A Mass Spring Model for Hair Simulation

Andrew Selle* Stanford University Industrial Light + Magic

Michael Lentine Stanford University

Ronald Fedkiw Stanford University Industrial Light + Magic

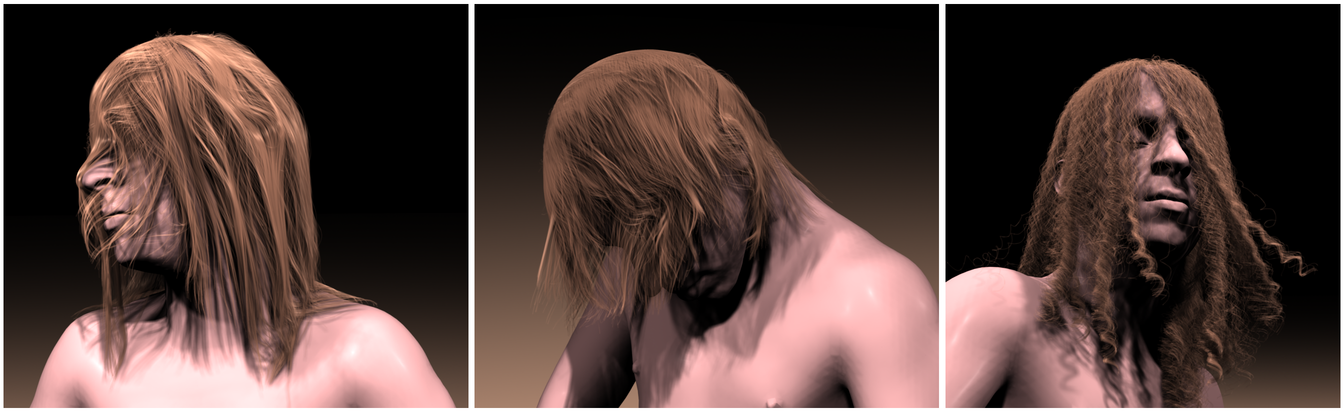


Figure 1: *Hair simulations without using any interpolation hairs. (Left) Ten thousand long straight hairs. (Middle) Ten thousand medium length straight hairs. (Right) Five thousand long curly hairs.*

Abstract

Our goal is to simulate the full hair geometry, consisting of approximately one hundred thousand hairs on a typical human head. This will require scalable methods that can simulate every hair as opposed to only a few guide hairs. Novel to this approach is that the individual hair/hair interactions can be modeled with physical parameters (friction, static attraction, etc.) at the scale of a single hair as opposed to clumped or continuum interactions. In this vein, we first propose a new altitude spring model for preventing collapse in the simulation of volumetric tetrahedra, and we show that it is also applicable both to bending in cloth and torsion in hair. We demonstrate that this new torsion model for hair behaves in a fashion similar to more sophisticated models with significantly reduced computational cost. For added efficiency, we introduce a semi-implicit discretization of standard springs that makes them truly linear in multiple spatial dimensions and thus unconditionally stable without requiring Newton-Raphson iteration. We also simulate complex hair/hair interactions including sticking and clumping behavior, collisions with objects (e.g. head and shoulders) and self-collisions. Notably, in line with our goal to simulate the full head of hair, we do not generate any new hairs at render time.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling;

Keywords: hair simulation, mass-spring models

1 Introduction

Hair dynamics is one of the most challenging phenomena to simulate because of the sheer number of hairs on the head and the complexity of the motion. Even so, hair simulation has important applications for visual effects, animated features, virtual hair styling and online stores. Consequently, researchers have developed methods for its simulation, such as the seminal work of [Rosenblum et al. 1991; Anjyo et al. 1992] (see also the survey [Ward et al. 2007b]). Most research has managed the complexity of hair simulation by focusing on the bulk behavior of hair, but in this paper, we seek to capture the subtle and intricate details of hair, requiring us to consider the non-collective behavior.

Some authors suggest that simulating every hair on the head is unnecessary as hair behaves as a collective (see e.g. [Petrovic et al. 2005]); they instead use aggregate hair simulation techniques. While the efficiency of these approaches is attractive, simulating aggregate geometry also has drawbacks. In particular, using fewer degrees of freedom (DOFs) forces hair to behave collectively even in situations where it should not (e.g. trendy, messy or tangled hairstyles). Increasing the DOFs by simulating each individual hair, as we propose, can capture these effects, but it also requires handling more interaction constraints explicitly. Another drawback of clumped models is that even though hair volume is easily maintained by using fluid incompressibility or repulsions with large radii, intricate collisions and interactions with small or high curvature objects (e.g. the ear, a comb, etc.) cannot be resolved. Additionally, the geometric inconsistency between simulated and rendered geometry can produce artifacts such as self or body interpenetrations. Lastly, subtle details such as stray hairs, clump separation, etc. are not handled by aggregate models. To address the above drawbacks, we propose using a non-clumped simulation technique. Though this requires more computation, this allows us to achieve non-collective behavior.

In this paper, we attempt to simulate as many individual hairs as possible by using a constitutive model that is simple and fast but also can model twist and curly hair. To do this, we use a mass/spring model which is computationally inexpensive and adept at modeling

*e-mail:{aselle,mlentine,fedkiw}@cs.stanford.edu both object collisions and self-collisions. This also has the advantage of being easily adaptable to an existing cloth or flesh simulator, thus leveraging existing time integration and interaction/collision handling. Unfortunately, mass-spring models have difficulty modeling twist, so we introduce a new altitude-spring based hair constitutive model that can model torsion. Fortuitously, the same model we use for hair torsion also provides an improved bending model for cloth and shape preservation for tetrahedra. We also improve time integration by introducing an unconditionally stable implicit linear spring model and a biased strain limiting approach for hair. Body collisions are more efficiently/accurately handled by introducing a better temporal interpolation scheme for level sets. Hair/hair interactions are modeled by using geometric collisions and a stiction model. Our method is demonstrated in several examples in which we only render simulated hairs.

2 Related Work

Though our focus is simulation, modeling hair geometry and styles is also an important research topic see [Kim and Neumann 2002; Choe and Ko 2005; Ward et al. 2006]. While these methods have modeled geometry with hair counts approaching the number in the head (e.g. [Yu 2001] uses fifty thousand hairs), techniques for hair simulation have typically simulated fewer.

2.1 Aggregate Hair Simulation

Researchers use different degrees of hair aggregation to manage computational complexity. From most aggregate to least, these can be categorized as continuum, clumped strand, adaptive clumped strand and non-clumped strand models. Early work focused on single hair non-aggregate simulation using either mass/spring models [Rosenblum et al. 1991] or projective dynamics [Anjyo et al. 1992] (a length-preserving generalized coordinate model), but both methods neglect torsion and self-collision/interactions. Thus, these methods parallelize trivially as each strand has no data dependencies on any other strand, meaning they could be easily scaled to 100,000 hairs. Nevertheless, the lack of twist and interaction modeling of these methods would not yield high fidelity simulations.

Hair modeled as a continuum implicitly treats interaction, which improves efficiency while maintaining hair volume. [Hadap and Magnenat-Thalmann 2001] uses a fluid continuum that models contact using viscosity together with rigid body chains embedded in the volume. [Bando et al. 2003] models hair using unconnected particles embedded in a fluid continuum, but their renderings show that sometimes stray particles appear to float in space. [Petrovic et al. 2005] simulates mass/spring guide hairs (neglecting twist), rasterizing them onto a level set grid to model interaction and volume, producing stylized and uniform hair. [Gupta et al. 2006] uses a simplified mass/spring mechanical model on lattice points to define a deformation field for embedded rendering hairs. While continuum approaches are efficient at modeling bulk behavior, they have difficulty modeling discontinuous effects like hair separation, and the highly intricate interacting geometry like curly hair, or individual hair twist (which we target).

Other authors have used a small set of discrete aggregate hair wisps or clumps that are simulated separately. [Plante et al. 2002] uses a mass-spring envelope lattice per guide hair in which rendering hairs are embedded. Unlike our mass-spring model, theirs is designed to be used with large envelope volumes (especially for interactions). [Choe et al. 2005] simulates guide hairs consisting of rigid body links attached with springs, synthesizing rendered hairs with a statistical model; collisions are modeled using penalty forces. [Hadap 2006] uses a sparse set of articulated rigid body (ARB) chains to produce hairs, foliage, and character ears. Collisions require the solution of a quadratic program making it less desirable for dense hairs. [Bertails et al. 2006] also simulates hair using a finite-element constitutive model (discussed below) using penalty methods for collisions.

Even though clumped models do obtain some discontinuous behaviors not handled by continuum models, clumped models still tradeoff some accuracy for efficiency. To capture more detail in only some areas of a model, authors have recently considered adaptive clumped models. [Bertails et al. 2003] uses a tree of varying size rigid body wisps and provided heuristics for forming and breaking. [Ward et al. 2003; Ward and Lin 2003] use projective dynamics and three discrete levels of detail (LODs) based on hair motion or camera distance to allow real-time simulation. While targeted at realtime, this method could be used with different settings to achieve higher quality simulations offline. However, the fundamental assumption of all adaptive schemes is that some areas do not require high fidelity, meaning computational savings can be achieved by varying simulation resolution. Unfortunately, this is not always the case; for example, in highly dynamic flipping or wind motions, the same detail is needed everywhere. Here, uniform approaches are more desirable as they do not incur the overhead of managing adaptivity. Nevertheless, if adaptivity is appropriate, these schemes provide a useful framework within which our mass-spring model could be integrated (see future work).

2.2 Strand dynamics

Researchers also have considered many mechanical models for strands which can often be applied outside of their original aggregate context. In fact, many papers only demonstrate a single strand, even though their mechanical models could be applied to hair. [Pai 2002] introduced the high order Cosserat (finite-element) model for flexible rods, although he did not consider dynamics. [Grégoire and Schömer 2006] modeled flexible tubes while also using a Cosserat-based scheme. [Spillmann and Teschner 2007] used a spatially adaptive dynamic Cosserat model for interactively tying knots in ropes in real-time. [Brown et al. 2004] also simulated knot tying, instead using an articulated rigid body system; gravity was neglected to help user interaction. The preceding three methods use interpenetration to detect collisions, so they would miss collisions if strand thickness was reduced to our hair strand size or if velocity was increased. [Wang et al. 2005] uses a specialized solver that includes torsion as a state variable to simulate thread.

Researchers have used mass-spring systems, projective dynamics, articulated rigid bodies, and other generalized or finite elements models as discussed above. Mass-spring models are advantageous as they are simple, commonly implemented, and handle collisions and constraints easily, but typically cannot model torsion, curly hair, or exact length preservation. Projective dynamics solves the length preservation problem at the expense of using generalized coordinates, making collisions and constraints more difficult. Generalized coordinate ARB schemes conserve length and allow twist to be modeled, but also make collisions and constraints more difficult. ARBs are also relatively expensive per body, especially if linear complementarity problem (LCP) collision methods are used. Higher order finite element models like Cosserat schemes (e.g. [Bertails et al. 2006]) can create intricate helical guide curves using very few DOFs. However, if one wanted to simulate thousands of interacting hairs, more DOFs per hair would be required to resolve collisions and contact, making higher-order methods less attractive. In particular, [Bertails et al. 2006] scales quadratically as the number of elements per hair are increased, and other high order methods are also more expensive per element. Moreover, the use of generalized coordinates in [Bertails et al. 2006], [Spillmann and Teschner 2007], and [Wang et al. 2005] makes collision/contact response more challenging. A summary of the tradeoffs of various strand models (including ours) is shown in Table 1.

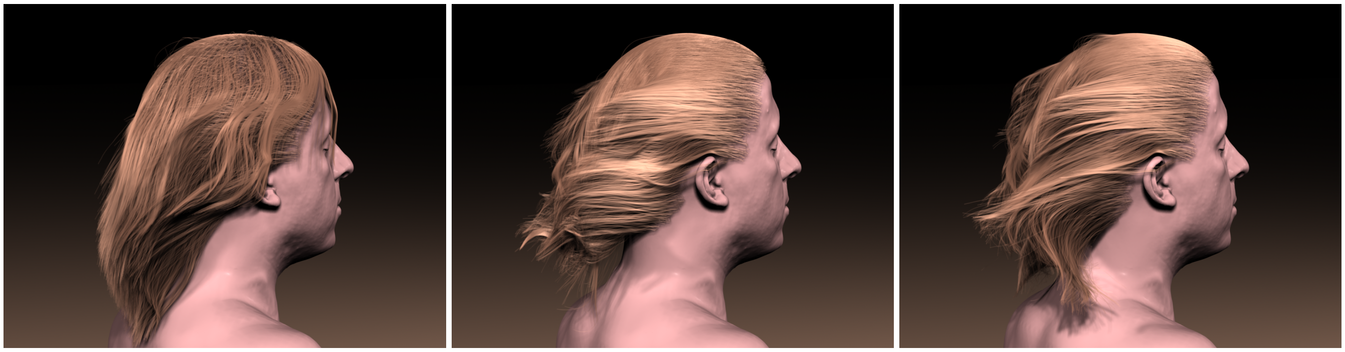


Figure 2: *A simulation of 10,000 medium length straight hairs with 25 segments each (500,000 total particles) on a stationary head while interacting with wind.*

3 Constitutive Model

Since our goal is to model as many realistically interacting hairs as possible, we prefer a mass-spring model. For efficiency, we use tetrahedra instead of representing coordinate systems explicitly. Furthermore, we avoid generally slower finite element based models, instead using simple linear stress-strain springs. In addition, to preserve effective coordinate systems, we propose a new and improved altitude spring model that can preserve a tetrahedron's volume.

3.1 Tetrahedral Altitude Springs

A major difficulty in volumetric simulations (both mass-spring and finite volume) is that a tetrahedron may collapse to zero volume or even invert to negative volume. In traditional finite elements, this inversion causes the simulation to fail (which can be remedied by [Irving et al. 2004]). In the case of a mass spring model with springs on each of the six tetrahedron edges, the tetrahedron can collapse to a plane and never recover because all spring force directions are in that plane. If the tetrahedron is inverted, the springs have an artificial rest state where the tetrahedron has the same absolute value of volume as the rest shape.

To prevent these problems, [Molino et al. 2003] places an *altitude spring* between each particle of the tetrahedron and a virtual node projected onto the plane of the opposite face (see Figure 3 (a) and (b)). Equal and opposite forces are applied to both the particle and the virtual node where the virtual node distributes its force barycentrically to the particles of the face. Unfortunately, altitude springs have problems in the case of highly stretched or degenerate tetrahedra as the virtual node can be outside the triangle. In this case, barycentric weights can be negative resulting in arbitrarily large forces on particles of the triangle (i.e. large forces on non-negative weights balance out large forces on negative weights). One might try to correct this by using only point/face pairs with non-negative barycentric weights, but in many configurations, no such altitude springs exist (Figure 3(c)).

Phenomena	Mass/ Spring	Proj. Dyn.	Rigid (chain)	Super- helices	Our Method
Bending	Yes	Yes	Yes	Yes	Yes
Torsion	No	No	Yes	Yes	Yes (alt. spring)
Non-stretching	No	Yes	Yes	Yes	Yes (strain limit)
Curliness	No	No	No	Yes	Yes (alt. spring)
Constraints	Easy	Tricky	Tricky	Tricky	Easy

Table 1: *Adapted table from [Ward et al 2007] that shows a comparison between various hair methods. The last column shows that our method relieves several problems with mass/spring models.*

The key to solving this problem is noting that in the point/face degenerate case two edges are *crossing*, which yields a non-degenerate direction for restoring the positive volume of the tetrahedron. In particular, the edge/edge altitude spring direction is mutually orthogonal to the two lines containing the edges (Figure 3(d)). Each spring endpoint is embedded on one of the lines using barycentric weights. Here, the barycentric weights can also be negative if the virtual nodes are not on the segment (Figure 3(e)) in which case the edge/edge altitude spring can also produce arbitrarily large forces. Fortunately, for any tetrahedron, there is at least one point/face or edge/edge altitude spring that has non-negative barycentric weights. *For any tetrahedron, the edge/edge or point/face spring that currently has the least length is guaranteed to have all non-negative barycentric weights, preventing unbounded forces* (see Appendix). Thus, we choose to use the single edge/edge or point/face spring that currently has the least length.

Point/face altitude springs compute length as $l = 6V/\|\boldsymbol{u} \times \boldsymbol{v}\|$ where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the vectors of the base triangle and V is the signed volume of the tetrahedron, while edge/edge altitude springs

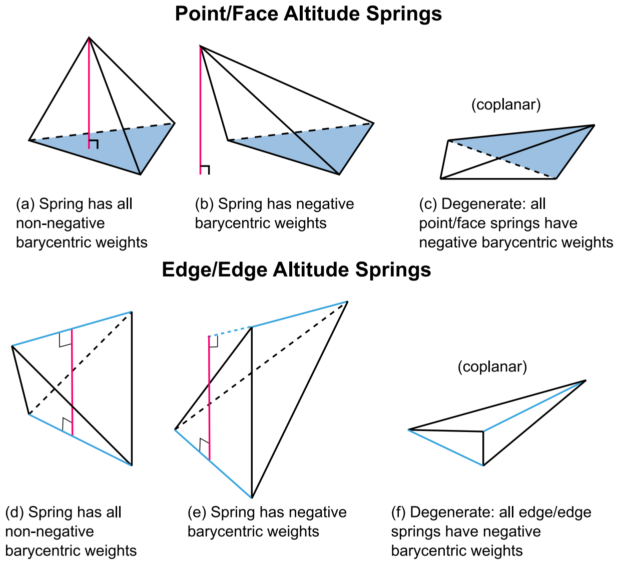
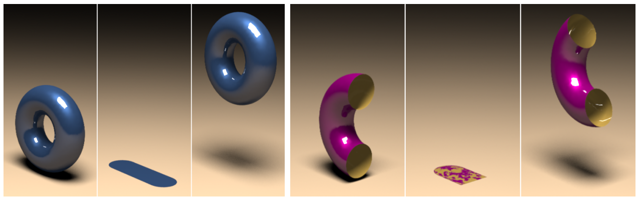

Point/Face Altitude Springs
(coplanar)
(a) Spring has all non-negative barycentric weights
(b) Spring has negative barycentric weights
(c) Degenerate: all point/face springs have negative barycentric weights
Edge/Edge Altitude Springs
(coplanar)
(d) Spring has all non-negative barycentric weights
(e) Spring has negative barycentric weights
(f) Degenerate: all edge/edge springs have negative barycentric weights

Figure 3: *Varying examples of point/face and edge/edge altitude springs are shown. Examples (a) and (d) represent ideal cases for the two types of altitude springs, whereas (b) and (e) show altitude springs of each type with negative barycentric weights. (c) shows all point/face altitude springs having negative weights, but there is still one edge/edge altitude spring that has non-negative weights. (f) shows all edge/edge altitude springs having negative weights, but one point/face altitude spring has non-negative weights.*



(a) (b)

Figure 4: *(a) A volumetric torus with no forces collapses to a puddle and subsequently recovers after the spring strengths are increased. (b) A similar test performed using a thin shell.*

use the same formula but with $\boldsymbol{u}$ and $\boldsymbol{v}$ replaced by the edge vectors. There are seven potential springs, 4 point/face and 3 edge/edge. A valid altitude spring exists unless all particles become colinear, i.e. every tetrahedron face is degenerate and every opposing edge is parallel. Thus to robustly select which spring to use, we find the largest $\|\boldsymbol{u} \times \boldsymbol{v}\|$ of any possible edge/edge or point/face spring. If $\|\boldsymbol{u}\|$ or $\|\boldsymbol{v}\|$ is too small, then an edge has collapsed, and it will be restored by normal edge springs so no altitude spring is used. If $\sin^2\theta = \|\boldsymbol{u} \times \boldsymbol{v}\|^2/(\|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2)$ is too small, then particles in the tetrahedron are colinear, and we use an edge/edge spring with a direction orthogonal to the line. Otherwise, the altitude spring is valid and guaranteed to have non-negative barycentric weights. We illustrate the efficacy of our altitude springs in Figure 4(a) where a tetrahedral torus model collapses to zero height and then the stiffness is increased on both the normal edge springs and our altitude springs causing the model to recover (similar to the test in [Irving et al. 2004]).

We also briefly consider a simple bending model for cloth or shells, where linear springs are placed on the triangle edges while bending springs connect opposite vertices across an edge shared by two triangles. If two triangles become coplanar this spring is in-plane and cannot recover the rest curvature, so an axial bending spring can be added that connects a virtual node on the shared edge to a virtual node on the bending spring (Figure 5). However, this is exactly an edge/edge altitude spring with the two triangles and the bending spring representing a tetrahedron. Thus instead of an axial bending spring we use our generalized altitude spring model described above. A simulation using this model is shown in Figure 4(b) where a shell model collapses to zero height, and then the stiffness of all springs is increased allowing it to recover.

3.2 An Altitude Spring Hair Model

If we model a hair as a series of connected line segments, stretching can be modeled with edge springs between every consecutive particle, and bending can be modeled with bending springs between every other particle. The edge springs and bending springs together form triangles that implicitly represent the orientation of the hair. Twist can be modeled by attaching torsion springs that connect each particle to a particle three particles away from it (see Figure 7(a)).

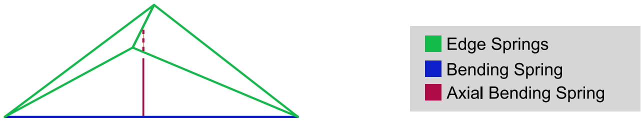


Figure 5: *A pair of triangles sharing an edge can have its bending modeled by two springs, a bending spring between the unshared vertices and an axial bending spring between a virtual node on the shared edge and a virtual node on the bending spring. The bending spring edge and the shared edge form a tetrahedron so this axial bending spring is really an edge/edge altitude spring.*

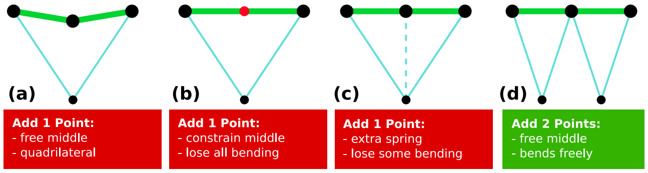


Figure 6: *Various options for creating perturbed points to form triangles in straight hair. (d) yields both non-degenerate triangles and free bending.*

These springs imply tetrahedra allowing the use of our generalized altitude springs to prevent collapse. This scheme works well on curly hair where no set of three consecutive particles is colinear so every tetrahedron has non-zero volume. Unfortunately, for straight hair all the particles are colinear meaning that the triangles formed by edge and bending springs have zero area so orientation (and thus twist) cannot be represented. We address this by introducing additional particles perturbed from the main hair axis to obtain non-zero area triangles.

If we duplicate and perturb the middle particle of two colinear segments, we could handle the original middle particle in three ways. If we did not constrain it, a quadrilateral is formed that can deform arbitrarily (Figure 6(a)). If we constrained it, then all bending degrees of freedom are lost (Figure 6(b)). If we left it unconstrained, but also attached a spring between it and the perturbed particle, bending is incorrectly constrained in one direction (Figure 6(c)). Thus, instead of creating a single new particle, we create two new particles at the midpoints of the two segments and perturb them to form two triangles (Figure 6(d)). These triangles should be fairly rigid so they are given springs on their edges that are as strong as the hair edge springs. From these triangles we can add bending and torsion springs to form a full hair model as shown in Figure 7(b) where a tetrahedron is formed whenever a torsion springs connects two opposite particles of a triangle pair. The process of ensuring triangles are non-degenerate can be seen as a way of ensuring a path of torsion springs through the hair (Figure 8). Thus instead of using

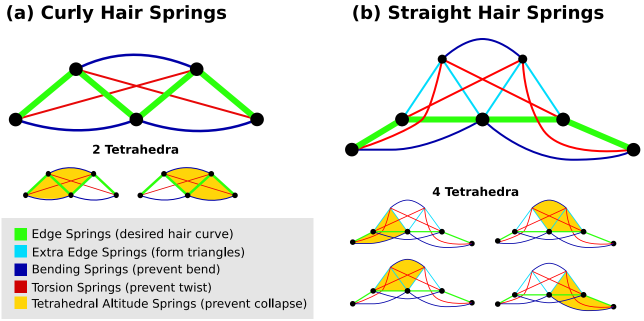

(a) Curly Hair Springs
(b) Straight Hair Springs
2 Tetrahedra
4 Tetrahedra
🟩 Edge Springs (desired hair curve)
🟦 Extra Edge Springs (form triangles)
🟦 Bending Springs (prevent bend)
🟥 Torsion Springs (prevent twist)
🟨 Tetrahedral Altitude Springs (prevent collapse)

Figure 7: *Straight and curly hair models using edge, bending, torsion, and altitude springs preserving the implied tetrahedra.*

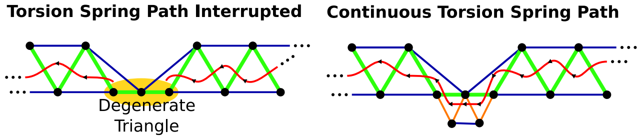

Torsion Spring Path Interrupted
Continuous Torsion Spring Path
Degenerate Triangle

Figure 8: *Triangles define orientations for penalizing twist, and torsion springs "trace" a continuous path through the non-degenerate triangles — but they are blocked at straight hair segments (left). The subdivision and perturbation of our method removes degeneracies so the path becomes continuous (right).*

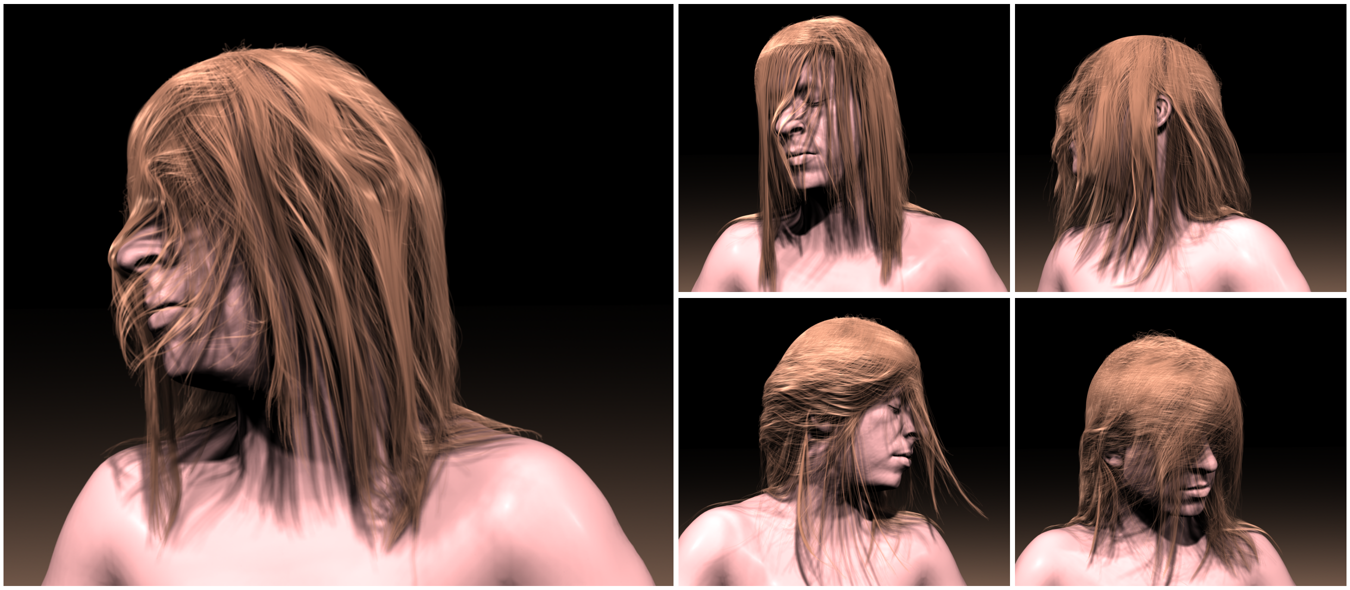


Figure 9: *A simulation of 10,000 long straight hairs with 50 segments each (1,000,000 total particles) on a character shaking his head from side to side.*

an explicit coordinate system we model an implicit one by using offset particles together with extra springs, causing a marginally higher simulation cost, but still fitting into a simple mass-spring framework.

Thus our algorithm prepares a curve for simulation by first sampling discrete points $x_1, \ldots, x_m$ equally in arclength. Then a segment x_i, x_{i+1} is subdivided if it is colinear with x_{i-1} or x_{i+2}. We perturb new particles off of the original curve such that the edge lengths of the newly created triangles are equal to the length of the parent segment. In addition, if there are many consecutive segments with perturbed particles, we rotate the perturbed particles about the hair axis to ensure good direction sampling. Afterwards, we consider every segment in order and build the appropriate spring connections. Figure 10 demonstrates the plausibility of our approach on single hair examples by comparing videos of real hair to our simulations. Figure 11 shows that we can simulate different curly stiffnesses (top) and degrees of curliness (bottom). We also note that one could always subdivide and perturb instead of detecting colinearity, albeit at the cost of unnecessary particles. One downside of this model is that there is some mixing of stretch, bend, and torsion forces which is analogous to using bending springs in cloth.

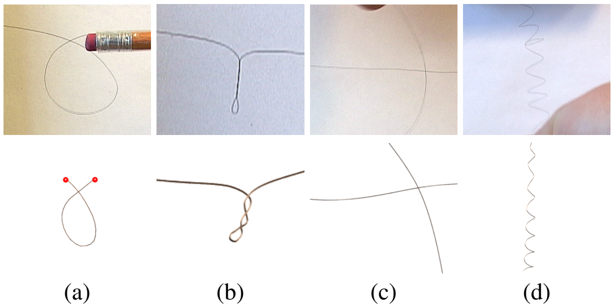


Figure 10: *Four single hair tests showing various behaviors observed in real hair (top) and reproduced by our model (bottom). (a) a hair is forced on either endpoint to buckle into a loop. (b) a hair is twisted at one end point causes tangling. (c) a hair is pulled across another to exhibit stickiness. (d) a curly hair is stretched and released.*

4 Time Integration

Given our hair model, time integration from time t^n to time t^{n+1} proceeds using a variant of [Sifakis et al. 2007] as follows:

1. $\boldsymbol{v}^{n+1/2} = \boldsymbol{v}^n + \frac{\Delta t}{2}\boldsymbol{a}(t^{n+1/2}, \boldsymbol{x}^n, \boldsymbol{v}^{n+1/2})$
2. Modify $\boldsymbol{v}^{n+1/2}$ with strain limiting
3. Modify $\boldsymbol{v}^{n+1/2}$ with self-repulsions
4. $\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t \boldsymbol{v}^{n+1/2}$
5. Body collisions modify $\boldsymbol{x}^{n+1}$ and $\boldsymbol{v}^n$
6. Self-collisions modify $\boldsymbol{x}^{n+1}$ and $\boldsymbol{v}^n$
7. $\boldsymbol{v}^{n+1/2} = \boldsymbol{v}^n + \frac{\Delta t}{2}\boldsymbol{a}(t^{n+1/2}, \boldsymbol{x}^{n+1/2}, \boldsymbol{v}^{n+1/2})$
8. Extrapolate $\boldsymbol{v}^{n+1} = 2\boldsymbol{v}^{n+1/2} - \boldsymbol{v}^n$
9. Modify $\boldsymbol{v}^{n+1}$ with self-repulsions

where $x^{n+1/2} = (x^n + x^{n+1})/2$. The velocity computed in step 1 is processed with our new strain limiting approach (Section 4.2) as well as self-repulsions before being used to update the positions in step 4. After the position update, body collisions are applied first in

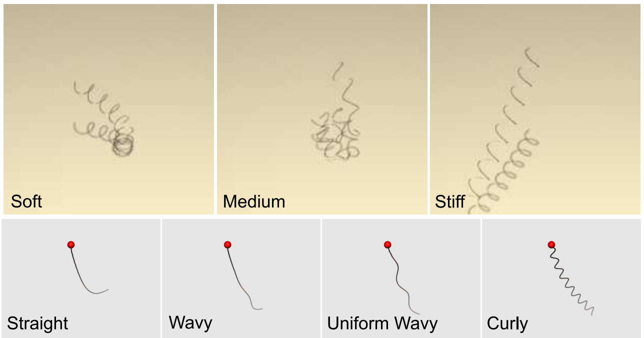


Figure 11: *(Top) A single curly hair dropped with varying stiffnesses showing that we can represent different types of curly hair. (Bottom) Varying degrees of waviness showing that our model can handle the full spectrum of hair curliness.*

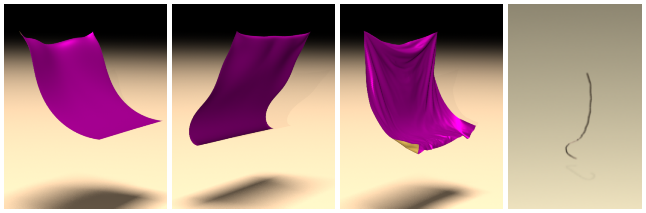


Figure 12: *Both cloth (left two figures) and hair (far right) integrated using one time step per frame. However, excessively large time steps in implicit methods lead to larger damping. The third figure from the left depicts the exact same cloth simulation with a smaller time step showing that the larger damping stems from the time step size, not the method.*

step 5 and then self-collisions in step 6 (both of which are discussed in Section 5). The velocity from steps 1-4 is discarded, and step 7 and 8 evolve the velocity forward in time using the trapezoidal rule, though increased stability can be obtained with backward Euler.

4.1 Implicit Linear Springs

Integrating stiff springs using explicit or even semi-implicit (implicitly damped) time integration can be intractable due to stringent time step restrictions. To increase the efficiency of our simulations we could use a fully implicit spring model, but this usually requires Newton-Raphson iteration ([Baraff and Witkin 1998] used one iteration) as well as techniques to symmetrize the forces. Instead, we propose a new discretization of linear springs with elastic forces that are truly linear in position.

The elastic force at time t^{n+1} of a spring between two points $\boldsymbol{x}_1^{n+1}$, $\boldsymbol{x}_2^{n+1}$ is

$$\boldsymbol{F}^{n+1} = k\left(\|\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1}\|/l_0 - 1\right)\hat{\boldsymbol{d}}^{n+1}$$

where $\hat{\boldsymbol{d}}^{n+1} = (\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1})/\|\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1}\|$ is the spring direction. If we rewrite the force as

$$\boldsymbol{F}^{n+1} = \frac{k}{l_0}\left((\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1})^\mathsf{T}\hat{\boldsymbol{d}}^{n+1} - l_0\right)\hat{\boldsymbol{d}}^{n+1}$$

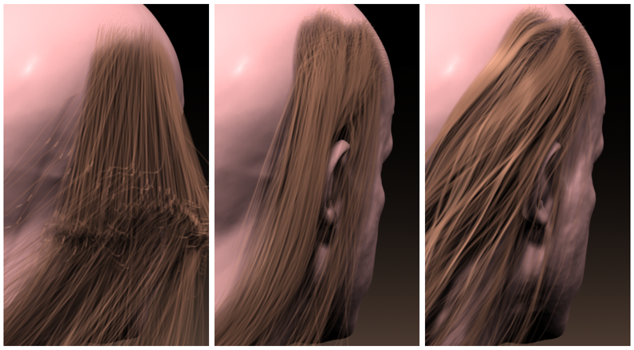
it is clear that fixing the spring direction at time t^n makes the force linear in $\boldsymbol{x}^{n+1}$, i.e.

$$\boldsymbol{F}^{n+1} = \frac{k}{l_0}\left((\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1})^\mathsf{T}\hat{\boldsymbol{d}}^{n} - l_0\right)\hat{\boldsymbol{d}}^{n}.$$

Substituting $\boldsymbol{x_i}^{n+1} = \boldsymbol{x_i}^n + \Delta t \boldsymbol{v_i}^{n+1}$ yields

$$\boldsymbol{F}^{n+1} = \frac{k}{l_0}\left((\boldsymbol{x}_2^{n} - \boldsymbol{x}_1^{n})^\mathsf{T}\hat{\boldsymbol{d}}^{n} - l_0\right)\hat{\boldsymbol{d}}^{n} + \Delta t \frac{k}{l_0}(\boldsymbol{v_2}^{n+1} - \boldsymbol{v_1}^{n+1})^\mathsf{T}\hat{\boldsymbol{d}}^{n}\hat{\boldsymbol{d}}^{n}.$$

The first term is the typical explicitly integrated elastic force, and the second term is the typical damping force from a semi-implicit method except that the damping coefficient is fixed at $\Delta tk/l_0$. Notably, this indicates that if one adds the exact correct amount of damping to the semi-implicit discretization, then the spring remains stable. In summary, springs with fixed directions have elastic forces that are linear in the position and only require a single implicit linear solve for the damping forces to be unconditionally stable. Of course the usual damping can still be added making the damping coefficient $\Delta tk/l_0+b$, where b is the usual damping coefficient. We note that while there are other approaches to deriving linear springs (see e.g. [Desbrun et al. 1999; Etzmuß 2002]), our particular one has the advantage that the linear term projects only in the spring direction, giving less damped results. Finally, note that these springs are readily included in the Newmark semi-implicit time integration just as any other force which is linear in the positions (similar to the zero restlength binding springs in [Sifakis et al. 2007]). Figure 12 shows cloth and hair integrated at the frame rate using these springs.



(a) Basic Mass/Spring (b) Our model (No self-interactions) (c) Our Full Model

Figure 13: *A tuft of hair simulated with (a) a basic mass spring model with no torsion springs or self-interaction, (b) our mass-spring model including altitude springs, implicit springs and strain limiting, and (c) our mass-spring model with self-adhesion and self-collisions. Note in (a) the long straight hairs that extend from the scalp to the neck actually represent one spring which is severely stretched. Also note how much more realistic (c) looks with the addition of self-collisions and stiction.*

4.2 Strain Limiting

Complex head motions can cause severe stretching especially in springs which have one of their two endpoints constrained to a character's head as seen in Figure 13(a). To alleviate high strain in cloth, [Provot 1995; Bridson et al. 2002] used strain limiting approaches that apply momentum conserving velocity impulses to particles attached by springs that exceed 10% deformation. Since correcting one spring potentially damages another, iteration is typically used. Since hair is relatively light compared to the head, we employ a biased strain limiting approach that marches from the root of the hair and projects the length by moving only the particle further from the root in the direction formed by the two particles. For the extra particles from the subdivision and perturbation in Section 3.2, we also project one of their two edge springs (the one closer to the root). Iteratively, once we find a new candidate position, we apply a velocity impulse to the $\boldsymbol{v}^{n+1/2}$ velocity to achieve that position. Note that the strain limiting in step 2 above only affects the velocity used to update the positions and has no direct effect on the velocity used for evolution in steps 5-9. One might alternatively suggest using an extremely stiff spring with a fully implicit integrator, however, implicitly integrated stiff springs cause an implicit solve to converge slower. One could also use a method similar to [Goldenthal et al. 2007], which would also likely be more expensive than strain limiting.

5 Interaction and Collisions

5.1 Body Collisions

We represent object collision geometry as a level set signed distance function which is negative inside and positive outside. Collisions (and friction) proceed as in [Bridson et al. 2003] with any intersecting point having its relative normal velocities constrained to prevent collision during the second backward Euler solve in step

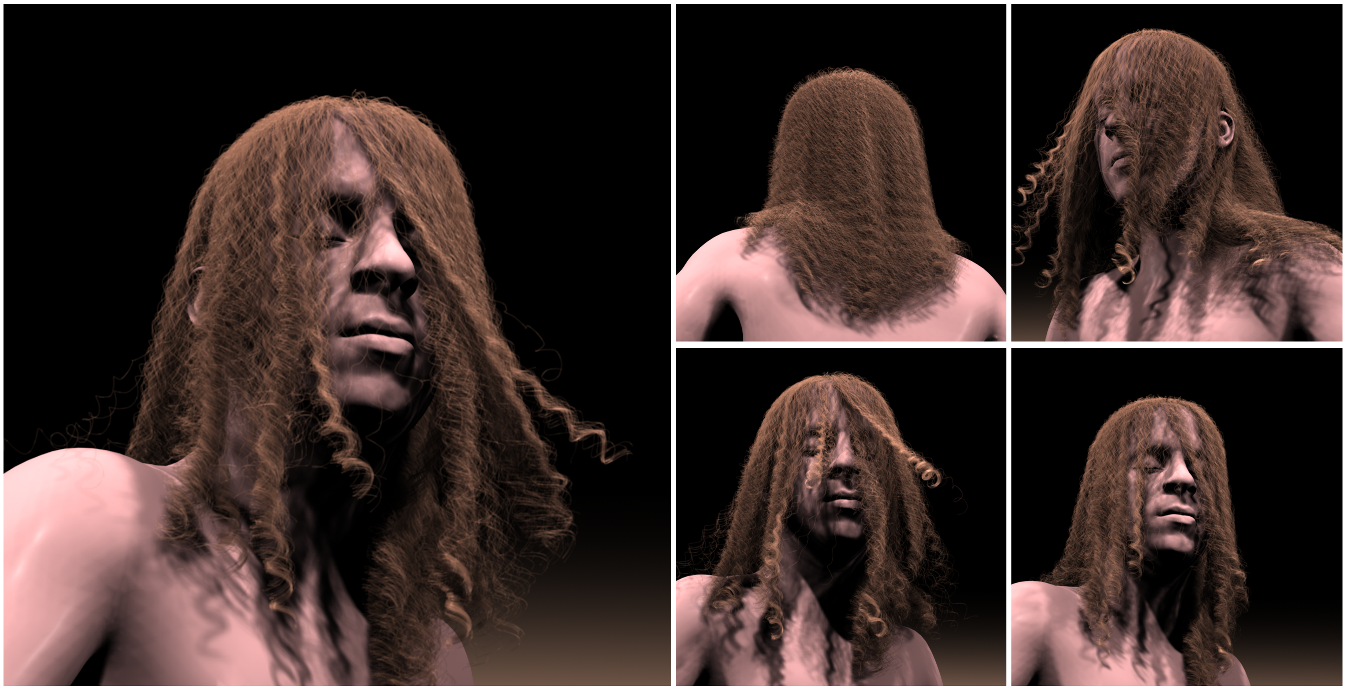
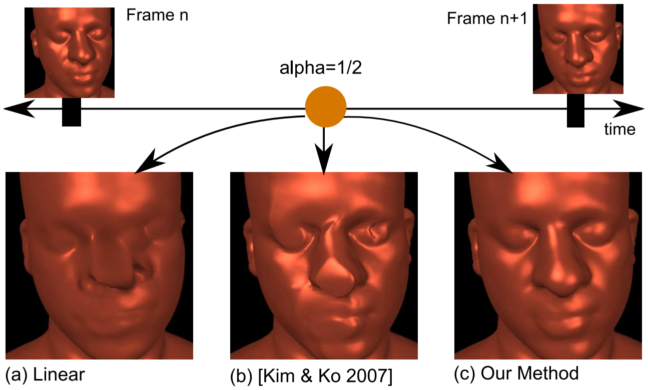


Figure 14: *A simulation of 5,000 long curly hairs with 50 segments each (250,000 total particles) on a character spinning around from back to front.*

7. As collisions can flatten the structure of hair, flattening curls, a fold preservation approach similar to theirs might be useful. As the head and torso are undergoing complex motion we create level sets at 60 frames per second (the frequency of our motion capture data). To represent level sets at an arbitrary time t, one could naïvely linear interpolate from the closest two frames. Instead we employ a semi-Lagrangian advection based interpolation scheme similar to [Kim and Ko 2007]. Using their scheme directly produces a discontinuous interpolation (not a problem for motion blur) as they only evolve *forward* from a time t^n. Instead we use the nearest two frames, interpolating forward and backward evolution via $\phi(\boldsymbol{x},\alpha) = (1-\alpha)\phi^n(\boldsymbol{x}-\Delta t\alpha\overline{\boldsymbol{v}}) + \alpha\phi^{n+1}(\boldsymbol{x}+\Delta t(1-\alpha)\overline{\boldsymbol{v}})$ where $\overline{\boldsymbol{v}} = (\boldsymbol{v}^n(\boldsymbol{x}) + \boldsymbol{v}^{n+1}(\boldsymbol{x}))/2$ with the collision body velocities $\boldsymbol{v}^n$ and $\boldsymbol{v}^{n+1}$, and $\alpha \in [0,1]$ is the interpolation fraction. See Figure 15.



(a) Linear
(b) [Kim & Ko 2007]
(c) Our Method

Figure 15: *Interpolation midway between two key frames is shown using three methods: (a) naïve linear interpolation, (b) Kim and Ko's one sided Eulerian evolution and (c) our new two-sided evolution interpolation.*

5.2 Stiction

Hair/hair interaction effects include friction and static charge adhesion which cause hair to stick together. In fact, the tendency of hair to merge together is one of the motivations for wisp and clump models. Springs can be useful for making and breaking dynamic constraints as in [Jimenez and Luciani 1993]. Similar techniques were then used in cloth [Bridson et al. 2003] and articulated rigid bodies [Criswell et al. 2006]. These techniques also were applied to hair in [Chang et al. 2002] (though they do not dynamically create new connections) and [Ward et al. 2004a; Ward et al. 2004b] for wet and styled hair (though connections were created only in response to interactive styling product application).

We created a stiction model in the same spirit as the approaches listed above. First, we create segment pairs when the shortest vector between two edges is less than a distance threshold ϵ_s. Then we store the interpolation fraction along each segment (thus defining two embedded nodes) and create a spring between them that has restlength ϵ_s (making a segment pair resist not only being pulled apart but also being compressed together). These springs are maintained with the same embedded node locations until the length of the spring exceeds a separation threshold. For efficiency, we search for pairs of segments using a bounding box hierarchy containing each segment, and we only allow n connections to be formed to a segment by using a max heap to keep the n closest segments to any given segment.

Connections may be broken as hairs separate beyond the distance threshold. [Ward et al. 2004a] broke connections after a force threshold was exceeded, but we avoid this, as it could improperly break a connection when two hairs are forced together. Other authors typically reduce the spring stiffness with distance to model different hairs separating at various times. We do not do this because we observed that real hair tends to exhibit discontinuous separation behavior (see Figure 10(c)) which is explained by the concentric scales that comprise a hair (see e.g. [Robbins 1994]). This is an example of one of the departures we make in modeling individual hairs as opposed to clumped or continuum models.

Adjusting stiction parameters allows a user to model various hair behaviors. For example, extra spring connections for stickiness along with a heavier mass could be used to model wet hair (see references above). Hair can also be made more or less clumped by varying the number of connections and the stiffness. Additionally, hair volume can be increased by using larger restlengths and stiffnesses. Doing so would retain a similar time step, as stiffer springs reduce the time step while longer restlengths increase it. Moreover, increasing the volume would cause hairs to be further apart, reducing the number of collision pairs—making the simulation easier. This is akin to the effect that repulsions have in the [Bridson et al. 2002] algorithm, where they reduce the need for more expensive geometric collisions. However, one potential hazard of increasing the restlength is that springs become more directionally biased which could affect rotational stability, but this might be remedied by using a tetrahedron based stiction force akin to our constitutive model.

5.3 Self-Repulsions and Collisions

For self-repulsions and self-collisions we adapt the techniques from [Bridson et al. 2002] using only the edge/edge repulsion and collision techniques. Whereas cloth interpenetration produces an obviously invalid state, any hair state is collision-free, and self-collisions are only apparent during motion. Thus added computational efficiency can be achieved by ignoring some collisions; in fact we do not use rigid impact groups [Provot 1995]. In that vein, we found that the stiction approach described above greatly reduces the number of edge/edge collisions but unfortunately also has the effect of bringing edges closer together causing many false positives to be detected. Thus for efficiency and robustness, we ignore collision pairs that have a stiction connection. A comparison between Figures 13(b) and 13(c) illustrates the improved realism due to stiction, self-repulsions and self-collisions.

6 Examples

We have produced a number of examples ranging from a single hair, a tuft of hair, and a full head of hair. We employ the standard [Kajiya and Kay 1989] anisotropic reflectance model and use deep shadow maps [Lokovic and Veach 2000]. One could also use the more accurate reflectance and scattering models of [Marschner et al. 2003; Moon and Marschner 2006]. We stress that we do not create any additional hair geometry at render-time, only rendering what we simulated.

We built a head and torso geometry by combining a laser scan of a head with the torso from the Visible Human Data Set [U.S. National Library of Medicine 1994]. We captured realistic and highly dynamic motion using an optical motion capture system. The resulting motion capture skeleton was used as a boundary condition for a quasistatic flesh simulation [Teran et al. 2005], and the triangulated mesh was deformed by barycentrically embedding each vertex. Each frame of the surface was then rasterized to create a level set for head and torso collisions. To grow hairs, we painted hair densities and seeded points using [Turk 1992]. The first few points at the hair root were embedded in the head model barycentrically to obtain a full coordinate constraint.

Hair Type	Particle Count	Avg m/ frame	% Newmark	% Collide	% Stiction
Curly	250k	4 m	44 %	10 %	20 %
Straight short	500k	10 m	70 %	3 %	13 %
Straight short wind	500k	12 m	70 %	9 %	13 %
Straight long (5k)	500k	17 m	64 %	16 %	14 %
Straight long (10k)	1,000k	38 m	64 %	3 %	21 %

Table 2: *Performance of simulating various hair types. We provide time/frame as well as a percent breakdown between time integration, collisions and stiction.*

We simulate full heads of hair in three different styles: long straight (Figure 9), medium straight (Figure 16) and long curly (Figure 14). In addition to body and head motion we also show hair affected by external forces such as wind in Figure 2. These examples illustrate clumping behavior as well as stray individual hairs giving a multiresolution feel that is difficult to capture with clumped or continuum models. The typical time steps were 6×10^{-4} s, although we chose our time step adaptively based on the strain rate and Courant condition.

We ran our examples with one to four quad processor Opteron machines in as little as a few minutes per frame to as much as an hour per frame, when highly dynamic motion capture data strained the system. Large examples (shown in Table 2) used four machines and smaller examples used one machine. Parallelization was carried out in a straightforward manner discussed in [Selle et al. 2007]. Simulation time (shown in Table 2) varies based on the velocities of the underlying head animations and wind forces even though the number of segments and time step is the same (required CG iterations may change). Note the “straight long” example where we ran 5k and 10k versions shows that doubling the number of elements approximately doubled the time which we found encouraging as conjugate gradient has larger than linear complexity. Comparing these numbers to other published results is not especially meaningful because of different hardware and levels of optimization. Also, since our examples have many more elements than others’ examples, we typically do not fit into either L1/L2 cache so a linear speedup of other methods should not be expected. Furthermore, we used high velocity motions which will make convergence slower than lower velocity examples.

7 Limitations

We attempted to simulate as many hairs on the head as possible and in doing so we generated many interesting, highly-detailed examples with up to a million particles. Unfortunately, we did not manage to simulate 100k hairs. Architectural improvements will naturally remedy this problem to some extent, but developing more efficient time evolution and collision approaches is also important. Thus an obvious limitation is that simulating more elements is more expensive than sparse clumped models making our method less suited for simulating hair in real time and less cost effective when a high level of detail is not required or necessary.

Using the Bridson approach for self-repulsions and collisions is a limitation as it is primarily designed for interactions of edges embedded in a surface. Although it can be used for hair collision, it is not an efficient way of doing so as it has trouble with very complex stacking configurations that sometimes occur in hair. In our model, we used stiction instead of collisions for some pairs to help with this. This is analogous to Bridson’s use of rigid groups when the correct collision responses could not be determined. Improving collision response techniques in the presence of extreme stacking would allow us to apply collision restitution to all pairs. One promising approach would be to use a tetrahedral mesh together with altitude spring based techniques. This approach would also help maintain hair shape (and volume) by preventing pointwise rotations during collision response. Another issue is that wrinkles in hair tend to be flattened by body collisions which could possibly be remedied by adapting the cloth fold preservation technique of [Bridson et al. 2003].

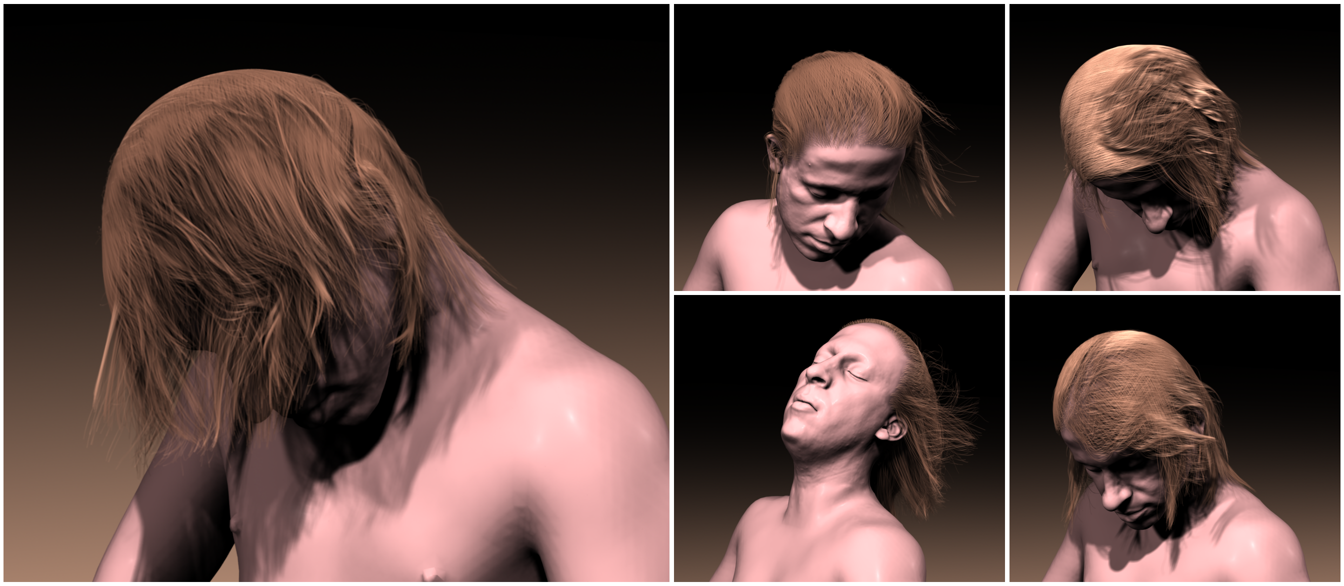


Figure 16: *A simulation of 10,000 medium length straight hairs with 25 segments each (500,000 total particles) on a character moving his head in a circular motion.*

Our spring-based torsion model also contains some parasitic bending influence which is analogous to some of the problems with linear bending spring models. Research had led to the development of better non-linear bending models, but recently bending approaches have come full circle by focusing on more efficient linear bending models. Nevertheless, improved mass-spring torsion models would be an interesting avenue of research.

8 Discussion

Our method departs from recent hair simulation methods in several ways. First, we directly simulate every hair we render so we can capture all the DOFs needed for high fidelity hair. We use a mass-spring model that also can capture torsion which allows us to use standard time integration and interaction schemes. In order to capture electrostatic attraction, we employ a per hair stiction and friction model whereas clumped models assume (and fix) the intra-clump interaction of hairs. Inter-clump interaction is modeled, but only in the aggregate sense, limiting discontinuous interaction behavior within clumps. Lastly, hair dynamics on aggregate models use clumped mass instead of more physical small mass hairs (which we use).

Our method represents one extreme of the performance/quality tradeoff existing between simulating individual hairs and simulating clumps or continuums. A practitioner that desires to simulate hair chooses a technique in this spectrum to satisfy his needs be them performance or fidelity. For example, in a film, a hero shot that needs very detailed single hair behavior and interactions would use our technique, whereas this is overkill for a character in the background. One might instead wish to use less computationally expensive methods such as super helices because it can represent interesting curly shapes with very few DOFs. In a film that requires stylized hair, where uniformity is desired, a volumetric/continuum method such as [Petrovic et al. 2005] would be ideal. In extremely time-limited scenarios such as a video game or simulator, an approach such as [Bando et al. 2003] might be applied.

An interesting avenue of research is considering combinations of techniques so that different levels of detail can be achieved in the same simulation. First, the number of hairs could be varied in our method which among other things would allow our method to be used in a clumped fashion to reduce computational cost. Thus, all the techniques for interpolating rendering hairs would be useful. One could also augment our interaction models with continuum approaches such as smooth particle hydrodynamics [Hadap and Magnenat-Thalmann 2001], etc. Another promising approach would be to use faster simulation techniques to iterate a basic look for a character. Then, our method could be influenced by the guide hairs to produce a simulation with the art direction and control of the coarser simulation and the intricate details of a large scale simulation.

In order to simulate 100,000 hairs, the bottlenecks at that particular resolution must be considered. The stiffness matrix and unknowns for time integration increase linearly in the number of hairs. Collision detection can theoretically scale poorly, but in our long straight example going from 5,000 to 10,000 hairs, we found that the relative percent of time on collisions (detection and response) went down, and in all of the examples in Table 2, they were below 16% of total simulation time. Even though we skipped collision *response* for stiction pairs, we did *detect all* pairs (including stiction pairs) using an axis-aligned bounding box hierarchy. As we scale to 100,000 hairs it is likely that although detection will not be a problem, more complex stacking configurations will necessitate better collision response. However, we note that detection might be sped by employing the acceleration techniques used in [Ward et al. 2007a].

In conclusion, We have presented a novel mass-spring hair model with the goal of simulating every individual hair on the head. Even though mass-spring models have recently been unpopular, we have shown that they can in fact model torsion to simulate hair effectively. However, the most immediate challenge is scaling the model to more hairs. Figure 17 shows a simulation with 5,000 hairs compared to one with 10,000 hairs showing that double the number of hairs has a significant impact on the simulation quality justifying our goal of simulating 100,000 hairs. We believe we have taken important steps towards obtaining higher fidelity hair simulation, and we look forward to future work.

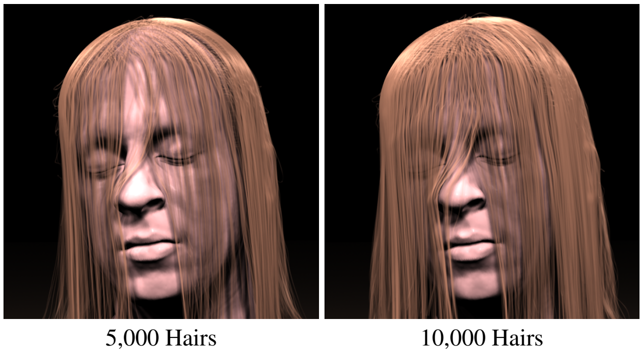


Figure 17: *A comparison of a simulation with 5,000 hairs versus one with 10,000 hairs. Notice the significant improvement when doubling the number of hairs, and that in this pose we would benefit from even more hairs.*

Acknowledgements

Research supported in part by a Packard Foundation Fellowship, an Okawa Foundation Research Grant, ONR N0014-06-1-0393, ONR N00014-02-1-0720, ONR N00014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF IIS-0326388, NSF ITR-0205671 and NSF CCF-0541148. We gratefully acknowledge Joyce Sheenye Pan and Justin Solomon for their efforts, Tamar Shinar for many helpful discussions, Joseph Teran for his sculpting skills and Pixar Animation Studios for providing RenderMan licenses. We would also like to thank Eftychios Sifakis for the "I am Legend" version of his likeness.

References

ANJYO, K., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. In *Comp. Graph. (Proc. ACM SIGGRAPH 92)*, ACM, vol. 26, 111–120.

BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2003. Animating hair with loosely connected particles. In *Comp. Graph. Forum (Eurographics Proc.)*, 411–418.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *ACM SIGGRAPH 98*, ACM Press/ACM SIGGRAPH, 43–54.

BERTAILS, F., KIM, T.-Y., CANI, M.-P., AND NEUMANN, U. 2003. Adaptive wisp tree - a multiresolution control structure for simulating dynamics clustering in hair motion. *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 207–213.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph. 25*, 3, 1180–1187.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. 21*, 3, 594–603.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 28–36.

BROWN, J., LATOMBE, J.-C., AND MONTGOMERY, K. 2004. Real-time knot-tying simulation. *Vis. Comput. 20*, 2, 165–179.

CHANG, J. T., JIN, J., AND YU, Y. 2002. A practical model for hair mutual interactions. In *Proc. ACM SIGGRAPH Symp. on Comput. Anim.*, 77–80.

CHOE, B., AND KO, H.-S. 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Trans. on Vis. and Comput. Graph. 11*, 2, 160–170.

CHOE, B., CHOI, M., AND KO, H.-S. 2005. Simulating complex hair with robust collision handling. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 153–160.

CRISWELL, B., DERLICH, K., AND HATCH, D. 2006. Davy jones' beard: rigid tentacle simulation. In *SIGGRAPH 2006 Sketches*, 117.

DESBRUN, M., SCHRÖDER, P., AND BARR, A. 1999. Interactive animation of structured deformable objects. In *Graph. Interface*, 1–8.

ETZMUSS, O. 2002. *Animation of Surfaces with Applications to Cloth Modelling*. PhD thesis, Tübingen.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3, 49.

GRÉGOIRE, M., AND SCHÖMER, E. 2006. Interactive simulation of one-dimensional flexible parts. In *Symp. on Solid and Physical Modeling*, 95–103.

GUPTA, R., MONTAGNOO, M., VOLINO, P., AND MAGNENAT-THALMANN, N. 2006. Optimized framework for real time hair simulation. In *CGI Proc. 2006*, 702–710.

HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. In *Comp. Graph. Forum (Eurographics Proc.)*, 329–338.

HADAP, S. 2006. Oriented strands: dynamics of stiff multi-body system. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 91–100.

IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 131–140.

JIMENEZ, S., AND LUCIANI, A. 1993. Animation of interacting objects with collisions and prolonged contacts. In *Modeling in computer graphics—methods and applications*, Springer-Verlag, B. Falcidieno and T. L. Kunii, Eds., Proc. of the IFIP WG 5.10 Working Conf., 129–141.

KAJIYA, J. T., AND KAY, T. L. 1989. Rendering fur with three dimensional textures. In *Comp. Graph. (Proc. ACM SIGGRAPH 90)*, ACM, 271–280.

KIM, D., AND KO, H.-S. 2007. Eulerian motion blur. In *Eurographics Workshop on Natural Phenomena 2007*, 39–46.

KIM, T.-Y., AND NEUMANN, U. 2002. Interactive multiresolution hair modeling and editing. *ACM Trans. Graph. 21*, 3, 620–629.

LOKOVIC, T., AND VEACH, E. 2000. Deep shadow maps. In *ACM SIGGRAPH 2000*, ACM Press/ACM SIGGRAPH, 385–392.

MARSCHNER, S. R., JENSEN, H. W., CAMMARANO, M., WORLEY, S., AND HANRAHAN, P. 2003. Light scattering from human hair fibers. *ACM Trans. Graph. 22*, 3, 780–791.

MOLINO, N., BRIDSON, R., TERAN, J., AND FEDKIW, R. 2003. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *12th Int. Meshing Roundtable*, 103–114.

MOON, J. T., AND MARSCHNER, S. R. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph. 25*, 3, 780–791.

PAI, D. K. 2002. Strands: Interactive simulation of thin solids using cosserat models. In *Proc. of Eurographics*, vol. 21 of *Comput. Graph. Forum*, Eurographics Assoc., 347–352.

PETROVIC, L., HENNE, M., AND ANDERSON, J. 2005. Volumetric methods for simulation and rendering of hair. Tech. Rep. 06-08, Pixar.

PLANTE, E., CANI, M.-P., AND POULIN, P. 2002. Capturing the complexity of hair motion. *Graph. Models 64*, 1 (january), 40–58.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graph. Interface*, 147–154.

ROBBINS, C. R. 1994. *Chemical and physical behavior of human hair*. Springer-Verlag, New York.

ROSENBLUM, R. E., CARLSON, W. E., AND TRIPP III, E. 1991. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *J. Vis. and Comput. Anim. 2*, 4, 141–148.

SELLE, A., SU, J., IRVING, G., AND FEDKIW, R. 2007. Highly detailed folds and wrinkles for cloth simulation. *IEEE Trans. on Vis. and Comput. Graph. (In Press)*.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 81–90.

SPILLMANN, J., AND TESCHNER, M. 2007. CoRDE: cosserat rod elements for the dynamic simulation of one-dimensional elastic object. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 209–217.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 181–190.

TURK, G. 1992. Re-tiling polygonal surfaces. In *Comput. Graph. (Proc. ACM SIGGRAPH 92)*, ACM, 55–64.

U.S. NATIONAL LIBRARY OF MEDICINE, 1994. The visible human project. http://www.nlm.nih.gov/research/visible/.

WANG, F., BURDET, E., DHANIK, A., POSTON, T., AND TEO, C. 2005. Dynamic thread for real-time knot-tying. *Eurohaptics Conf., 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, 507–508.

WARD, K., AND LIN, M. C. 2003. Adaptive grouping and subdivision for simulating hair dynamics. In *Pacific Graph.*, 234.

WARD, K., LIN, M. C., LEE, J., FISHER, S., AND MACRI, D. 2003. Modeling hair using level-of-detail representations. In *Proc. of Comput. Anim. and Social Agents (CASA)*, 41.

WARD, K., GALOPPO, N., AND LIN, M. C. 2004. Modeling hair influenced by water and styling products. In *Proc. of Comput. Anim. and Social Agents (CASA)*, 207–214.

WARD, K., GALOPPO, N., AND LIN, M. C. 2004. Simulating and rendering wet hair. In *SIGGRAPH 2004 Sketches*, ACM Press, 42.

WARD, K., GALOPPO, N., AND LIN, M. 2006. A simulation-based vr system for interactive hairstyling. *Virt. Reality Conf., 2006*, 257–260.

WARD, K., GALOPPO, N., AND LIN, M. 2007. Interactive virtual hair salon. *Presence: Teleoper. Virt. Environ. 16*, 3, 237–251.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation and rendering. *IEEE Trans. on Vis. and Comput. Graph. 13*, 2, 213–234.

YU, Y. 2001. Modeling realistic virtual hairstyles. In *Pacific Graph.*, 295–304.

Appendix: Proof of altitude spring existence

We first show that for a given tetrahedron with at least one positive area triangle, there always exists a point/face or edge/edge pair that has non-negative weights. Consider every non-zero area triangle $\triangle ABC$ and project the fourth point D into its plane as $P(D)$. Then there are three cases: (1) $P(D)$ is inside $\triangle ABC$, meaning the point/face altitude spring has non-negative weights. (2) $P(D)$ is outside $\triangle ABC$ and the convex hull consists of $\triangle P(D)AB$ (relabelling without loss of generality). Then the projection of $\triangle P(D)AB$ is inside $\triangle DAB$ so $P(C)$ is inside $\triangle DAB$ making that a point/face altitude with non-negative weights. (3) $P(D)$ is outside $\triangle ABC$ and the convex hull is a quadrilateral so that segments intersect yielding a edge/edge pair with non-negative weights.

Next, the minimum altitude has non-negative weights because any invalid edge/edge or point/face pair is not the minimum altitude. Suppose an invalid point/face pair, then if the points projection on the face yields a quadrilateral, then a smaller distance must exist between the resulting edge/edge pair because the projected point is the highest distance from the plane by definition and is incident on one of the crossing edges. A similar argument occurs if the point face is invalid with the convex hull being a triangle. In the case where the edge/edge pair is invalid, then if it is coplanar, it is tied in altitude with everything. Otherwise, the length can be reduced by choosing a different pair.