

Detail Preserving Continuum Simulation of Straight Hair

Aleka McAdams[†] Andrew Selle^{*} Kelly Ward^{*} Eftychios Sifakis^{†*} Joseph Teran^{†*}

[†]University of California, Los Angeles

^{*}Walt Disney Animation Studios

Abstract

Hair simulation remains one of the most challenging aspects of creating virtual characters. Most research focuses on handling the massive geometric complexity of hundreds of thousands of interacting hairs. This is accomplished either by using brute force simulation or by reducing degrees of freedom with guide hairs. This paper presents a hybrid Eulerian/Lagrangian approach to handling both self and body collisions with hair efficiently while still maintaining detail. Bulk interactions and hair volume preservation is handled efficiently and effectively with a FLIP based fluid solver while intricate hair-hair interaction is handled with Lagrangian self-collisions. Thus the method has the efficiency of continuum/guide based hair models with the high detail of Lagrangian self-collision approaches.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: hair simulation, continuum models

1 Introduction

Simulating hair on a virtual character remains one of the most challenging aspects of computer graphics. As hair is an integral part of creating many virtual characters, the problem is especially important. Unfortunately, the massive number of hairs interacting and colliding makes this task especially challenging. Many approximations for simulating hair exist, but they typically fail to provide the amount of detail that real hair exhibits. Several applications, such as feature films, aim to capture the high degree of complexity caused by several thousand interacting hair strands.

Even though individual hair dynamics scale well to multiple hairs (as each hair is dynamically decoupled), accurately simulating many hairs interacting with each other remains a challenge. Instead, numerous approaches have been developed to manage the complexity of many hairs interacting by simulating a smaller set of guide hairs (typically no more than several hundred) that interact with large repulsion forces, interpolating a larger number of hairs for rendering. This leads to very efficient simulation times, but a limited amount of hair detail is captured (ignoring stray hairs such as the so-called “flyaways”) because, essentially, each guide hair represents hundreds (or even thousands) of actual hairs.

Alternatively, there have been several methods that treat every simulated hair as part of a fluid-like continuum volume. These ap-



Figure 1: (Left) Real hair exhibiting intricate webbing as result of complex collision/contact. (Right) Our method creating a similar effect.

proaches naturally model hair interaction without explicit collisions. However, the downside is that intricate features of individual hairs are lost because each hair is enslaved to the continuum.

Our technique factors hair computation into two parts: a coarse, highly coupled volumetric behavior, which is efficiently modeled by a continuum; and a finer, more locally coupled Lagrangian particle simulation of hair. Unlike previous continuum-based approaches that only simulate guide hairs that do not interact directly, our method simulates many hairs (several thousand) that are allowed to collide directly (as well as through the volume). We handle self-collisions more efficiently than fully Lagrangian collision models because the volume does most of the work towards resolving collisions. Thus, our method can capture the intricate details of many individual strands while efficiently maintaining the overall hair volume.

2 Related Work

Hair modeling is an active area of research that can be divided into the following categories: hair shape modeling, hair strand dynamics, hair complexity management and interaction, hair shading and self-shadowing. Static hair shape modeling, such as [Yu 2001; Kim and Neumann 2002; Choe and Ko 2005], hair shading and shadow computation are beyond the scope of our paper, so we do not discuss them further. A more detailed discussion on these topics can be found in the recent survey paper [Ward et al. 2007].

Strand Dynamics. Many hair simulation techniques consider different geometric and constitutive models for modeling individual hair strands or clumps of hair starting with mass spring systems [Rosenblum et al. 1991] and projective dynamics [Anjyo et al. 1992]. Mass/spring simulated lattice deformers can be added to model torsion [Plante et al. 2002; Selle et al. 2008]. Rigid body chaining methods are also commonly used [Brown et al. 2004; Choe et al. 2005; Hadap 2006]. Techniques based on elastic rod theory have become popular starting with [Pai 2002] and continuing with [Grégoire and Schömer 2006; Bertails et al. 2006]. [Spillmann and Teschner 2007] also uses elastic rod theory but adaptively changes the number of points that define the hair curve. Most recently, [Bergou et al. 2008] provides a novel modeling of twist as deviation from a canonical frame as well as a method to evolve this separately quasistatically. Work in strand dynamics has been extensive, and our technique can be used with the practitioner’s model of choice, but for simplicity we use the mass/spring model.

ACM Reference Format

McAdams, A., Selle, A., Ward, K., Sifakis, E., Teran, J. 2009. Detail Preserving Continuum Simulation of Straight Hair. *ACM Trans. Graph.* 28, 3, Article 62 (August 2009), 6 pages. DOI = 10.1145/1531326.1531368. <http://doi.acm.org/10.1145/1531326.1531368>.

Copyright Notice

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, fax +1 (212) 869-0481, or permissions@acm.org.

© 2009 ACM 0730-0301/09/03-ART62 \$10.00 DOI 10.1145/1531326.1531368. <http://doi.acm.org/10.1145/1531326.1531368>

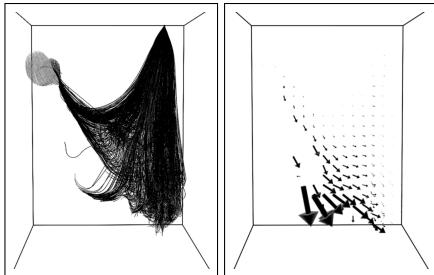


Figure 2: Our method models the hair both as curves (left) and as a volumetric velocity field (right).

Hair Interaction and Complexity. The simplest (but most expensive) approach to hair complexity management is simulating and rendering every hair [Rosenblum et al. 1991; Anjyo et al. 1992; Selle et al. 2008]. While this theoretically yields the most detail, it is the most intractable, especially if self-collisions are considered. Thus most practitioners use a clumping or continuum level of detail scheme that trades accuracy for computational tractability:

Continuum models. [Hadap and Magnenat-Thalmann 2001] uses a Lagrangian fluid (smoothed particle hydrodynamics) technique to model fluid forces on a mass/spring system. While this efficiently handles bulk behavior, detailed hair behavior is lost. In addition, since smoothed particle hydrodynamics (SPH) is used, forcing fluid interaction through spring like forces, this approach is effectively as numerically difficult as standard self-repulsion spring forces. [Bando et al. 2003] also uses a continuum approach while discarding particle connectivity for real-time applications. Unfortunately, this results in visual artifacts in renderings because of the lack of connectivity in addition to the continuum’s loss of high fidelity detail. [Petrovic et al. 2005] presented a continuum approach that rasterized hair to an Eulerian velocity field and level set with the goal of reducing computational complexity, rather than improving on fidelity. Dynamics on the grid was limited to repeated averaging (an approximation of viscosity) to obtain friction effects. Unlike our method, they did not consider the incompressibility or other fluid behaviors that could be imparted on the velocity field. In addition, the goal of the work was to model relatively simple hairstyles on stylized characters, so the detail lost by using a coarse grid was acceptable for their application.

Clumped models. There are many techniques that simulate hair interactions through a sparse set of disjoint guides that collide with one another [Bertails et al. 2006; Hadap 2006; Gupta et al. 2006; Chang et al. 2002]. A greater set of rendered hair strands are then either interpolated between these guides or clumped to the guide strands. [Plante et al. 2002] used a lattice whose vertices were simulated with a mass/spring model to define many more interpolated hairs. [Choe et al. 2005] synthesized additional hairs from a sparse set of simulated guide hairs with a statistical model. [Bertails et al. 2006] used a combination of interpolation and clumping to generate rendered strands. Many clumped techniques also use adaptivity to merge or split groups of hair to reduce computation where and when the required detail is low [Bertails et al. 2003; Ward et al. 2003; Ward and Lin 2003]. The use of sparse clumps often allows the use of cheap repulsion penalties as opposed to geometric collisions because guides are expected to have thickness, and in Section 4.2 we will discuss how this breaks down with thousands of guides.

3 Method Overview

Our approach combines a Lagrangian hair solver with an Eulerian fluid simulator (see Figure 2). We use a mass/spring model for the hair particles, connecting each particle in sequence with a spring

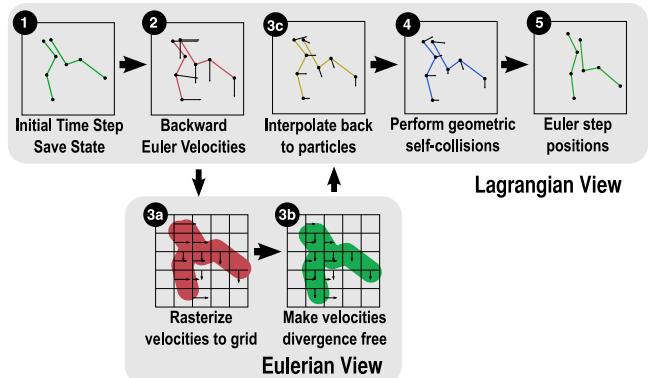


Figure 3: Outline of the Lagrangian (top) and Eulerian (bottom) components of our method.

and every other particle with a bending spring. Torsion could be added to the mass/spring model, or a different strand model (such as an articulated rigid body or Cosserat model) could be used. Each time integration step of the dynamic system proceeds as follows (see Figure 3):

1. Save collision-free position and velocity
2. Velocity step $v_i^{*n+1} = v_i^n + \Delta t a_i(x_i^n, v_i^{*n+1})$
3. Use the volume technique to modify v_i^{*n+1} into a corrected velocity \hat{v}_i^{n+1} (Substeps 3a, 3b, 3c)
4. Modify \hat{v}_i^{n+1} for collisions to get v_i^{n+1}
5. Compute final position $x_i^{n+1} = x_i^n + \Delta t v_i^{n+1}$

where x_i^n , v_i^n is the time t_n position and velocity of the i th particle, respectively, and Δt is the time step. Note that step 4 is discussed in Section 4.2 and implemented using [Bridson et al. 2002]. $a_i(x_i^n, v_i^{*n+1})$ is separated into a linear damping part and non-linear elastic part as in [Bridson et al. 2003] to preserve elastic modes. Step 3 rasterizes the hair volume to a grid representing the hair density and velocity, modifies the velocity field to handle bulk self-interaction and then applies the modified information back to the particle velocities (Section 4.1).

4 Solver

4.1 Volume

A volume method is an ideal way to handle self-interaction of many hairs because when many hairs are near each other, their contact and collisions allow a propagation of any force through hairs in contact. This behavior is similar to the behavior of a fluid, which conserves momentum and mass. For computational efficiency, we draw on incompressible fluid simulation techniques to model our hair volume. Density modeling is still possible as discussed later in Section 4.1.1.

The two common approaches used to model fluid volumes are Lagrangian techniques (smoothed particle hydrodynamics, vortex particle methods) and Eulerian techniques (common pressure/velocity incompressible solvers such as [Stam 1999]). Eulerian techniques tend to be more efficient because nearest neighbor searches are unnecessary. The downside of Eulerian approaches is that they are most natural on uniform grids that limit the amount of detail available. Even particle based methods actually limit detail, because high degree unstructured stencils also create numerical smoothing. For our purposes, we will use an Eulerian approach and compensate for this lack of detail by using high resolution Lagrangian collision handling techniques to capture high fidelity detail.

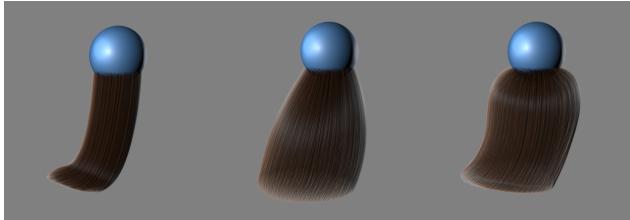


Figure 4: A demonstration of volumetric density control. (Left) high density and low volume. (Middle) moderate density and moderate volume. (Right) low density and high volume.

An increasingly popular approach for fluid advection is the FLIP method (introduced to graphics by [Zhu and Bridson 2005]) which replaces the traditional Eulerian advection equation of velocity $v_t + (v \cdot \nabla)v = 0$ with a Lagrangian step $x_i^{n+1} = x_i^n + \Delta t v_i^n$. This velocity is rasterized to a grid and made divergence free using the Chorin projection method [Chorin 1967]. The divergence free velocity field is compared to the original grid velocity field, and this difference is interpolated to the particles and applied as an impulse. This can be thought of as a particle fluid method that uses incompressible techniques to provide an implicit modeling of incompressible behavior rather than an explicit one, providing a convenient way to communicate between the grid and particles. In fact, [Losasso et al. 2008] demonstrate a hybrid grid/particle fluid technique that uses FLIP as a coupling mechanism between the grid and the particles. Given the input candidate velocities $v_i^{\star n+1}$ and positions x_i^n from step 2, we follow steps similar to [Losasso et al. 2008], though it is important to note our candidate positions contain the effects of elasticity.

At each timestep, we resize our grid to fit the volume of hair, keeping the grid spacing constant. We then rasterize candidate time t^{n+1} segments at their candidate locations defined by forward Euler, instead of *particles* directly, to ensure good coverage of the volume. Consider a segment with particles $(i, j) \in S$ at its candidate positions (i.e. $(x'_i, x'_j) = (x_i^n + \Delta t v_i^{\star n+1}, x_j^n + \Delta t v_j^{\star n+1})$). Given a point x in space which has distance $d_{ij}(x)$ to the segment, define a weight $w_{ij}(x) = \max(0, r - d_{ij}(x))$, where r is a user-defined radius of influence for each segment. We typically used $r = 3\Delta x/2$ in our simulations where Δx is the grid size. Then the rasterized velocity at any point is

$$v(x) = \frac{\sum_{(i,j) \in S} w_{ij}(x)[(1 - \alpha_{ij}(x))v_i + \alpha_{ij}(x)v_j]}{\sum_{(i,j) \in S} w_{ij}(x)}$$

where $\alpha_{ij}(x)$ is the interpolation fraction of the closest point on the segment. Velocities and weights are computed on the faces (for MAC velocities) and cell centers for density control and separation condition computation.

Once the velocities and weights are rasterized, we must setup a Poisson system $\nabla^2 p = \nabla \cdot v_{\text{grid}}^{n+1}$ to project the grid velocities to be divergence free in step 3b. Any cell with a weight lower than a threshold is set as a Dirichlet $p = 0$ condition, and any face that is inside a kinematic collision body is given a zero Neumann boundary condition and the velocity on that face is set to the object velocity. A divergence source term is set to target density (see Section 4.1.1). We solve the system with preconditioned conjugate gradient and then compute the divergence free velocity field $v_{\text{grid}}^{n+1} = v_{\text{grid}}^{\star n+1} - \nabla p$. Additional viscosity could also be added at this stage to create additional friction (variable viscosity is also a possibility, see [Carlson et al. 2002; Rasmussen et al. 2004]).

Finally, in step 3c, FLIP is used to apply this velocity back to the i th particle using the formula

$$\hat{v}_i^{n+1} = \xi [v_i^{\star n+1} + (I(x_i^n, v_{\text{grid}}^{n+1} - v_{\text{grid}}^{\star n+1}))] + (1 - \xi) I(x_i^n, v_{\text{grid}}^{n+1})$$



Figure 5: A volumetric separation constraint can be defined to control the amount of sticking. (Left) no separation condition is applied. (Right) an easy to satisfy separation condition is specified to produce immediate decoupling.

where $I(x, v)$ is trilinear interpolation at location x of a vector field v and ξ controls the amount of FLIP vs. PIC. In our simulations, we typically used a value of $\xi = .95$.

4.1.1 Density Control

The rasterized cell weights are used to define a density which can be targeted to a user defined density as in [Losasso et al. 2008] by using a divergence source term in the Poisson equation. Near collision objects this method can lose effectiveness because kinematic velocity constraints interfere with divergence. Thus, we also modify the fixed velocities on Neumann faces if the face's weight is less than the density target. We simply add to the constrained velocity in the collision body's normal direction. This is analogous to a penalty repulsion in Lagrangian dynamics, but handling it in the Poisson equation means it will be made consistent globally. The results of density control can be seen in Figure 4.

4.1.2 Separation Control

A problem with volume/continuum approaches to hair is that nearby hairs are forced to behave similarly. This is desired when hair is under compression because it forces the velocity to be zero at the center, preventing interpenetration. However, when two regions of hair have disparate velocity fields, it may be desirable to control the amount of sticking this artificial coupling can create (Figure 5). In fact, this problem also appears in fluid techniques when solid objects are coupled to the fluid.

To prevent unwanted sticking, we compute a hair separation condition during the rasterization process. Consider a face of the grid having the two incident cell velocities v_1 and v_2 . A face is considered separating if $v_1 \cdot n - v_2 \cdot n < \gamma$, where n is the vector pointing from cell 1 to cell 2 and γ is a scalar separation parameter. This means that the domain of the grid should be decoupled at this face, and cell 1 should not see the pressures on cell 2 and *vice versa*. The row of the matrix of each cell is modified to see the other cell as a ghost Dirichlet $p = 0$ cell. This is accomplished simply by zeroing a_{ij} and a_{ji} in the matrix (preserving symmetry). Unfortunately, performing this change means we cannot project the face velocity because the gradient stencil $\frac{1}{\Delta x}(p_2 - p_1)$ is no longer defined. Thus, interpolation of velocities to particles in cell 1 or 2 for FLIP is not defined so these particles are not changed during the FLIP update. Even so, their collisions are resolved by Lagrangian self-collisions. See Figure 5 right for results of this approach.



Figure 6: A hand-animated character walking with a hairstyle simulated by our method. The rightmost image is a closeup. Images ©Disney. All Rights Reserved.

4.2 Lagrangian Collisions

Once step 3 is done, we have a velocity field that roughly considers self-collision and global collisions; however, it misses fine details because it has been computed at the coarser resolution of the grid. If we had only a few hundred guides, repulsions might be a good option for removing the remaining collisions, but at the densities of our examples, they become impractical. This is because repulsions are proximity based, so as density increases, the fact that thickness must decrease means that the repulsions look smaller compared to the velocities. Thus we turn to swept self-collisions to handle our fine collisions.

Geometric collisions have been studied extensively for cloth simulation because they are essential for preventing visual artifacts. The current state of the art is based on [Bridson et al. 2002] which uses a three stage process to ensure no collisions are missed. First, contact is preconditioned using penalty-based repulsions that are small enough to prevent visual artifacts. Second, self-collisions are applied to stop as much interpenetration as possible. In this step, collisions are detected by coplanarity tests between nearby point/face and edge/edge pairs, and impulses are applied to colliding pairs. As new collisions may have been created by these impulses, this step is iterated over until all collisions are resolved. Third, rigid groups (impact zones) are used as a final safety net, postconditioning the collisions. Subsequent papers have typically focused on improving the latter two components. For example [Sifakis et al. 2008] replaces the second step with a globally coupled collision scheme and [Harmon et al. 2008] improves rigid impact zones.

For cloth, improving collisions through better post-conditioning is a useful technique because failures in repulsions and self-collisions typically do not result in significant visual artifacts. In hair, however, a deluge of repulsions and collisions applied using relatively unstable edge/edge interactions results in configurations that would be difficult to correct with a better rigid group technique. [Selle et al. 2008] mentions many of these difficulties and in fact they resorted to turning off collisions in contact cases to prevent these pitfalls.

In our implementation, we follow the collision algorithm in [Brid-

son et al. 2002] for edge/edge pairs. However, the placement of our volume handling before applying self-collisions not only makes finding a final collision free configuration possible, but it actually preconditions the collision step—effectively replacing the repulsion step of Bridson. This allows us to prevent poor configurations before they create visual artifacts in the self-collision and rigid group steps which is essential in hair where the lack of stabilizing point/face interactions make collisions potentially more damaging. In fact, in many ways it is a better preconditioner than proximity-based repulsions, because our volume formulation considers velocities as well as positions. We illustrate the effectiveness of a volumetric response to stacking collisions in Figure 7.

5 Results

We have demonstrated our technique with a range of examples in our prototype system; one can easily implement it with a standard Lagrangian hair simulator and Eulerian fluid solver. Examples are rendered with Renderman and standard hair shading models. It would be feasible to apply the recent subsurface scattering and shadowing acceleration techniques of [Moon et al. 2008; Zinke et al. 2008; Bertails et al. 2005], some of which might benefit from our rasterized volume. While we do not generate any additional strands at render time, traditional interpolation or clumping techniques can naturally be utilized if more strands are desired for rendering detail. A typical example uses a grid with 60^3 cells in the volumetric step; however, this number is dynamically updated as the volume of hair changes shape.

We also demonstrate complex contact and collision using a braid pattern consisting of 1500 hairs with approximately 100 particles each in Figure 9. The three hair sections begin in a very loose configuration with the top particle of each hair fixed. As gravity pulls the hairs downward, a braid pattern emerges due to our Eulerian/Lagrangian collisions. Figure 6 shows our technique applied to an animated character walking with 10,000 simulated hairs. Our last example (Figure 8) compares our method to Lagrangian and Eulerian collisions alone. A bundle of 1200 hairs is draped across a perpendicularly hanging bundle of 1200 more hairs (240,000 particles). Whereas purely Lagrangian collisions create highly active collision impulses due to inadequate hair coupling, purely Eulerian collisions are overly damped and fail to resolve the collisions. Our result shows a moderate amount of coupling from the volume together with fine details obtained with Lagrangian collisions. Figure 1 shows that our method compares favorably to real hair behavior.

A comparison of runtimes for the three techniques shows the purely Eulerian technique has the lowest average runtime per frame

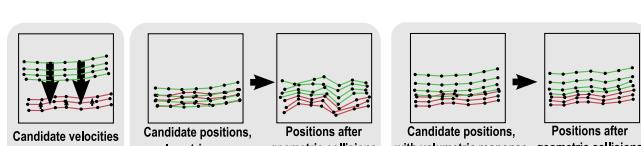


Figure 7: Collisions under stacking. A comparison of geometric collisions with (right) and without (left) volumetric preconditioning.

	Our Hybrid Method	Lagrangian Collisions	Eulerian Collisions
Both Lagrangian collisions and our method successfully stop the falling bundle while the Eulerian collisions do not.			
Our method exhibits a more natural coupling which still exhibits high fidelity detail.			
Geometric collisions time	1.14 min/frame	9.34 min/frame	N/A
Volume substep time	3.65 min/frame	N/A	3.55 min/frame
Total time	7.97 min/frame	13.63 min/frame	5.67 min/frame

Figure 8: A cylindrical bundle of hairs is dropped on another bundle of suspended hairs with a timing table (2400 hairs).

(5.6m). Although our method computes both volumetric and Lagrangian collisions, it is still significantly faster (8m) per frame than Lagrangian collisions alone (13.6m), showing the effectiveness of the Eulerian divergence-free solve as a collision preconditioner. A further breakdown of timing can be found in Figure 8, and we note that the remainder of time is spent on time integration.

We note that simulation time was about 15 minutes (26.9% Lagrangian collisions, 33.9% volumetric, 39.2% mass/ spring time integration) per frame for the character in Figure 6 on a single machine which is an improvement over the 16-way parallel runtimes of [Selle et al. 2008]. The time step was determined by the mass spring Courant condition, though we found in many examples that the volumetric step provided some extra stability, allowing us to relax the time step restriction.

6 Discussion

While we believe our technique makes high-fidelity interactions tractable, there are some limitations. We have not tried this method on non-straight hair as straight hair tends to exhibit the most complicated stacking configurations. Bridson’s collision handling algorithm in generalized coordinates may be difficult to apply because impulses are given in maximal coordinates, and the associated generalized coordinate response can be ambiguous. The FLIP impulse will be similarly difficult to apply. Thus, in future work we plan to experiment with the maximal coordinate mass-spring torsion model of [Selle et al. 2008]. We note, however, that the braid simulation (1500 hairs, 150 particles each) in Figure 9 shows both large and small scale interactions, showing promise for applications of our method.



Figure 9

In addition, the resolution of the volume creates some numerical viscosity and in particular angular velocity dissipation. This can be controlled by reducing the use of the volume (at the expense of less efficiency) or by increasing the resolution of the grid. Additionally, if the grid is too coarse, pieces of hair that become severely tangled may not be able to separate. Similarly since friction is modeled partially by viscosity (numerical and modeled), it is somewhat inaccurate. So in the future we would like to investigate using an octree grid, to allow different resolutions in different parts. For example, using the work of [Ward et al. 2003] in a large volume of hair, hair that is not visible may not require a highly detailed velocity field. Creating a level set by applying a fast marching method to the previous timestep’s density volume could derive a refinement criterion. Parallelization of the grid rasterization, Poisson solve, and FLIP application would also allow higher resolutions. We expect these algorithms to map well to multicore architectures and graphics hardware. We are also interested in applying this technique to cloth as a replacement to the Bridson repulsions to better precondition cloth as a replacement to the Bridson repulsions to better precondition cloth collisions.

7 Conclusion

We have presented a technique that hybridizes Lagrangian and Eulerian hair simulation techniques. Inspired by recent FLIP and SPH fluid technology, we show that our model can be useful as a way of controlling the integration of volume based forces. In addition we show how the volume can ease collision difficulties with hair by acting as an improved preconditioner. We have shown that the factorization of hair interaction into a coarse, globally-coupled phenomenon and a highly detailed Lagrangian view is an effective strategy. This improves efficiency by allowing both types of behaviors to be captured in the most natural way possible.

Acknowledgements

A. McAdams is supported in part by NSF DMS-0502315. E. Sifakis and J. Teran are supported in part by DOE 09-LR-04-116741-BERA, NSF DMS-0652427, NSF CCF-0830554, ONR N000140310071, and an Intel Larrabee Research Grant. We would like to acknowledge Maryann Simmons, Arthur Shek, and Joe Marks for useful discussions and support. Additionally, we appreciate the Walt Disney Animation Studio artists providing us with our animated character example.

References

- ANJKO, K., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. In *Comp. Graph. (Proc. SIGGRAPH 1992)*, 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.
- BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Trans. on Graph.* 27, 3, 1–12.
- 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., MÉNIER, C., AND CANI, M.-P. 2005. A practical self-shadowing algorithm for interactive hair animation. In *Graph. Interface*, 71–78.
- BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÈQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Trans. on Graph.* 25, 3, 1180–1187.
- BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proc. of SIGGRAPH 2002*, ACM, vol. 21, 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.
- CARLSON, M., MUCHA, P., VAN HORN, R., AND TURK, G. 2002. Melting and flowing. In *Proc. of the ACM SIGGRAPH Symp. on Comput. Anim.*, vol. 21, 167–174.
- CHANG, J., JIN, J., AND YU, Y. 2002. A practical model for hair mutual interactions. In *ACM SIGGRAPH/Eurographics Symp. on Comp. Anim.*, 73–80.
- 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.
- CHORIN, A. 1967. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* 2, 12–26.
- 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 *SCA '06: Proc. of the 2006 ACM SIGGRAPH/Eurographics Symp. on Comput. anim.*, 91–100.
- HARMON, D., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2008. Robust treatment of simultaneous collisions. *ACM Trans. on Graph.* 27, 3, 1–4.
- KIM, T.-Y., AND NEUMANN, U. 2002. Interactive multiresolution hair modeling and editing. In *Proc. of SIGGRAPH 2002*, ACM, vol. 21, 620–629.
- LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R., 2008. Two-way coupled SPH and particle level set fluid simulation.
- MOON, J. T., WALTER, B., AND MARSCHNER, S. 2008. Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph.* 27, 3, 1–7.
- 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., Pixar Animation Studios.
- PLANTE, E., CANI, M.-P., AND POULIN, P. 2002. Capturing the complexity of hair motion. *Graph. Models* 64, 1, 40–58.
- RASMUSSEN, N., ENRIGHT, D., NGUYEN, D., MARINO, S., SUMNER, N., GEIGER, W., HOON, S., AND FEDKIW, R. 2004. Directable photorealistic liquids. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 193–202.
- 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., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Trans. on Graph.* 27, 3, 1–11.
- SIFAKIS, E., MARINO, S., AND TERAN, J. 2008. Globally coupled impulse-based collision handling for cloth simulation. In *ACM SIGGRAPH/Eurographics Symp. on Comp. Anim.*
- 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 on Comput. Anim.*, 209–217.
- STAM, J. 1999. Stable fluids. In *Proc. of SIGGRAPH 1999*, ACM, 121–128.
- 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., 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.
- ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Trans. on Graph.* 24, 3, 965–972.
- ZINKE, A., YUKSEL, C., WEBER, A., AND KEYSER, J. 2008. Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. on Graph.* 27, 3, 1–10.