Real-time Hair Mesh Simulation

Kui Wu*
University of Utah

Cem Yuksel†
University of Utah

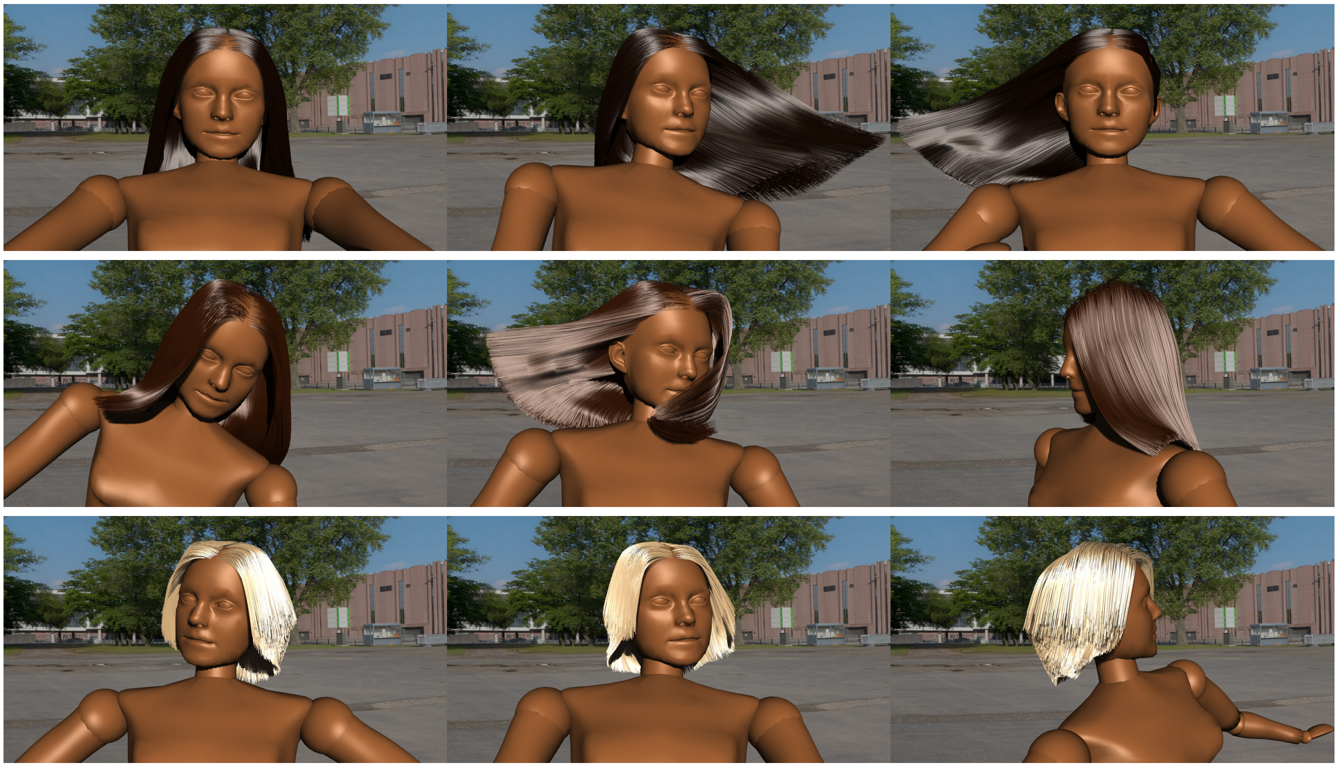


Figure 1: *Example frames captured from our real-time hair mesh simulation running at less than 6 milliseconds per frame on the CPU.*

Abstract

We present a robust real-time hair simulation method using hair meshes. Leveraging existing simulation models for sheet-based cloth, we introduce a volumetric force model for incorporating hair interactions inside the hair mesh volume. We also introduce a position correction method that minimizes the local deformation of the hair mesh due to collision handling. We demonstrate the robustness of our hair simulation method using large time steps with fast motion, and we show that our method can recover the initial hair shape even when the hair mesh goes through substantial deformation.

Keywords: hair mesh, hair simulation, real-time simulation

Concepts: •**Computing methodologies** → **Physical simulation;**

1 Introduction

Realistic hair simulation at real-time frame rates is an important problem in computer graphics. The high geometric complexity of a typical realistic hair model, which can involve more than a hundred thousand hair strands, combined with the highly anisotropic behavior of hair (with low bending and shear resistance coupled with high stretch resistance) makes developing computationally efficient and robust hair simulation models a challenge. Furthermore, our knowledge about the underlying physical interactions between individual hair strands involving collisions, friction, electrostatic forces, and especially various hair products has been limited. This makes it extremely difficult to solve the hair simulation problem using a bottom-up approach that begins with formulating individual hair strand motion with low-level interactions, and scales these formulations up to a complete hair model with realistic behavior.

Therefore, in this paper we have taken the opposite, top-down approach. We begin with a simplified representation of the full hair model in the form of a *hair mesh* [Yuksel et al. 2009]. Instead of formulating the individual hair strand interactions, we formulate the collective behavior of a group of hair strands that reside within each hair mesh extrusion, which is called a *bundle*. This allows us to build a highly efficient computational system for hair simulation that can achieve a desired high-level hair motion behavior, controlled by a small number of high-level parameters.

*e-mail:kwu@cs.utah.edu

†e-mail:cem@cemyuksel.com

I3D 2016, February 27–28 2016
ISBN: 978-1-4503-4043-4/16/03

DOI: http://dx.doi.org/10.1145/2856400.2856412

Our simulation system leverages the existing work on cloth simulation with large time steps [Baraff and Witkin 1998]. Many artists recognized the perceptual similarities between hair and cloth behavior, and successfully simulated hair using existing cloth solvers. However, the fact that hair is a volumetric structure, while sheet-based cloth is modeled as a thin surface, substantially limits what could be achieved with this approach. The hair simulation method we present introduces the missing volumetric element and allows simulating a hair mesh using a system that is reminiscent of existing cloth solvers. Therefore, we believe that our simulation system can be easily integrated into existing simulation systems, allowing robust and high-quality hair simulation at real-time frame rates.

Furthermore, we introduce a collision handling method that produces volumetric deformations for position correction. This allows resolving collisions using only the outer surface of the hair mesh, thereby reducing the complexity of collision detection and handling, which can be a bottleneck when using arbitrary meshes as collision objects. Furthermore, our position correction minimizes the energy injected due to position changes. This is a particularly important problem with fast motion and large time steps that can lead to large penetrations, and correcting them can compress the volumetric structure of hair, adding considerable energy.

We show that we can simulate hair meshes with large time steps and fast motion at real-time frame rates (Figure 1). We also demonstrate that the robustness of our formulation allows the initial hair mesh shape to be recovered even after the hair mesh goes through extraordinary deformations.

2 Background

Hair simulation has been a long standing problem in computer graphics [Ward et al. 2007]. In this section we briefly overview the prior work in this area.

The first mass-spring based simulation system for hair was proposed by Rosenblum et al. [1991]. However, in order to limit the stretching of the strand, the springs had to be stiff, which led to numerical instability unless the time-step was sufficiently small. Much later, Selle et al. [2008] improved this approach by introducing altitude springs to simulate the complex behavior of human hair along with a semi-implicit spring solver for stability. They aimed at simulating each and every hair strand, which resulted in an expensive simulation system, especially for real-time animation.

Anjyo et al. [1992] considered each hair strand as a chain of rigid sticks and used one-dimensional projective differential equations to solve the dynamics of hair motion. Since this formulation lacks torsional hair stiffness, a full 3D hair motion cannot be properly simulated. Moreover, this method cannot handle external forces well, as the motion is processed from the root to the hair tips. Lee and Ko [2001] extended this method by adding an extra force to preserve the hairstyle under collision.

Chang et al. [2002] represented hair strands as rigid multi-body chains and simulated only a subset of hair strands as guide strands. Simulated guide strands were connected to nearby guide strands for introducing hair-hair interaction. Choe et al. [2005] proposed a hybrid model, which takes advantage of both the mass-spring model and the rigid multi-body chain. They could formulate an implicit integrator by introducing springs that connect the rigid bodies for achieving more efficient and stable simulation.

The Kirchhoff's theory for elastic rods has been utilized by Bertails et al. [2006] for accurately predicting the motion of individual hair strands. One advantage of this model is that inextensibility is intrinsically handled. This model is particularly effective for simulating the dynamics of curly hair strands. However, the model was parameterized using reduced coordinates, making it difficult to introduce external constraints. An extension of this work formulated each element locally, thereby allowed solving the dynamics of a super-helix in linear time [Bertails 2009].

There are also simulation methods for wisp-based hair models [Yang et al. 2000; Kim and Neumann 2002]. Plante et al. [2001] embedded wisps in a deformable envelope simulated using a mass-spring system. The main computational complexity of this model is in the collision detection and handling among wisps that form a hairstyle, which was also a source of flickering. Bertails et al. [2003] introduced the adaptive wisp tree model in order to reduce the collision computation. This model is effective for large-scale to mid-scale behavior of long and thick hairstyles that can utilize dynamic splitting and merging of the adaptive tree.

Some researchers draw a parallel between hair and fluid motion. Hadap and Magnenat-Thalmann [2001] used a Lagrangian fluid formulation to model the interactions between hair strands. Bando et al. [2003] also used a Lagrangian fluid formulation for sampling the volume. McAdams et al. [2009] proposed a hybrid Eulerian and Lagrangian fluid system for simulating hair that could preserve volume. While these approaches could generate the bulk hair motion, they operate on a large number of interacting particles, which makes them expensive, especially for real-time hair simulation.

For achieving more realistic interactions between individual hair strands, Daviet et al. [2011] introduced a physically based friction model between hair strands as well as hair and the body. This model could produce impressive results, especially for small-scale hair interactions between small groups. Derouet-Jourdan et al. [2013] built on this model, recognizing the importance of computing the initial hair state coming out of a modeling system, so that the simulation does not alter the rest shape of the input hairstyle.

In order to simulate hair at real-time frame rates, researchers preferred simplified hair models, such as 2D strips [Koh and Huang 2001], a cubic lattice representation [Volino and Magnenat-Thalmann 2004], and short hair strips [Yang and Huang 2002]. Bando et al. [2003] used a continuum approach while discarding particle connectivity for real-time applications. Tariq and Bavoil [2008] introduced a fast technique for inter-hair collisions using a hair density field represented by a voxelized grid. Recently, Müller et al. [2012] proposed a *Follow The Leader* method to simulate inextensible hair and fur in real-time using position-based dynamics [Müller et al. 2007]. This method can simulate a large number of individual hair strands in real-time, but the interaction between strands are handled using a simple density field, which limits the realism. Sanchez-Banderas et al. [2015] extended this method by introducing shape preservation constraints. Chai et al. [Chai et al. 2014] proposed a real-time hair simulation method based on a data-driven interpolation of guide strands. Their approach can animate 150 thousand strands in real-time by using a small number of guide strands and produces exceptionally high-quality animations, but it requires previously generated full simulation data for the examples it can reproduce at run time.

In this paper we describe simulation of hair meshes using a volumetric force model. Teschner et al. [2004] introduced multiple constraints forces model to preserve length, area, and volume. For robust simulation of inverted tetrahedra, Irving et al. [2006] used diagonalization of the deformation for determining the recovery force. Twigg and Kačić-Alesić [2011] provided an optimization approach for preserving the initial hair shape after applying gravity. There is also prior work on robust collisions handling for volumetric elements [Faure et al. 2008; Sifakis et al. 2008; Allard et al. 2010; Müller et al. 2015].

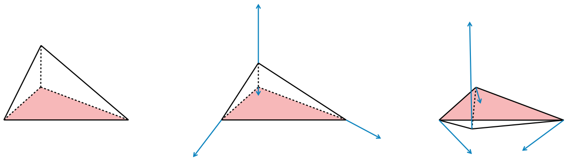

(a) *Rest Shape* **(b)** *Compressed* **(c)** *Inverted*

Figure 2: *The volume constraint force is always in the direction to recover the volume. Blue lines are the force directions applied on a tetrahedron with a particular (a) rest shape, which is (b) compressed and (c) inverted.*

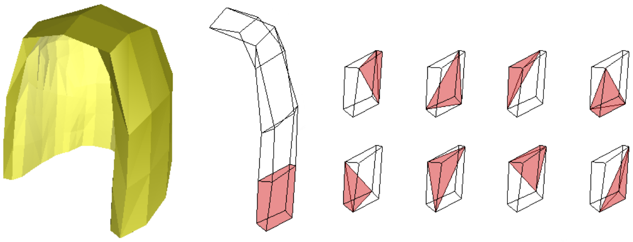


Figure 3: *Tetrahedralization of a hair mesh: (a) the hair mesh model, (b) a bundle of the hair mesh highlighting a single prism, and (c) eight tetrahedra that are generated inside this prism.*

3 Hair Mesh Simulation

The simulation system we introduce in this paper animates the hair mesh structure, which is a popular approach for modeling hair [Yuksel et al. 2009]. Our simulation framework is based on the sheet-based cloth simulation method of Baraff and Witkin [1998]. We use the same force formulations and the Modified Conjugate Gradient method for implicit Euler integration. We employ collision constraints for the surface vertices of the hair mesh that go through collision.

We introduce a volumetric force model that resists against volume compression and expansion. This volumetric force, combined with the other forces along the horizontal edges of the hair mesh, models the interactions between hair strands. We also present a collision handling scheme that minimizes the variation in relative vertex positions due to position correction.

3.1 The Volume Force Model

We use a volume force defined in a tetrahedron, calculated similar to Teschner et al. [2004]. Let $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ be the vertices of the tetrahedron and V_0 be its rest volume. Our volume force is defined by the condition

$$C_{volume} = \frac{1}{6}\mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3) - V_0 \ , \tag{1}$$

where

$$\begin{aligned}\mathbf{e}_1 &= \mathbf{x}_1 - \mathbf{x}_0\\ \mathbf{e}_2 &= \mathbf{x}_2 - \mathbf{x}_0\\ \mathbf{e}_3 &= \mathbf{x}_3 - \mathbf{x}_0\end{aligned}$$

are three edge vectors of the tetrahedron. This leads to a volume force f_i on each vertex i defined as

$$f_i = -k\frac{\partial C_{volume}}{\partial x_i} C_{volume}(x) \ , \tag{2}$$

where k is the stiffness parameter. Thus, the force on each vertex is

$$\begin{aligned}f_0 &= k\,(\mathbf{e}_2 - \mathbf{e}_1) \times (\mathbf{e}_3 - \mathbf{e}_1)\,C_{volume}\\ f_1 &= k\,(\mathbf{e}_3 \times \mathbf{e}_2)\,C_{volume}\\ f_2 &= k\,(\mathbf{e}_1 \times \mathbf{e}_3)\,C_{volume}\\ f_3 &= k\,(\mathbf{e}_2 \times \mathbf{e}_1)\,C_{volume} \ .\end{aligned}$$

Note that our volume constraint allows a tetrahedron to recover its volume even when it is flattened or inverted. Figure 2 shows that our volume constraint leads to forces that push the tetrahedron in the right direction for recovering its rest volume. Also notice that our force formulation differs from the tetrahedral volume force used by Selle et al. [2008], which introduces only a single spring force, the direction of which is selected based on the momentary shape of the tetrahedron. While their approach minimizes the number of spring forces, it is not temporally coherent, which can lead to flickering.

Our volume force model operates on a tetrahedron; therefore, we need to convert the hair mesh volume into a collection of tetrahedra. The hair mesh structures we consider are composed of triangular and rectangular prisms, and it is easy to convert each triangular or rectangular prism into three or five disjoint tetrahedra respectively. On the other hand, this conversion is not symmetrical and would lead to bias in the way that the volume force would be applied to the vertices. Therefore, we generate partially overlapping tetrahedra instead, placed at all corners of each prism, as shown in Figure 3. While this configuration does not completely cover the volume of the prism and the tetrahedra of a prism overlap, they provide a symmetrical force distribution and are affected by any volume change that the prism can go through.

3.2 Collision Handling

Our collision handling strategy involves collision detection followed by position correction and velocity update. We use the surface of the hair mesh as the collision object; therefore, our approach is compatible with existing collision detection methods. For faster collision detection, we only consider the external vertices of the hair mesh.

Position correction, however, cannot be handled by only moving the intersecting external vertices, since the internal vertices of the hair mesh might be penetrating a collision object as well. Including the internal vertices (or the internal structure of the hair mesh) in the collision detection does not resolve this issue, because the position correction should place the internal vertices farther away from the surface of the collision object. Even when the internal vertices do not penetrate the collision object, limiting the position correction to external vertices deforms the hair mesh volume, thereby injecting deformation energy into the simulation. While our volume force would act to recover the hair volume in the subsequent simulation steps, the injected energy due to position correction can lead to flickering in hair motion.

We solve this problem by first updating the positions of the penetrating external vertices and then minimizing the deformation by updating the positions of all other vertices accordingly (Figure 4). Let $\mathbf{x}$ be the vector that denotes the positions of all hair mesh vertices before collision handling. We first modify this vector by repositioning the penetrating external vertices and moving them outside of the collision objects. We denote the updated vector with position

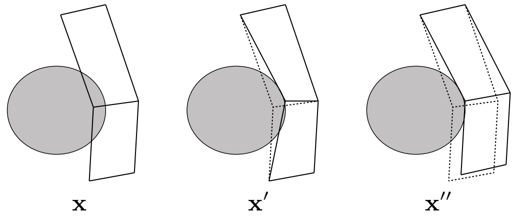


Figure 4: *Collision handling: $\mathbf{x}$ is the positions before collision handling, $\mathbf{x}'$ is the positions after position correction, and $\mathbf{x}''$ is the final positions after deformation minimization.*

corrections as $\mathbf{x}'$. This new vector does not include any penetrating external hair mesh vertices, but it may include excessive deformation due to position correction and penetrating internal hair mesh vertices. We compute the final positions of the hair mesh vertices $\mathbf{x}''$ by solving a minimization problem that minimizes the local deformations due to position correction. This minimization is of the form

$$\min \sum_j \sum_i \left\| (\mathbf{x}''_i - \mathbf{x}''_j) - (\mathbf{x}_i - \mathbf{x}_j) \right\|^2 \tag{3}$$

for all pairs of vertices i-j, such that vertices i and j are connected with a hair mesh edge, subject to the constraint that external vertices that went through position correction remain collision-free. Note that this is an over-constrained problem, since there are more edges then vertices. This minimization can be written in matrix form as

$$\min \left\| \mathbf{A}^T \mathbf{A} \mathbf{x}'' - \mathbf{A}^T \mathbf{A} \mathbf{x} \right\|^2 , \tag{4}$$

where A is the edge connection matrix, such that each row of A corresponds to an edge and contains one 1 at column i and one -1 at column j for an edge that connects vertices i and j, subject to the same constraints.

We solve the deformation minimization using the Modified Conjugate Gradients algorithm [Baraff and Witkin 1998]. This allows us to enforce the constraints within the Conjugate Gradients solver and directly use the matrix $\mathbf{A}^T\mathbf{A}$, instead of adding the constraints into the matrix. As a result, we can precompute the matrix $\mathbf{A}^T\mathbf{A}$ at the beginning of the simulation. We use $\mathbf{x}'$ as the initial guess. Figure 5 shows an example hair mesh going through position correction and deformation minimization.

4 Implementation and Results

We tested our hair mesh simulation model using a single-threaded CPU implementation. Therefore, our implementation does not take advantage of parallelism that could be exploited for faster simulation results. Nonetheless, we can achieve fast and robust hair simulation that takes less than 6 milliseconds per frame with relatively low-resolution hair mesh models, shown in Figure 1. All timings are measured on a 3.20GHz Intel i7-3930K CPU.

Our test setup uses different hair mesh models that are attached to a mannequin, which is animated using the motion capture data from ARCSim [Narain et al. 2012; Narain et al. 2013; Pfaff et al. 2014]. For collision geometry we use a number of ellipsoids that approximate the body of the mannequin. This way, collision detection can be handled quickly, but the differences between the mannequin body and the collision ellipsoids lead to gaps and visible intersections.

We render the hair model by generating the hair strands from the hair mesh, as described by Yuksel et al. [2009]. At each time step, we pass the modified hair mesh vertex positions as a texture to the geometry shader on the GPU and generate the individual hair strands using the geometry shader. We compute the hair shadows using Deep Opacity Maps [Yuksel and Keyser 2008].

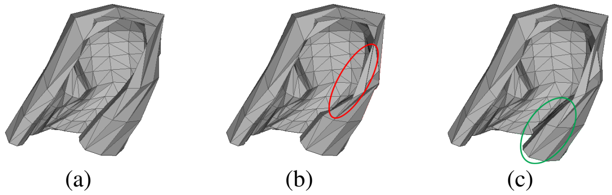


Figure 5: *Collision handling: (a) hair mesh before collision handling, (b) hair mesh after position correction moves some vertices in the red circle due to collision with the head, and (c) the final positions after deformation minimization moves some points, particularly in the green circle.*

Figure 6 gives the percentage of the simulation times used for different operations with different hairstyles and resolutions. Our performance tests consistently show that about half of the simulation time is spent on the modified conjugate gradients solver for animating the hair mesh, and about a quarter of the simulation time is used for computing the forces. The deformation minimization takes 10% to 20%, depending on the complexity of the hair mesh model. The rest of the simulation time is spent on animating the mannequin, computing the surface normals, and collision detection.

The simulation times we measured are almost linearly correlated with the number of hair mesh vertices. We can simulate the low-resolution short hair model with 215 vertices in Figure 1 at 4.8 milliseconds per frame, while the other two hair models in the same figure with 361 vertices took about 6 milliseconds per frame. Our implementation could simulate a higher resolution version of the long hair model with 2305 vertices in about 36 milliseconds per frame.

Robustness is one of the important features for our system. In order to test it we increase the playback speed of mannequin motion up to a factor of 10. This forces the hair mesh model to go through drastic deformations. While motion with such high speeds is unrealistic, it is not unusual for virtual characters to go through unrealistically fast motion. In our tests the hair mesh model could always recover its initial shape, regardless of the severity of the deformation it un-

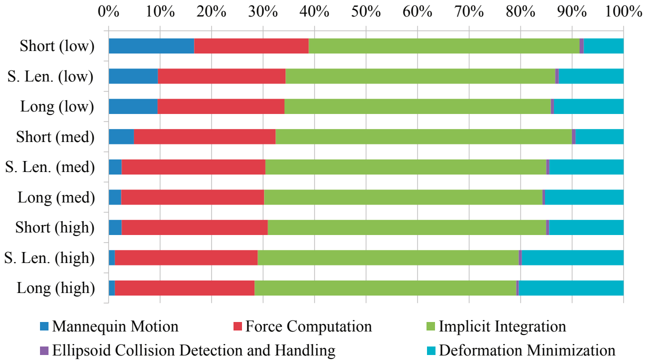


Figure 6: *Performance of different hair styles shown in Figure 1 (short, shoulder-length, and long) at different resolutions (low, medium, and high).*

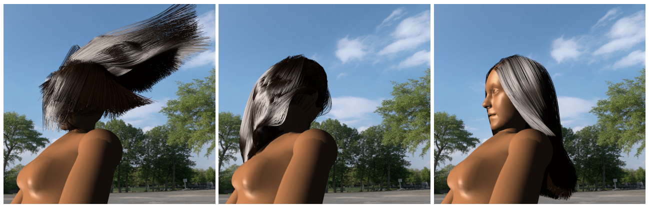


Figure 7: *Hair mesh recovering from an extreme deformation after body motion is turned off: (left) initial deformation, (middle) intermediate state, and (right) the recovered rest pose.*

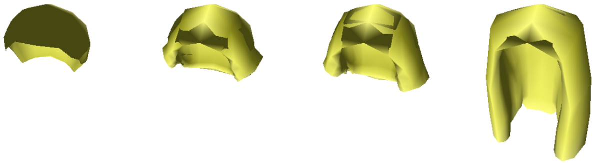


Figure 8: *Testing the robustness of our simulation model: (left) hair mesh collapsed by placing the vertices on their root points, (middle) small number of recovery steps taken, and (right) the final rest shape recovered.*

dergoes, even without adaptive time stepping. One example recovery from an extreme deformation is shown in Figure 7.

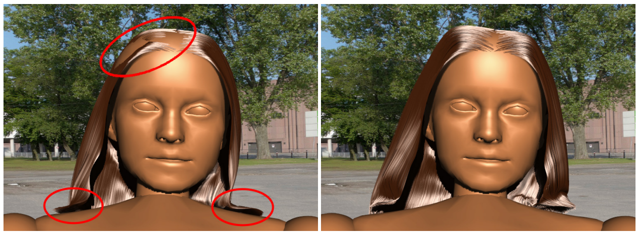
To test the robustness of our volume force model further, we collapsed the hair mesh by moving all vertices exactly onto their corresponding root points. We use no collision objects in this test. As can be seen in Figure 8, the hair mesh can quickly recover from this pathological configuration as well.

We show the importance of our deformation minimization in Figure 9. When no deformation minimization is used, the volume force alone can fail to recover the volume altered by position correction. Using small time steps mitigates this problem. However, large time steps can lead to relatively large penetrations and the volume force may need several iterations for recovering the hair volume. While deformation minimization can successfully preserve the volume of the hair mesh, disabling it can produce partially collapsed hair mesh volume and fail to resolve the penetration of the internal hair mesh vertices, as shown in the highlighted regions in Figure 9.

5 Conclusion and Future Work

We have presented a hair mesh simulation system, introducing a volumetric force model and a deformation minimization approach for properly handling collisions. Our implementation shows that our method can produce high-quality hair animations at real-time frame rates. We also demonstrated that our hair mesh simulation is robust and it can recover the initial hair shape even after it undergoes extreme deformations.

Simulating hair using a hair mesh has an important limitation: the topological structure of the hair remains intact throughout the simulation. In other words, our simulation method cannot split neighboring hair bundles, if such splits are not present in the input hair mesh model. On the other hand, this limitation provides an extra level of simulation control and the ability to recover the hair shape after substantial deformation, both of which are desirable features for most physically based animation applications in both real-time and offline animation systems.



No Deformation Minimization With Deformation Minimization

Figure 9: *Our deformation minimization recovering hair volume: showing results with and without deformation minimization after a number of time steps.*

One important future direction would be eliminating the restriction that the hair mesh topology remains intact throughout the simulation, using an adaptive splitting and merging approach. Also, our deformation minimization can lead to penetration of vertices that are not constrained, which can be addressed in future work.

Our results show that the hair mesh simulation approach we presented is a powerful method for simulation hair in real-time graphics applications, and it can also be utilized in high-quality offline animations. Our hair simulation approach can be used for simulating more complex hair mesh models. However, complex hair mesh models may also require handling self-collisions of the hair mesh, which can easily become the bottleneck of the simulation.

Acknowledgements

We thank Ian Mallett for his help. This work was supported in part by NSF grant #1538593.

References

ALLARD, J., FAURE, F., COURTECUISSE, H., FALIPOU, F., DURIEZ, C., AND KRY, P. G. 2010. Volume contact constraints at arbitrary resolution. *ACM Trans. Graph. 29*, 4, 82:1–82:10.

ANJYO, K.-I., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. *SIGGRAPH Comput. Graph. 26*, 2 (July), 111–120.

BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2003. Animating hair with loosely connected particles. *Computer Graphics Forum 22*, 3, 411–418.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proceedings of SIGGRAPH '98*, 43–54.

BERTAILS, F., KIM, T.-Y., CANI, M.-P., AND NEUMANN, U. 2003. Adaptive wisp tree: A multiresolution control structure for simulating dynamic clustering in hair motion. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, 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. In *ACM SIGGRAPH 2006 Papers*, ACM, SIGGRAPH '06, 1180–1187.

BERTAILS, F. 2009. Linear time super-helices. *Comput. Graph. Forum 28*, 2, 417–426.

CHAI, M., ZHENG, C., AND ZHOU, K. 2014. A reduced model for interactive hairs. *ACM Trans. Graph. 33*, 4 (July), 124:1–124:11.

CHANG, J. T., JIN, J., AND YU, Y. 2002. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '02, 73–80.

CHOE, B., CHOI, M. G., AND KO, H.-S. 2005. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '05, 153–160.

DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, SA '11, 139:1–139:12.

DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., DAVIET, G., AND THOLLOT, J. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph. 32*, 6 (Nov.), 159:1–159:10.

FAURE, F., BARBIER, S., ALLARD, J., AND FALIPOU, F. 2008. Image-based collision detection and response between arbitrary volume objects. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 155–162.

HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. *Computer Graphics Forum 20*, 3, 329–338.

IRVING, G., TERAN, J., AND FEDKIW, R. 2006. Tetrahedral and hexahedral invertible finite elements. *Graph. Models 68*, 2 (Mar.), 66–89.

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

KOH, C., AND HUANG, Z. 2001. A simple physics model to animate human hair modeled in 2d strips in real time. In *Computer Animation and Simulation 2001*, Springer-Verlag New York, Inc., New York, NY, USA, 127–138.

LEE, D.-W., AND KO, H.-S. 2001. Natural hairstyle modeling and animation. *Graph. Models 63*, 2 (Mar.), 67–85.

MCADAMS, A., SELLE, A., WARD, K., SIFAKIS, E., AND TERAN, J. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph. 28*, 3 (July), 62:1–62:6.

MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (Apr.), 109–118.

MÜLLER, M., KIM, T.-Y., AND CHENTANEZ, N. 2012. Fast simulation of inextensible hair and fur. In *VRIPHYS*, Eurographics Association, J. Bender, A. Kuijper, D. W. Fellner, and E. Guérin, Eds., 39–44.

MÜLLER, M., CHENTANEZ, N., KIM, T.-Y., AND MACKLIN, M. 2015. Air meshes for robust collision handling. *ACM Trans. Graph. 34*, 4 (July), 133:1–133:9.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (Nov.), 147:1–10. Proceedings of ACM SIGGRAPH Asia 2012, Singapore.

NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. 32*, 4 (July), 51:1–51:8.

PFAFF, T., NARAIN, R., DE JOYA, J. M., AND O'BRIEN, J. F. 2014. Adaptive tearing and cracking of thin sheets. *ACM Trans. Graph. 33*, 4 (July), 110:1–110:9.

PLANTE, E., CANI, M.-P., AND POULIN, P. 2001. A layered wisp model for simulating interactions inside long hair. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 139–148.

ROSENBLUM, R. E., CARLSON, W. E., AND TRIPP, E. 1991. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation 2*, 4, 141–148.

SÁNCHEZ-BANDERAS, R., BARREIRO, H., GARCÍA-FERNÁNDEZ, I., AND PÉREZ, M. 2015. Real-time inextensible hair with volume and shape. In *Congreso Español de Informática Gráfica, CEIG'15*, 1–8.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Trans. Graph. 27*, 3 (Aug.), 64:1–64:11.

SIFAKIS, E., MARINO, S., AND TERAN, J. 2008. Globally coupled collision handling using volume preserving impulses. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 147–153.

TARIQ, S., AND BAVOIL, L. 2008. Real time hair simulation and rendering on the gpu. In *ACM SIGGRAPH 2008 Talks*, ACM, New York, NY, USA, SIGGRAPH '08, 37:1–37:1.

TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., AND GROSS, M. 2004. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, 312–319.

TWIGG, C. D., AND KAČIĆ-ALESIĆ, Z. 2011. Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 225–236.

VOLINO, P., AND MAGNENAT-THALMANN, N. 2004. Animating complex hairstyles in real-time. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, VRST '04, 41–48.

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 Transactions on Visualization and Computer Graphics 13*, 2 (Mar.), 213–234.

YANG, G., AND HUANG, Z. 2002. A method of human short hair modeling and real time animation. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, PG '02, 435–.

YANG, X. D., XU, Z., YANG, J., AND WANG, T. 2000. The cluster hair model. *Graphical Models 62*, 2, 85 – 103.

YUKSEL, C., AND KEYSER, J. 2008. Deep opacity maps. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2008) 27*, 2, 675–680.

YUKSEL, C., SCHAEFER, S., AND KEYSER, J. 2009. Hair meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009) 28*, 5, 166:1–166:7.