation of the deformable cloth, and then to identify and recover from collisions, putting the cloth into a new collision-free state. Identifying and solving these collisions are at the heart of this approach.

### C. Proximity Detection and Repulsion Forces

Proximity detection is identification of close parts of the cloth object. This is performed by triangulating the cloth particles, then applying collision detection on this triangulated data to have the coordinates of the close points. In such triangulations, two common configuration arise: Point-triangle and edge-edge. The point coordinates are found as barycentric coordinates within the triangulation. The collision detection step can be accelerated using bounding volumes, which may be organized or hierarchically structured for additional increase in speed.

The colliding point (or triangle) pairs are used to generate normals in the direction of collision. Repulsion forces are computed along the normals and these forces are distributed to triangle corners (using barycentric collision coordinates). This step also requires handling of friction, which requires friction directions and forces to be computed as well. The impulse on particles are then applied to resolve the collisions. Yet, this method does not guarantee penetration-free simulations. If large repulsion forces are applied to separate collisions, the objects cannot approach one another and seem to be floating over a distance. Handling stiff repulsion forces/ springs is also computationally expensive, while it provides robustness and scalability.

### D. Robust Collisions

High-speed collisions cannot be detected efficiently by the approaches which try to recover from errors after internal dynamics are simulated. This requires identification of the exact time and position of the contact before it happens. [21] follows this approach, where the current non-intersection position and particle velocities are used to compute the next position. Finding such intersection information requires parametrization over time in volumetric space. Provot could reduce the 5th order polynomial equations which are used to detect point-triangle collisions to a cubic equations, while assuming constant velocity in his proposed collision handling pipeline throughout.

In [21], collisions are solved using inelastic collisions and a repulsion-based logic. After collision response is generated using the current state, the new state may include new collisions. This requires iterating over the collisions again. Yet, this does not guarantee converging to a collision-free solution.

To further speed up the simulation and to avoid convergence problem, rigid impact zones (or zones of impact) can be defined [21], [8]. These zones are created initially per particle, then grow as the particles collide with each other. It is based on the observation that self-colliding regions are restricted in relative motion because of friction. Yet, a careful management of these zones is required, total linear and angular momentum of the zone must be conserved through iterations and these zones should be short-lived and small, and the zone must be able to break apart, since these zones may converge to large regions which have no internal physical dynamics (rigid regions of cloth).

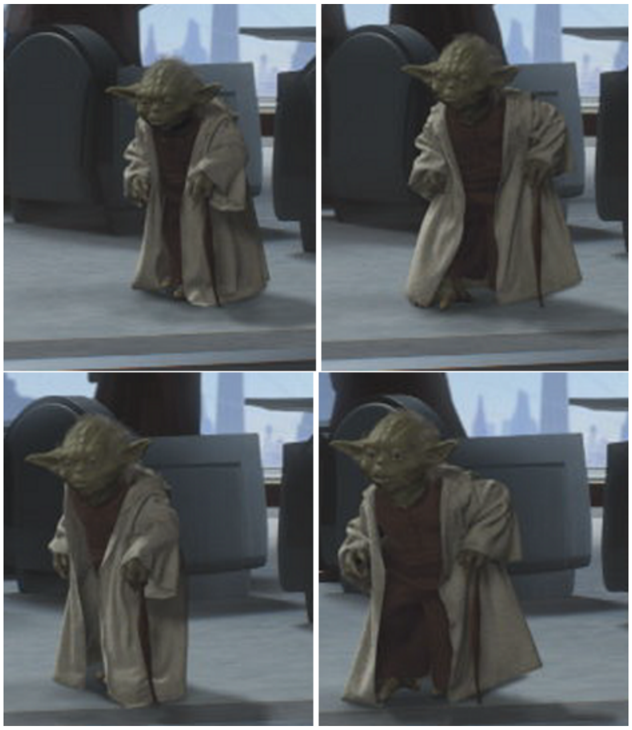


Figure 15: Collision detection as applied in industry production animations.

Variants of the methods presented in this section has been used with different internal simulation techniques of cloth structures. Figure 15 on page 6 shows frames from a production animation which has been supported by such collision detection techniques.

Figure 16: Boo’s cloth may intersect itself, but can recover gracefully later.

Another method wort mentining is proposed by Baraff, Witkin and Kass [4]. They note that one of the weak points of previously proposed algorithms is that they result in tangles in cloth after collision detection fails, or the fail-safe methods cannot handle cases where the colliding geometries intersect each other over the cloth. To deal with this problems, a history-free collision response approach (GIS) which can untangle any intersecting cloth geometries (even if the initial state is colliding), and a global collision detection method that can deal with pinches that occur during self-colliding character bodies. Flypapering makes sure that regions of cloth inside the self-interseting body regions are stable through simulations so that visual artifacts do not occur. GIS approach performs collision boundry detection, flood fills the curve regions and applies repulsion or attraction forces on the nodes of these colliding regions. The results has been used in the animation Monsters Inc. (as shown in Figure 16 on page 6).

## IV. GEOMETRIC TECHNIQUES

This section is aimed to show that cloth animation can be applied on deformed surface geometries of objects without applying physical methods as presented above. The methods described here present assume the cloth tightly wraps the surface of the model.

### A. *Cloth without Cloth*

[1] presents a simple idea for deforming surface meshes along the joints of an articulated body Figure 17 on page 7. This method can generate simple wrinkle patterns on joint regions using a very fast approach. Yet, it can only represent limited cloth behavior and requires uniform UV coordinates on the mesh so that artifacts are not generated.

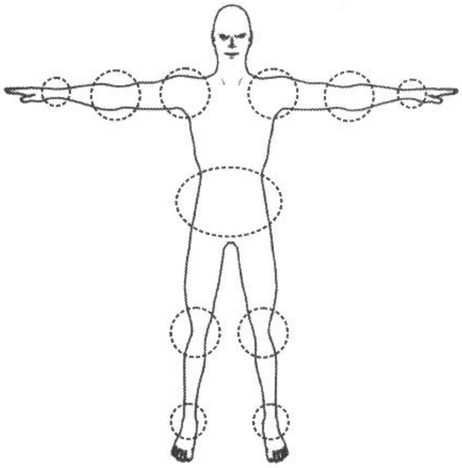


Figure 17: Candidate joint locations of a human skeleton, as usual

This method uses one texture to represent wrinkle-free surface and another one for wrinkled. Normal maps are used for this purpose in the study. Another texture defines a blending regions of these textures. Using a blending weight between two surface textures, the surface visual can be animated by applying the blending map on the regions affected by the joint rotation. The point of interest on the joint can be computed straightforward in 1 DoF joints, yet 2 and 3 DoF joints can also be used to identify the interest point, which involves projection of bone planes onto one another.

This method has been partly extended in [22], which does not assume blending on joint regions only. The key idea represented there involves creating multiple masks and wrinkle regions per mesh and blending over these multiple regions to achieve increase expressiveness. This key idea can be observed in Figure 18 on page 7.

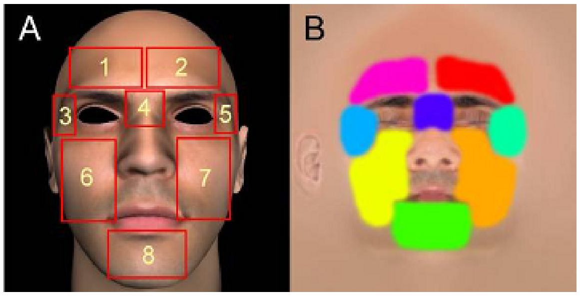


Figure 18: Proposed influence regions over a human face model [22]

### B. *Wrinkling Coarse Meshes on the GPU*

The previous method presented uses static wrinkle maps for cloth-like surface animations. The wrinkles over the surface of a triangular mesh can also be computed dynamically, as demonstrated in [16]. This method can work on top of any mesh deformation algorithm, such as bones-skinning, morphing and also physically simulated models, which is one of the strengths of this method. The method employs stages that can maintain consistency in the cloth deformations parameters computed per vertex. The shading (rendering to screen) process can involve techniques such as bump mapping and parallax mapping, which can produce high fidelity texturing on coarsely tessellated meshes, as used by interactive PC applications. The proposed method and the stages are parametrized to allow different wrinkle and compression profiles of the cloth, which can generate results as shown in Figure 19 on page 7.

Figure 19: Wrinkle patterns as computed by the method in [16]

To be able to compute deformation per vertex, the initial model if first cleared of duplicate vertices, which may have been used to generate discontinuous texturing over the model. During this clean-up procedure, a vertex adjacency texture is created for further steps of the algorithm, which can provide regional connectivity data of vertices.

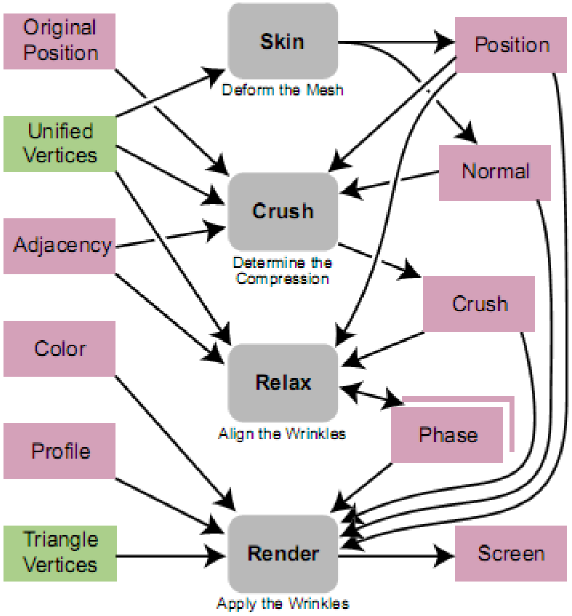


Figure 20: Steps of cloth deformation on the GPU [16]

The deformation of the models are computed in four passes, all on the GPU Figure 20 on page 8. The first pass applies skinning, although any technique which can deform initial mesh data can be used instead. The second pass computed per-vertex compression data using post and pre tangent spaces. This data is composed of wrinkle direction and amplitude. The missing phase term is computer in the next step. This step uses randomization initially and applies regressions to converge to a more global solution. The last step is rendering of the wrinkles on the surface, in which lighting and texturing is treated separately. Lighting step requires generating normals along the wrinkle "waves", which also requires adjacency information of vertices to generate the height profile. Cloth texturing is also modified using the new wrinkle waves, using parallax mapping approach. I have observed that this step may also use relief mapping for higher quality sampling of textures along the height-map profile.

## V. PARALLEL PHYSICAL TECHNIQUES

Recently, GPU's are used to effectively and interactively animate cloth-like objects under physical integrations. The models presented here are based on spring-mass model and try to identify and exploit the parallel structure of some basic solutions in cloth animation.

[25] identifies solving the sloth simulation problem on GPU using OpenGL API in defining simulation steps, also using the power of programmable shaders as exposed by GLSL. Two of the basic problems are solved in parallel in this work and its previous studies, the dynamics of cloth and the collision detection with environment. At the time of writing this report, we have not seen a parallel method which can deal with self-intersections.

The simulation outline as the following:

1) For every particle, apply external forces (such as gravity and wind)
2) In each relaxation step, for each cloth particle
    a) Evaluate the spring constraints and forces
    b) For every intersectable scene geometry, check for collisions and solve collisions by moving the particle out of collided volume.

Step 1 can be implemented using a single pass over entire particles. Also, step 2-b can be implemented in a single pass, where a single texture containing world collision data can be traversed for each particle. Yet, step 2-a requires multiple passes for each relaxation step.The integrator used in this step is chosen as the Verlet integrator, $P(t+1) = P(t)+k(P(t)-P(t-1))+\Delta t^2 F(t)$. This integrator basically works as shown in Figure 21 on page 8.

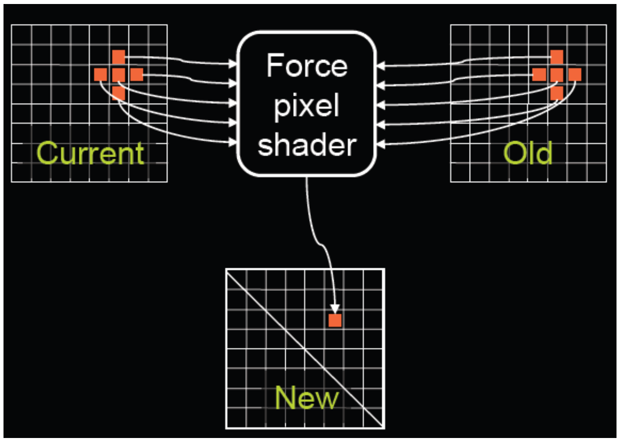


Figure 21: Performing the integration, for each global spring type

It is important that the spring types that are processed in step 2-a are independent. The initial proposal by the NVIDIA SDK 9.5 assigned each of simulated springs a separate pass, resulting in 8 passes for structural and shear type springs. The later approach, as described in [25], uses 4 passes to solve 8 springs, where each pass updates different independent regions of the cloth. Also, irregular mesh topologies can be handled using geometric images concept [13].

[25] also discusses how the new features of DirectX 10 API can be put into good use for physical cloth simulation. With the new approach, the vertex data is stored in vertex buffers instead of textures, and geometry and vertex shaders are used as the basic building block of the simulation, instead of pixel shader, which is suitable for texture-based operations. Using a simple geometry shader pass, it is also noted that up to 6 spring distance constraints can be evaluated on the geometry shader itself, while independent constraint groups are processed in vertex shader as previously noted.

ATI currently approaches physical cloth simulation in collaboration with Havok on the GPU using the emerging

OpenCL API [19] and Figure 22 on page 9. In 2009, ATI has demonstrated use of the GPU through OpenCL in Havok's OpenCL based implementation, which offers cross-platform fast physical cloth simulation to all developers . The professional solution offered by NVIDIA was based on their propriety physics engine, PhysX, which required NVIDIA hardware for acceleration.

Figure 22: Havok cloth working on AMD hardware through OpenCL API

## REFERENCES

[1] Homam Bahnassi and Wessam Bahnassi. Cloth without cloth. In *ShaderX5*.

[2] Yosuke Bando, Takaaki Kuratate, and Tomoyuki Nishita. A simple method for modeling wrinkles on human skin. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 166, Washington, DC, USA, 2002. IEEE Computer Society.

[3] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1998. ACM.

[4] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. *ACM Trans. Graph.*, 22(3):862–870, 2003.

[5] David E. Breen, Donald H. House, and Phillip H. Getto. A physically-based particle model of woven cloth. *The Visual Computer*, 8(5&6):264–277, 1992.

[6] David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372, New York, NY, USA, 1994. ACM.

[7] Robert Bridson. Cloth collision and contact. In *SIGGRAPH '05: ACM SIGGRAPH 2005 courses*, 2005.

[8] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 594–603, New York, NY, USA, 2002. ACM.

[9] Jimmy Chim and Hyunsuk Kim. Dynamic skin deformation and animation controls using maya cloth for facial animation. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 175–175, New York, NY, USA, 2002. ACM.

[10] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. *ACM Trans. Graph.*, 21(3):604–611, 2002.

[11] Mathieu Desbrun, Peter Schröder, and Alan Barr. Interactive animation of structured deformable objects. In *Proceedings of the 1999 conference on Graphics interface '99*, pages 1–8, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[12] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):to appear, 2007.

[13] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. Geometry images. *ACM Trans. Graph.*, 21(3):355–361, 2002.

[14] Donald H. House and David E. Breen, editors. *Cloth modeling and animation*. A. K. Peters, Ltd., Natick, MA, USA, 2000.

[15] Paul Jacobs. Real time cloth animation techniques - student project.

[16] Jörn Loviscach. Wrinkling coarse meshes on the gpu. In *Proceedings of Eurographics 2006, Computer Graphics Forum 25*, 2006.

[17] Patricia Moore and Derek Molloy. A survey of computer-based deformable models. *International Machine Vision and Image Processing Conference*, 0:55–66, 2007.

[18] Hing N. Ng and Richard L. Grimsdale. Computer graphics techniques for modeling cloth. *IEEE Comput. Graph. Appl.*, 16(5):28–41, 1996.

[19] Unknown 1 Online. Havok show opencl based havok cloth on ati gpus.

[20] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In Wayne A. Davis and Przemyslaw Prusinkiewicz, editors, *Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society, 1995.

[21] Xavier Provot. Collision and self collision handling in cloth model dedicated to design gaments. In *Computer Animation and Simulation '97*, 1997.

[22] C.D.G. Reis, H. Bagatelo, and J.M. Martino. Real-time simulation of wrinkles. In *WSCG 2008*, 2008.

[23] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, New York, NY, USA, 1987. ACM.

[24] Pascal Volino, Martin Courchesne, and Nadia Magnenat Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 137–144, New York, NY, USA, 1995. ACM.

[25] Cyril Zeller. Cloth simulation. White paper, NVIDIA, February 2007.