Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Simulation of Clothing with Folds and Wrinkles

R. Bridson,[1] S. Marino[2] and R. Fedkiw[3]

[1] Stanford University, rbridson@stanford.edu
[2] Industrial Light + Magic, smarino@ilm.com
[3] Stanford University, Industrial Light + Magic, fedkiw@cs.stanford.edu

Abstract

Clothing is a fundamental part of a character's persona, a key storytelling tool used to convey an intended impression to the audience. Draping, folding, wrinkling, stretching, etc. all convey meaning, and thus each is carefully controlled when filming live actors. When making films with computer simulated cloth, these subtle but important elements must be captured. In this paper we present several methods essential to matching the behavior and look of clothing worn by digital stand-ins to their real world counterparts. Novel contributions include a mixed explicit/implicit time integration scheme, a physically correct bending model with (potentially) nonzero rest angles for pre-shaping wrinkles, an interface forecasting technique that promotes the development of detail in contact regions, a post-processing method for treating cloth-character collisions that preserves folds and wrinkles, and a dynamic constraint mechanism that helps to control large scale folding. The common goal of all these techniques is to produce a cloth simulation with many folds and wrinkles improving the realism.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation, I.3.5 [Computer Graphics]: Physically based modeling

1. Introduction

Clothing is an integral part of both live action and computer generated characters. The relationship between tailoring, body composition and material selection are equally important for either medium as these qualities make each garment unique to every individual, real or fictional. Clothing folds, wrinkles and stretches to conform to its wearer, it sticks to itself and other pieces of clothing, snags, etc. The appearance of a piece of clothing comes primarily from response to these conditions, and thus it is essential for clothing created using computer graphics to model them.

A judicious choice of cloth model is imperative both to obtain the desired look and feel of the cloth, and to obtain simulation results in a reasonable amount of time. We present a mixed explicit/implicit time integration scheme that combines the flexibility of explicit methods for handling nonlinearities (such as the biphasic nature of cloth) with the speed of implicit methods. One of the keys to obtaining high levels of visual detail, i.e. dynamic folds and wrinkles, is to have a good model for bending. One of our novel contributions is a derivation of the *only* physically correct family of bending forces that act between pairs of triangles. Furthermore, our model is capable of retaining and restoring artist-sculpted features of a garment by using nonzero rest angles.

Once equipped with a reasonable dynamic model of our character's clothing, the cloth must interact with its environment. This means collision detection and modeling for both self-collision and cloth-object or cloth-character collision. There are a number of techniques for treating self-collision, either untying the cloth[60, 61, 6] or stopping all collisions before they happen[47, 30, 9]. All of these papers illustrated that it is important to have a robust self-collision strategy in order to model wrinkling and folding. For cloth-object collision, we use a level set approach modeling the (possibly deforming) objects with a signed distance function defined on a grid[41]. A number of authors have used implicit surfaces for volumetric collisions, and we illustrate that this approach flattens out the cloth—removing visual detail—by moving it to a smooth surface (the zero isocontour). We propose a new dynamic interface forecasting technique and a new post-processing technique which help to alleviate this difficulty.

At this point, one can carry out physically plausible clothing simulation, but this is nevertheless insufficient to model compelling real world clothing. Not only do many of the physical interactions which directly effect the appearance of a garment occur at a granularity much finer than the current simulation level, but also actors will wear clothes that are too long, short, baggy, heavy, etc. for dramatic effect. This is remedied on the set with a number of practical techniques, and thus we present a dynamic sticking constraint that mimics some of these techniques.

2. Related Work

We generally refer the reader to the text of House and Breen[27], the review article of Ng and Grimsdale[39], the early work of Weil[63] and Terzopoulos et al.[55], the dynamic B-splines of Thingvold and Cohen[56], the CAD apparel system of Okabe et al.[40], and the work of Breen et al.[8] using experimentally determined measurements for cloth properties.

Volino and Magnenat-Thalmann[62] compared the efficiency of a number of time integration methods. Eberhardt et al.[18] and Parks and Forsyth[42] strived to attain more accuracy in order to obtain more detailed simulations. However, for those *just* interested in speed, see the work of Desbrun et al.[16] which extends the implicit time integration of Baraff and Witkin[5] making further approximations that result in an even faster algorithm. We take a middle road, accurate enough to avoid visual artifacts such as artificial damping but fast enough for a demanding film production environment.

Models for bending forces were discussed in [55, 60, 46, 5, 24], etc. Related work on engineering simulations of "shells" is discussed in Grinspun et al.[25] for example. Wrinkling has been considered by a number of authors, for example Aono[3] proposed a "wave" model, Kunii and Gotoda[35] considered singularity theory, and Kang et al.[32] used wrinkled cubic splines. Our bending model is closely related to that of Baraff and Witkin[5], but we follow a more direct approach than taking variational derivatives of an energy. These derivatives apparently are not simple to take, and thus Baraff and Witkin suggest simplifying approximations to make the math tractable. Moreover, they do not give the exact formulas as we do below, simplifying implementation. A recent independent paper on bending by Grinspun et al.[24] following the same energy approach avoids the derivatives by using automatic differentiation, but otherwise arrives at a model similar to ours, and illustrates the ability to capture the behavior of stiff shells.

Potential field methods were pioneered in robotics[33]. These methods represent objects as zero level sets of functions and use the gradients of a potential field to repel points outside of objects. Terzopoulos et al.[55] surrounded objects with repulsive force fields making use of the inside/outside function for collisions. Sclaroff and Pentland[51] (see also Pentland and Williams[44]) proposed generalized implicit functions with fast inside/outside tests for collision detection. Deformable objects have been modeled directly with implicit surfaces[22, 15], testing the sample points of one with the inside/outside function of the other. A number of authors used spheres, ellipsoids and other implicit surfaces to detect and resolve collisions for cloth, skin, hair and muscles, see e.g. [48, 66, 58, 4, 49, 34]. Fisher and Lin[20] used a penalty based formulation to enforce a non-interpenetration constraint between deformable bodies. They used the fast marching method to partially update the distance functions only in regions where the objects are deforming or colliding. Hirota et al.[26] took a novel approach defining the signed distance function in the undeformed material coordinates of their finite element mesh in order to better deal with self interpenetration. Then instead of using individual point repulsion, they integrated over the intersection region to obtain a more accurate repulsion force. They apply this to an impressive flesh simulation using the visible human data set[59].

Jimenez and Luciani[31] modeled static friction with an adhesion force that was activated when the relative tangential velocity of two bodies in contact was below a threshold. A zero rest length spring was used to anchor the contact point to the location it was in when the adhesion force was activated. This spring is removed when the force it exerts exceeds a threshold proportional to the normal force. Chang et al.[10] introduced the concept of static links to model forces that cause hair to clump together. These are initialized by matching up the closest points between different strands, and connecting these with springs. When the length of these springs exceeds a threshold, the static link is permanently broken. The dynamic sticking constraints presented in this paper are a related mechanism to allow animators more control over the motion of clothing without it appearing unnatural.

3. Mixed Explicit/Implicit Time Integration

Our motivation for introducing yet another time integration method for cloth simulation is to combine the flexibility and simplicity of explicit methods (such as Runge-Kutta) with the efficiency of implicit schemes (such as backward Euler). In particular, we use explicit integration on the elastic forces (those that are independent of velocity) and implicit integration on the damping forces (the velocity-dependent forces). The algorithm is presented below, with x denoting positions, v velocities, and a accelerations:

- $v^{n+1/2} = v^n + \frac{\Delta t}{2} a(t^n, x^n, v^n)$ (explicit)
- Modify $v^{n+1/2}$ to get $\tilde{v}^{n+1/2}$ to limit strain, etc.
- $x^{n+1} = x^n + \Delta t \tilde{v}^{n+1/2}$ (explicit)
- $v^{n+1} = v^{n+1/2} + \frac{\Delta t}{2} a(t^{n+1}, x^{n+1}, v^{n+1})$ (implicit)
- Modify v^{n+1} in place to limit strain, etc.

Essentially we are combining the explicit second order accurate leapfrog scheme for the position update with an implicit second order accurate trapezoidal rule for the velocity update. To see that the velocity update is indeed the trapezoidal rule substitute $v^{n+1/2}$ from the first line into the fourth line to obtain

$$v^{n+1} = v^n + \tfrac{\Delta t}{2}\left(a(t^n,x^n,v^n) + a(t^{n+1},x^{n+1},v^{n+1})\right).$$

Overall this is a second order accurate method that is stable for $\Delta t < O(\Delta x\sqrt{\rho/k^e})$ where ρ is the density, k^e is the material stiffness, and Δx is the smallest edge length of the mesh, independent of how large the damping forces are. This is a critical point, since a fully explicit scheme like Runge-Kutta has a much more stringent quadratic time step restriction dependent on the damping forces, $\Delta t < O(\Delta x^2\rho/k^d)$ where k^d is the material damping parameter. The extra factor of Δx means an order of magnitude more steps are required. For example, if a dimensionless $\Delta x = 0.1$ requires ten time steps per frame, then $\Delta x^2 = 0.01$ would require one hundred steps. Another alternative would be to use a fully implicit scheme with larger step sizes[5], but this introduces significant artificial damping in the elastic modes. Since we only go implicit on the damping modes, we only introduce this extra artificial damping where there is damping already. Recently, the artificial damping present in the implicit method of Baraff and Witkin[5] was mitigated by the carefully tailored bending model of Choi and Ko[11]. Since we do not need to alleviate this artificial damping, we are instead free to develop a more sophisticated bending model that isn't restricted to rectangular grids and flat rest positions (see section 4).

An additional advantage of only going implicit on velocities is that while the forces in cloth simulation are typically nonlinear and depend on positions in a complicated manner, the damping forces are often linear in the velocities with a symmetric Jacobian. This means we can use a fast conjugate gradient solver without need for simplifying approximations such as those made in Baraff and Witkin[5]. Furthermore, this solution is very fast: typically less than ten and sometimes as few as one or two iterations are needed, even without preconditioning. Since no preconditioner is needed, we don't need to explicitly assemble a matrix, reducing overhead and allowing a simple black-box structure where the conjugate gradient solver calculates damping forces simply through function calls. For additional speed, the position-dependent terms in the damping forces (which are held constant during these iterations) may be computed once and cached.

The use of an explicit update for position allows us to modify the velocities to enforce a wide variety of constraints, and further allows a strain limiting procedure[46, 9] where springs are limited to a maximum 10% extension beyond their rest length. For practical purposes, this means we can use weaker springs (with a more relaxed time step restriction), and qualitatively capture the biphasic nature of fabric. An additional innovation introduced here is to limit the *compression* of the springs to 0% in the same manner, forcing cloth to buckle out of the plane into attractive folds and wrinkles instead of unrealistically compressing (see Choi and Ko[11] for an extended discussion of this behavior, though we observe that the technique in that paper cannot generalize to unstructured meshes with curved rest positions as used here). Naturally, the limits 0% and 10% can be adjusted to match the desired look of the simulated clothing: stretchier materials will have looser limits.

For highly damped materials the above algorithm is stable but not "monotone". Spurious oscillations may be introduced because the first explicit half step of the velocity may overshoot before being damped back down to stability by the implicit half step. This spurious velocity at $v^{n+1/2}$ may cause difficulty when used to update the position. We observe that even fully implicit methods suffer from these oscillations with large time steps when second order or higher accuracy is required, e.g. the BDF2 method used in Choi and Ko[11]. Moreover, in line two (above) we specifically process the velocity in order to improve its character for the position update, highlighting the importance of a good midstep velocity at $t^{n+1/2}$. These difficulties can be alleviated by simply switching the order of the velocity steps, using the fact that backward Euler is unconditionally monotone:

- $v^{n+1/2} = v^n + \frac{\Delta t}{2}a(t^n,x^n,v^{n+1/2})$ (implicit)
- Modify $v^{n+1/2}$ in place to limit strain, etc.
- $x^{n+1} = x^n + \Delta t v^{n+1/2}$ (explicit)
- $v^{n+1} = v^{n+1/2} + \frac{\Delta t}{2}a(t^{n+1},x^{n+1},v^{n+1/2})$ (explicit)

This corresponds to the central Newmark scheme (see e.g. Hughes[29]). Note that the trapezoidal rule is used between $v^{n-1/2}$ and $v^{n+1/2}$, not v^n and v^{n+1} as before, thus we only get second order accuracy and stability (independent of damping forces) if the time step Δt remains constant between time steps. If a variable time step is desired, as is usually the case, we no longer have the trapezoidal rule and lose both accuracy and stability. We recommend an improved approach that updates the position with an implicitly calculated velocity at $t^{n+1/2}$, but still uses the trapezoidal rule between v^n and v^{n+1} (not $v^{n-1/2}$ and $v^{n+1/2}$):

- $\tilde{v}^{n+1/2} = v^n + \frac{\Delta t}{2}a(t^n,x^n,v^{n+1/2})$ (implicit)
- Modify $\tilde{v}^{n+1/2}$ in place to limit strain, etc.
- $x^{n+1} = x^n + \Delta t\tilde{v}^{n+1/2}$ (explicit)
- $v^{n+1/2} = v^n + \frac{\Delta t}{2}a(t^n,x^n,v^n)$ (explicit)
- $v^{n+1} = v^{n+1/2} + \frac{\Delta t}{2}a(t^{n+1},x^{n+1},v^{n+1})$ (implicit)
- Modify v^{n+1} in place to limit strain, etc.

This allows variable step sizes with second order accuracy and monotone behavior at the expense of an additional implicit solve. Lines one, four, and five of the original algorithm are preserved as the last three lines of this algorithm, updating the velocity with the trapezoidal rule. The first three lines of this new algorithm use the more stable first three lines of the second algorithm (above) to obtain a robust position update.

4. An Accurate Model for Bending

The physics of cloth bending are poorly understood. The dynamics of anisotropic fibers twined together and woven into a sheet of fabric constantly interacting with massive deformations and friction is certainly more difficult to model with a two-dimensional continuum than for example steel. However, several basic qualitative properties of such a model can be identified that are essential for a plausible simulation, and without these a model is incorrect.

In order to handle unstructured triangle meshes and get finer, more robust control over bending than in vertex-centric models, we posit as our basic bending element two triangles sharing an edge. Our bending elements will be based on the dihedral angle and its rate of change, as in Baraff and Witkin[5] and extended in Grinspun et al.[24]. We label the element as in figure 1, with vertex positions x_i and velocities v_i, $i = 1,\dots,4$, and angle θ between the normals n_1 and n_2.

The vector of the four velocities $v = (v_1, v_2, v_3, v_4)$ and the vector of bending forces $F = (F_1, F_2, F_3, F_4)$ live in a 12 dimensional linear space. One can select a basis for this space identifying twelve distinct "modes" of motion. For bending it is natural to select for the first eleven modes the three rigid body translations, the three (instantaneous) rigid body rotations, the two in-plane motions of vertex 1, the two in-plane motions of vertex 2, and the one in-line stretching of edge 3–4. None of these change the dihedral angle, and thus should not participate in bending force calculations. This leaves the twelfth mode, the bending mode, which is the unique mode orthogonal to the other eleven up to an arbitrary scaling factor. This mode changes the dihedral angle but does not cause any in-plane deformation or rigid body motion. Let us call it $u = (u_1, u_2, u_3, u_4)$. From the condition of orthogonality to the in-plane motions of vertices 1 and 2, we find that u_1 is parallel to $\hat{n}_1$ and u_2 is parallel to $\hat{n}_2$. From the condition of orthogonality to the in-axis stretching of edge 3–4, we see that $u_4 - u_3$ must be in the span of $\hat{n}_1$ and $\hat{n}_2$. Orthogonality to the rigid body translations implies that the sum $u_1 + u_2 + u_3 + u_4$ is zero, and hence $u_3 + u_4$ is also in the span of $\hat{n}_1$ and $\hat{n}_2$, thus u_3 and u_4 are each in this span. Finally, after making u orthogonal to rigid rotations (which we can conveniently choose to be about the axes $\hat{n}_1$, $\hat{n}_2$ and $\hat{e}$) we end up with

$$u_1 = |E| \frac{N_1}{|N_1|^2} \qquad u_2 = |E| \frac{N_2}{|N_2|^2}$$

$$u_3 = \frac{(x_1 - x_4) \cdot E}{|E|} \frac{N_1}{|N_1|^2} + \frac{(x_2 - x_4) \cdot E}{|E|} \frac{N_2}{|N_2|^2}$$

$$u_4 = -\frac{(x_1 - x_3) \cdot E}{|E|} \frac{N_1}{|N_1|^2} - \frac{(x_2 - x_3) \cdot E}{|E|} \frac{N_2}{|N_2|^2}$$

up to an arbitrary scaling factor, where $N_1 = (x_1 - x_3) \times (x_1 - x_4)$ and $N_2 = (x_2 - x_4) \times (x_2 - x_3)$ are the area weighted normals and $E = x_4 - x_3$ is the common edge. Thus u_1 and u_2 are inversely proportional to their distance from the common edge, and u_3 and u_4 are a linear combination of u_1 and u_2 based on the barycentric coordinates of x_1 and x_2 with respect to the common edge. The bending elastic and damping forces *must* be proportional to this mode. One immediate observation is that orthogonality to rigid body modes implies these forces conserve linear and angular momentum. In fact, every bending model based on two triangles that does *not* use exactly these force directions will violate either the fundamental conservation laws or will influence in-plane (i.e. non-bending) deformations. While some may argue that in reality, in-plane and bending deformations are subtly coupled, the exact nature of this coupling varies between materials and is not understood for even the simplest fabrics, thus it is wisest to avoid adding arbitrary and artificial coupling.

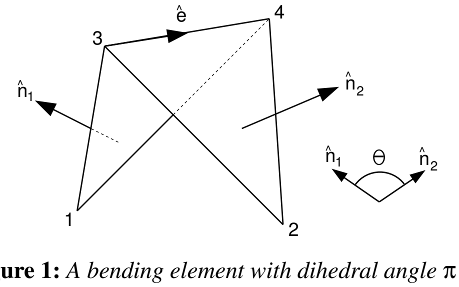

Figure 1: *A bending element with dihedral angle $\pi - \theta$.*

For simplicity we choose the magnitude of elastic force so that

$$F_i^e = k^e \frac{|E|^2}{|N_1| + |N_2|} \sin(\theta/2)\, u_i,$$

for $i = 1,\dots,4$. The elastic bending stiffness k^e is a mesh-independent material property, the middle factor scales this according to the anisotropy of the mesh (so the look of the cloth doesn't change significantly with remeshing), and the sine factor measures how far from flat the cloth is. This is the simplest quantity to compute that smoothly and monotonically increases from a minimum when sharply folded at $\theta = -\pi$, is zero when flat at $\theta = 0$, and rises to a maximum at the other sharply folded state $\theta = \pi$. We use the formula $\sin(\theta/2) = \pm\sqrt{(1 - \hat{n}_1 \cdot \hat{n}_2)/2}$ where the sign is chosen to match the sign of $\sin\theta$, which is just $\hat{n}_1 \times \hat{n}_2 \cdot \hat{e}$. Naturally more complex nonlinear models, e.g. including powers of θ for increased resistance at sharper angles, are possible. But we stress that these factors must multiply all of the forces so that the force directions and proportionalities do not change.

In many cases an artist desires that particular folds should consistently appear in a character's clothing to define their look. Even the best tailoring may not do this when the character is in motion, but one tool in cloth simulation that can overcome this is sculpting folds directly into the garment. We can straightforwardly model this with non-zero rest angles. Other manifolds such as skin, skin-tight synthetic suits, molded rubber, etc. also naturally have non-flat resting positions. To account for these we use

$$F_i^e = k^e \frac{|E|^2}{|N_1|+|N_2|}\left(\sin(\theta/2) - \sin(\theta_0/2)\right) u_i,$$

for $i = 1,\dots,4$ where θ_0 is the rest angle. Nonzero rest angles are also useful in other areas: e.g. Grinspun et al.[24] derived a similar model to simulate stiff but curved materials such as hats and creased paper, Chang et al.[10] used nonzero rest angles to model curly hair styles, and Nedel and Thalmann[38] designed "angular springs" to aid in volume preservation of muscle tissue. Figures 3 and 4 show examples of nonzero rest angles with widely different bending stiffnesses.

The damping component of the bending forces depends on the rate of change of the dihedral angle and acts to slow it down. Taking the time derivative of $\cos\theta = \hat{n}_1 \cdot \hat{n}_2$ we find after simplification that

$$d\theta/dt = u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3 + u_4 \cdot v_4.$$

Parenthetically, this is essentially an alternative derivation of the bending mode u, i.e. it is the gradient (the steepest ascent direction) of the dihedral angle. Then the damping bending forces are

$$F_i^d = -k^d |E| (d\theta/dt) u_i,$$

where we again include a mesh scaling factor $|E|$ so that k^d is a material property and may be kept constant as the mesh is refined, possibly anisotropically. This can be rewritten as $F^d = -k^d |E| (u \otimes u) v$, showing that the damping forces are linear in the velocities and have a symmetric negative semi-definite Jacobian matrix that allows efficient linear solves (with conjugate gradients, for example) for implicitly integrating the velocities. Again, while a more complex non-linear function of $d\theta/dt$ may be used for the force magnitude (any such function will preserve symmetry), a model that uses a combination of the velocities other than $\sum_i u_i \cdot v_i$ or generates forces in different directions is not correct: such a model is damping motion other than bending.

5. Collisions

Robust collision detection and response is essential to ensure that digital clothing interacts properly with the character and its environment in a visually realistic and pleasing manner. This fundamental mechanism is, after all, the means by which a digital actor's performance is mapped onto their costume.

Here we concentrate on the problem of cloth–object collisions: for the separate problem of cloth self collision we use the technique presented in Bridson et al.[9]. The two can be combined by making adjustments for object collisions before running the self-collision code.

5.1. Initializing Level Set Collision Objects

We start out by initializing our grids with small positive values, and then any point interior to an object is assigned a small negative value. Then a fast marching method[57, 52] can be used to turn this into a signed distance function. For rigid bodies, this is a one time cost at the beginning of the simulation, but for deformable objects these signed distance functions need to be updated at every time step. This can be accomplished with methods that are entirely local to the interface (only operating in a band) and are thus computationally inexpensive[1, 43]. Moreover, since simulations often proceed in layers and the animation of the character and flesh simulation are usually complete before cloth simulation begins, a one time cost can be incurred to pre-process the level set collision objects for an entire animation. Then they can be used for multiple cloth simulations without incurring any additional cost. We minimize the storage costs by using adaptive distance functions (ADF's)[21] which are stored in an octree data structure, and we note that numerical algorithms have been devised to solve level set equations on these structures[54, 53]. The nodal values of the fine grid are constrained to match those of the coarse grid wherever gradation occurs in order to alleviate difficulties with discontinuities, see e.g. Westermann et al.[64]. Signed distance functions can also be sculpted or modeled[45, 37, 13] or obtained from range images[12, 65, 14].

We will also make use of object velocities in our method (see below). For deformable bodies, we map the local velocity of the object to the points interior to the object, and then use a velocity extrapolation method[2] to extend the velocities to the grid points outside the object as well (see also Enright et al.[19] where this was used to define water velocities in the air region). Note that this does not need to be done for rigid bodies since their pointwise velocities are intrinsically defined throughout space by their translational and rotational components.

For each frame of animation, we map the signed distance functions for each body part onto a single octree grid by choosing the smallest potential value at each grid point. Points that are interior to two or more level sets can be flagged as being caught in between two body parts (e.g. in the armpit), and the combined level set will have a local minimum on a surface between these body parts. Barraquand and Latombe[7] noted that spurious minima could appear between two volumes or in concave regions of a single volume, and advocated filling regions or randomization to escape these regions. Problems with spurious minima in concave regions were also pointed out by Gibson[23]. In the case of pinching cloth, we simply want the cloth to remain on this surface of local minima as noted by Baraff et al.[6] who introduced a method for accomplishing this.

When our pre-calculated level set collision objects are required at times in between the cached values, it is simple to interpolate the necessary information as has been done for motion blur[19]. Other methods exist as well, for example rasterizing the time swept surface[50].

5.2. Collision Detection and Response

Although Duff[17] observed that merely checking points from one object inside the other is not sufficient to catch all collisions—more complex tests may be needed especially when the sample points do not adequately resolve the object—we have found the pointwise approach sufficient with our densely sampled cloth models. Consider a collision between a point p on our cloth with velocity v_p and the level set collision object. At the spatial location of p, we use the local level set value ϕ, the outward pointing normal $N = \nabla\phi$, and the local velocity of the level set collision object v.

A novel component of our method is that we *do not* simply push points with $\phi < 0$ to the surface of the collision object in the direction of $\nabla\phi$. Instead we compute the distance p must travel along N to resolve the future interference of the point with the object. We refer to this as *interface forecasting* and note that it is similar in spirit to precontact breaking forces[36] in that they enhance both stability and smoothness of the result. Gibson[23] suggested the possibility of anticipating collisions, and we propose a novel numerical algorithm that exploits this suggestion here. In a time interval $\triangle t$, we predict that our point will move to the $\phi^{new} = \phi + \triangle t(v_p - v)\cdot N$ isocontour, where the second term accounts for the relative motion between the point and the collision object in the normal direction. If ϕ^{new} is positive, the point will end up outside the level set and nothing needs to be done, but otherwise we need to process a collision. First, we calculate the normal and tangential velocities for both the point and the level set using the scalar $v_N = v \cdot N$ and the vector $v_T = v - v_N N$. Then we update both the normal and tangential components to obtain $v_p^{new} = v_{p,N}^{new} N + v_{p,T}^{new}$. The new normal velocity is calculated as $v_{p,N}^{new} = v_{p,N} - \phi^{new}/\triangle\tau = v_N - \phi/\triangle\tau$ which predicts the point to lie exactly on the zero isocontour after a time $\triangle\tau$. Note that this algorithm is different than just using the collision object from the next time step as $\triangle\tau$ is in general a relaxation time. Moreover, we can add friction to the relative tangential velocity $v_{T,rel} = v_{p,T} - v_T$ using the formula from Bridson et al.[9],

$$v_{T,rel}^{new} = \max\left(0, 1 - \mu \frac{v_{p,N}^{new} - v_{p,N}}{|v_{T,rel}|}\right) v_{T,rel}$$

where μ is the friction coefficient. This handles both the static and kinetic friction cases. Finally, $v_{p,T}^{new} = v_T + v_{T,rel}^{new}$. Figure 5 demonstrates this collision response algorithm.

5.3. Preserving Folds And Wrinkles

Accurately and completely resolving collisions is of the utmost importance for any cloth animation system. However, in doing so it is paramount to retain the high frequency features which occur at close points of contact between cloth and objects. Such imperfections are the substance of reality, and it is in these subtleties that computer generated clothing achieves (or fails to achieve) a sustainable illusion. When processing collisions as outlined in the last section, there is no guarantee that all intersections are removed. In fact, practical experience seems to indicate that allowing this freedom enables the development of more folds and wrinkles. Moreover, after the simulation the resulting mesh may be subdivided and/or mapped to a different geometric primitive for rendering. If we simply projected interpenetrating points to the surface[28, 9], we would smooth out folds and wrinkles. We instead propose a new post-processing method that preserves these important features.

Instead of projecting all points to the zero isocontour flattening out detailed wrinkles that happen to penetrate the collision volume, we project everything into an interval $[0, \tau]$ where τ is a user defined tolerance. (To ensure no intersection of the cloth triangles with objects, one can easily project the points into $[\varepsilon, \tau]$ with $\varepsilon > 0$.) Points with ϕ values larger than τ are not modified, and otherwise we define a monotonically increasing function f that maps $[-\infty, \tau]$ to $[0, \tau]$. The monotonicity of f preserves the up and down wavy patterns characteristic of wrinkles, i.e. all the cloth points inside the collision volume are brought above the surface into the interval $[0, \tau]$ without significantly compressing their relative depths. The only practical restrictions on f are that it quickly fall off to zero by an estimate of the deepest likely cloth interpenetration $\hat{\tau}$ so that we essentially map $[-\hat{\tau}, \tau]$ to $[0, \tau]$. This allows as much room as possible for the cloth wrinkles to be expressed in the interval $[0, \tau]$. For each point, the algorithm uses the local ϕ value and the function f to calculate what the new ϕ value should be. Once all points know their desired locations, we iteratively move them in the appropriate direction, i.e. $\pm N$, until they reach their desired ϕ isocontour. Figure 2 illustrates this idea and figure 5 shows results obtained using this procedure. It should be mentioned that this process should not be applied to points in the cloth mesh which are pinched between two or more level sets, as such a procedure will produce a visually incorrect result by erroneously moving points whose desired positions reside within $\phi < 0$ (see Baraff et al.[6]).

6. Dynamic Sticking Constraints

There are many times that a physically accurate simulation of the cloth will give a visually unappealing result. In live action filming, this is remedied with a number of practical techniques including adding weights in hems, taping parts of the clothing to the body, artificially sewing parts of the clothing together, etc. Clothing is controlled on the set, and thus we need to control it during our simulations. To this end we present a practical dynamic sticking constraint between cloth and either itself or object geometry.

Potential constraints are initialized through the specification of two types of regions: *adhering regions* are specified

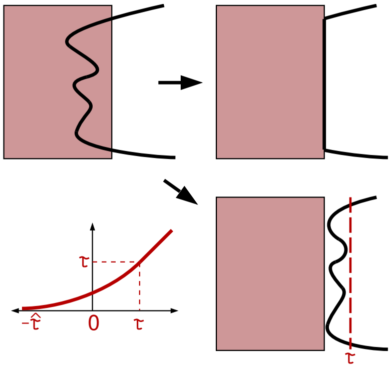

Figure 2: *Cloth penetrating an object may be pushed just to the surface (upper right) destroying wrinkles, or may be pushed to a band outside the object using our monotone mapping (lower right) preserving wrinkles.*

on the cloth mesh and *sticky regions* are denoted on either object geometry or on the cloth mesh itself. We implement these regions as time varying attributes so that interactions can be animated and timed to yield specific performances. Regardless of the object or cloth geometry upon which a sticky region is defined, we triangulate these sticky regions and constrain the velocity of the triangle vertices so that the triangles move with the underlying geometry. Then when a node from an adhering region of the cloth moves within some user prescribed tolerance of a triangle in a sticky region, the constraint is activated and the current offset of the adhering point in the triangles local coordinate system is recorded. Our goal is to constrain the adhering point to stay in this same relative location translating and rotating with the triangle. This is similar to Chang et al.[10], except that our constraints can be broken and formed many times during a simulation similar to Jimenez and Luciani[31]. Since our method is applied dynamically, a given adhering point may come within a user prescribed tolerance of many triangles and thus have offsets linking it to many points. We resolve this by attracting our adhering point to the average position dictated by the constraining triangles.

Instead of attracting our adhering point to its target position with a zero rest length spring[31, 10], we instead dynamically (i.e. the spring location and length change every time step) attach a zero rest length spring between the predicted location of the adhering point, $x_{new} = x_{old} + \triangle\tau v$, and the predicated location of the target point at the end of a user prescribed time interval $\triangle\tau$. Similar to Chang et al.[10], we deactivate the constraint if the distance between the adhering point and the target point exceeds a threshold, and the spring constant smoothly decreases to zero as this threshold is approached. This gives a smooth deactivation of the dynamic constraint. Figure 6 illustrates our dynamic sticking constraints with a digital garment.

7. Conclusions

Throughout this paper we have stressed methods that preserve folds and wrinkles in cloth simulations. This included a mixed explicit/implicit time integration scheme, a derivation of physically correct bending forces with possibly nonzero rest angles for modeling wrinkles into the cloth, an interface forecasting collision response method for enhanced dynamic behavior, a new post-processing technique that pushes cloth into an interval while preserving relative depths and thus wrinkles, and a dynamic sticking constraint for controllability.

8. Acknowledgements

Research supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N00014-03-1-0071, ONR N00014-02-1-0720, NSF ITR-0121288 and NSF ACI-0205671. In addition, R. B. was supported in part by a Stanford Graduate Fellowship.

We would like to thank Dennis Turner, Samir Hoon and the Terminator 3 production; Juan Luis Sanchez and the Harry Potter 2 production; and Ryan Kautzman, Nick Rasmussen, Steve Sullivan and Cliff Plumer of Industrial Light + Magic.

We would also like to thank David Baraff for pointing out an error in a preliminary version of this manuscript. This led to the correct scaling for the bending forces.

References

1. D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, (118):269–277, 1995.
2. D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.
3. M. Aono. A wrinkle propagation model for cloth. In *Proc. 8th International Conf. of the Computer Graphics Society on CG International '90*, pages 95–115, 1990.
4. A. Aubel and D. Thalmann. Realistic deformation of human body shapes. In *Proc. Computer Animation and Simulation*, pages 125–135, 2000.
5. D. Baraff and A. Witkin. Large steps in cloth simulation. *Comput. Graph. (SIGGRAPH Proc.)*, pages 1–12, 1998.
6. D. Baraff, A. Witkin, and M. Kass. Untangling cloth. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 22, 2003.
7. J. Barraquand and J.-C. Latombe. Robot motion planning: a distributed representation approach. *Int'l. J. Robotics Research*, 10(6):628–649, 1991.
8. D. E. Breen, D. H. House, and M. J. Wozny. Predicting the drape of woven cloth using interacting particles.

Comput. Graph. (SIGGRAPH Proc.), pages 365–372, 1994.

9. R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:594–603, 2002.

10. J. T. Chang, J. Jin, and Y. Yu. A practical model for hair mutual interactions. In *Proc. ACM SIGGRAPH Symposium on Computer Animation*, pages 77–80, 2002.

11. K.-J. Choi and H.-S. Ko. Stable but responsive cloth. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:604–611, 2002.

12. B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Comput. Graph. (SIGGRAPH Proc.)*, pages 303–312, 1996.

13. B. Cutler, J. Dorsey, L. McMillan, M. Müller, and R. Jagnow. A procedural approach to authoring solid models. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21(3):302–311, 2002.

14. J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proc. First International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 428–438. IEEE, 2002.

15. M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *Proc. SIGGRAPH 95*, pages 287–290, 1995.

16. M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Graphics Interface*, pages 1–8, 1999.

17. T. Duff. Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry. *Comput. Graph. (SIGGRAPH Proc.)*, pages 131–137, 1992.

18. B. Eberhardt, O. Etzmuß, and M. Hauth. Implicit-explicit schemes for fast animation with particle systems. In *Proc. of Eurographics Workshop on Computer Animation and Simulation*, pages 137–151, 2000.

19. D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:736–744, 2002.

20. S. Fisher and M. C. Lin. Deformed distance fields for simulation of non-penetrating flexible bodies. In *Comput. Anim. and Sim. '01*, Proc. Eurographics Workshop, pages 99–111, 2001.

21. S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proc. SIGGRAPH 2000*, pages 249–254, 2000.

22. M.-P. Gascuel. An implicit formulation for precise contact modeling between flexible solids. In *Proc. SIGGRAPH 93*, pages 313–320, 1993.

23. S. F. F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *Proc. of IEEE Symp. on Vol. Vis.*, pages 23–30, 1998.

24. E. Grinspun, A. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *ACM Symp. Comp. Anim.*, 2003.

25. E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:281–290, 2002.

26. G. Hirota, S. Fisher, A. State, C. Lee, and H. Fuchs. An implicit finite element method for elastic solids in contact. In *Comput. Anim.*, 2001.

27. D. H. House and D. E. Breen, editors. *Cloth modeling and animation*. A. K. Peters, 2000.

28. P. Howlett and W. T. Hewitt. Mass-spring simulation using adaptive non-active points. In *Computer Graphics Forum*, volume 17, pages 345–354, 1998.

29. T. J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Prentice Hall, 1987.

30. S. Huh, D. N. Metaxas, and N. I. Badler. Collision resolutions in cloth simulation. In *Computer Animation*. IEEE, 2001.

31. S. Jimenez and A. Luciani. Animation of interacting objects with collisions and prolonged contacts. In B. Falcidieno and T. L. Kunii, editors, *Modeling in computer graphics—methods and applications*, Proc. of the IFIP WG 5.10 Working Conference, pages 129–141. Springer-Verlag, 1993.

32. Y.-M. Kang, J.-H. Choi, H.-G. Cho, and D.-H. Lee. An efficient animation of wrinkled cloth with approximate implicit integration. *The Visual Computer*, 17(3):147–157, 2001.

33. O. Khatib and J. F. Le Maitre. Dynamic control of manipulators operating in a complex environment. In *Proc. 3rd Int'l. CISM-IFToMM Symposium*, pages 267–282, 1978.

34. T.-Y. Kim and U. Neumann. Interactive multiresolution hair modeling and editing. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21(3):620–629, 2002.

35. T. L. Kunii and H. Gotoda. Modeling and animation of garment wrinkle formation processes. In *Proc. Computer Animation*, pages 131–147, 1990.

36. W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel samping. *Comput. Graph. (SIGGRAPH Proc.)*, pages 401–408, 1999.

37. K. Museth, D. Breen, R. Whitaker, and A. Barr. Level set surface editing operators. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21(3):330–338, 2002.

38. L. P. Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. In *Proc. Computer Graphics International*, pages 156–165, 1998.

39. H. N. Ng and R. L. Grimsdale. Computer graphics techniques for modeling cloth. *IEEE Computer Graphics and Applications*, pages 28–41, 1996.

40. H. Okabe, H. Imaoka, T. Tomiha, and H. Niwaya. Three dimensional apparel CAD system. *Comput. Graph. (SIGGRAPH Proc.)*, pages 105–110, 1992.

41. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.

42. D. Parks and D. Forsyth. Improved integration for cloth simulation. In *Proc. of Eurographics*, Computer Graphics Forum. Eurographics Assoc., 2002.

43. D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, (155):410–438, 1999.

44. A. Pentland and J. Williams. Good vibrations: modal dynamics for graphics and animation. *Comput. Graph. (Proc. SIGGRAPH 89)*, 23(3):215–222, 1989.

45. R. N. Perry and S. F. Frisken. Kizamu: a system for sculpting digital characters. *Comput. Graph. (SIGGRAPH Proc.)*, pages 47–56, 2001.

46. X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface*, pages 147–154, May 1995.

47. X. Provot. Collision and self-collision handling in cloth model dedicated to design garment. *Graphics Interface*, pages 177–89, 1997.

48. R. E. Rosenblum, W. E. Carlson, and E. Tripp III. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *Journal of Visualization and Computer Animation*, 2(4):141–148, 1991.

49. I. Rudomin and J. Castillo. Distance fields applied to character animation. In *Proc. of Eurographics*, volume 21 of *Computer Graphics Forum*. Eurographics Assoc., 2002.

50. W. J. Schroeder, W. E. Lorensen, and S. Linthicum. Implicit modeling of swept surfaces and volumes. In *Proc. of Vis.*, pages 40–55. IEEE Computer Society Press, 1994.

51. S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. *Comput. Graph. (Proc. SIGGRAPH 91)*, 25(4):247–250, 1991.

52. J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.*, 93:1591–1595, 1996.

53. J. Strain. Fast tree-based redistancing for level set computations. *J. Comput. Phys.*, 152:664–686, 1999.

54. J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

55. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Comput. Graph. (Proc. SIGGRAPH 87)*, 21(4):205–214, 1987.

56. J. A. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. *Comput. Graph. (SIGGRAPH Proc.)*, pages 129–137, 1992.

57. J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control*, 40:1528–1538, 1995.

58. Russel Turner and Enrico Gobbetti. Interactive construction and animation of layered elastically deformable characters. *Computer Graphics Forum*, 17(2):135–152, 1998.

59. U.S. National Library of Medicine. The visible human project, 1994. http://www.nlm.nih.gov/research/visible/.

60. P. Volino, M. Courchesne, and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. *Comput. Graph. (SIGGRAPH Proc.)*, pages 137–144, 1995.

61. P. Volino and N. Magnenat-Thalmann. Accurate collision response on polygonal meshes. In *Proc. of Computer Animation*, pages 154–163, 2000.

62. P. Volino and N. Magnenat-Thalmann. Comparing efficiency of integration methods for cloth simulation. In *Proc. Computer Graphics International*, pages 265–274, 2001.

63. J. Weil. The synthesis of cloth objects. *Comput. Graph. (SIGGRAPH Proc.)*, pages 49–54, 1986.

64. R. Westermann, L. Kobbelt, and T. Ertl. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Vis. Comput.*, 15(2):100–111, 1999.

65. R. T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.

66. J. Wilhelms and A. Van Gelder. Anatomically based modeling. *Comput. Graph. (SIGGRAPH Proc.)*, pages 173–180, 1997.