A Layered Wisp Model for Simulating Interactions inside Long Hair

Eric Plante
Taarna Studios Inc.
Current affiliation: discreet. Eric.Plante@discreet.com

Marie-Paule Cani
iMAGIS-GRAVIR/IMAG, joint lab of CNRS, INPG, INRIA, UJF. Marie-Paule.Cani@imag.fr

Pierre Poulin
Université de Montréal. poulin@iro.umontreal.ca

Abstract. This paper presents a method for animating long hair while modelling both interactions between the hair and the character's body and between different hair wisps. Our method relies on a layered model paradigm. Hair is structured into a number of volumetric wisps whose motion and deformation are computed using a layered model: A first layer, the skeleton curve, computes the large scale motion of a wisp. This skeleton is coated by a second layer, the deformable wisp envelope, linked to the skeleton through highly viscous springs. A third layer is used for rendering the individual hair strands within each wisp. During motion, anisotropic interactions are computed between hair wisps, in addition to interactions with character body: two quasi-parallel wisps are allowed to interpenetrate while a viscous collision is computed between colliding wisps of different orientation. This results in a visually-realistic animation, that captures both continuities and discontinuities that can be observed in thick, long hair.

Keywords: hair, collision detection and response, layered models, physically-based animation, natural phenomena.

1 Introduction

Since the first appearance of synthetic humans in computer graphics, hair has been a major obstacle in producing realistic characters. In particular, the motion of hair is rather difficult to animate, due to the large number of primitives involved (about 100,000 hair strands), and to the complexity of its interactions with the 3D scene and with itself.

Most techniques to animate hair are based on dynamics. One of the first attempts was presented by Rosenblum *et al.* [14]. The movement of each individual hair strand is approximated using a chain of point-masses linked by stiff springs and hinges. The enormous amount of calculation required for their examples limited the number of s-trands to about 1000, which might explain why subsequent research has not followed this approach.

Instead, several researchers [7, 2, 16] have been inspired by an approach introduced by Anjyo *et al.* [3]. The initial position of each strand is determined by a cantilever beam simulation, and its movement is simulated with a set of rigid sticks, only from root to tip. Penetrations of sticks into the character body are detected and avoided during this process. However, since a given stick has no influence on the sticks closer to the root, the motion of a hair strand cannot be adequately modified during a collision. Most researchers simply avoid the problem by animating only short hair that do not reach the shoulders.

Watanabe and Suenaga [19] were the first ones to take advantage of the coherence in the movement of hair by animating wisps of hair. They were followed by Kim [11] and by Koh [12], who respectively modelled wisps using thin-shell volumes and semi-transparent polygon strips. None of these methods use physically-based animation of wisps. They mainly focus on improving the aspect of hair, and easing the modelling of hair-styles.

For approaches based on dynamics, coherence in hair motion has been exploited by animating only a few strand guides, and interpolating the motion of the strands in between [7]. However, this approach only detects collisions between the guides and the character, causing interpolated strands to possibly penetrate surfaces. This happens when initially neighboring guides end up on opposite sides of the head. A solution for modelling hair strands that avoid obstacles is to attach them to the streamlines of a fluid flow [9]. However, this technique generates neighboring hair that always have the same orientation, and thus fail to capture the case when two wisps of different orientations collide. Both the interpolation and the fluid flow approaches can lead to animations that appear too continuous, restricting these methods to fairly straight and clean hair.

The very nature of the previous approaches has caused researchers to ignore phenomena which have a great influence on the movement of hair, namely hair self-interactions in the form of inelastic collisions and friction between hair strands. With these methods, the kinetic energy is not sufficiently dissipated, and the hair does not tend to come back to its rest density after compression. This explains why computer generated hair always seems too "light" and lacks "volume". In later works from Hadap *et al.* [10], the problem of volume is addressed thanks to their fluid continuum approach, yet the strong discontinuities typical of long hair in movement are still absent.

2 Overview

Similarly to previous techniques, our hair motion is derived from physically-based animation. In doing so, our goal is not one of exact physical correctness, but instead, of increased visual realism obtained within a reasonable amount of computation time. We believe that modelling the complex interactions that occur both within the hair and between the hair and obstacles is a key step towards this goal.

It is generally accepted that the movement of hair strands should be approximated by calculations at a coarser level. Perhaps the inherent difficulties of previous techniques can be explained by the underlying assumption that this coarser level is still a single hair strand. They animate a few strands from which they deduce the positions of others. The approach we present in this paper breaks away from animating individual strands. Instead, we cluster groups of strands into wisps, modelled as anisotropic viscous volumes.

Since the wisp provides the coarser level of computation required to attain reasonable calculation times, all wisps can be animated, and thus the aforementioned disadvantages of interpolation are avoided. Our approach allows for more precise and efficient collision detection with the character's body, and can model the discontinuities easily observed in thick, long hair. Most importantly, complex hair interactions are simulated easily and relatively efficiently, by modelling their effects on both the shape and the motion of wisps.

The next section presents our model for hair wisps. Section 4 explains how wisps are used for processing interactions. Implementation details and results are given in Section 5. We finally conclude and discuss future directions.

3 A Layered Model for Hair Wisps

Interactions seem to be a dominent factor in hair motion, since each hair strand is always in contact with others. However, simulating collisions and friction between 100,000 hair strands would be prohibitively expensive. This paper is based on the idea of structuring hair into a number of deformable wisps. Each wisp, modelled as a viscous volume, simulates the motion of neighboring hair strands of similar orientations, subject to friction with one another. Wisps are also used for processing other interactions inside hair at a coarser level. This section details the layered model we use for a wisp.

3.1 Modelling a Hair Wisp

Defining layered models [6] is a very good way of modelling complex objects to animate. Such models decrease the complexity of the phenomena to model by structuring the object into a small set of distinct layers that may interact together. The different aspects of the model, embedded into different layers, may be simulated and controlled at totally different scales. This paradigm has been used, for instance, for modelling animated characters [6, 15], and for simulating a variety of complex deformable bodies [17, 5]. Most often, one of the layers is used for capturing high scale motion, another one models deformations, and a third defines the geometry to render. We are using the same kind of paradigm for animating hair wisps.

In our model, a wisp is structured into:

1. A *skeleton curve* that defines its large-scale motion and deformations.
2. A deformable *envelope* that coats the skeleton, and defines the deformation of the wisp sections around it.
3. A certain number of *hair strands* that are distributed inside the wisp envelope and that are only used at the rendering stage of the process.

The remainder of this section details these layers.

3.2 Wisp Skeleton

In our model, the general (global) movement and deformation of a wisp is modelled by its skeleton, which defines a curve in space. We animate this curve using point dynamics, thus neglecting the dynamic twisting of hair wisps. This approximation is acceptable since wisps of hair offer a relatively strong resistance to torsion.

The first segment of the wisp skeleton is not part of the simulation: It is set to penetrate inside the character's head, and thus provides a reference point that always stays outside the head.

Instead of using a chain of rigid links for the remaining part of the skeleton, as in most previous approaches [3, 7, 2, 16], we use a chain of point-masses linked by linear damped springs, thus modelling wisps of varying lengths. This allows to simulate wavy and curly hair, whose wisps can stretch, as well as almost straight hair. However, we should keep in mind that using a rigid sticks animation, although less general with respect to the kind of hair that can be modelled, would have optimized the computations for straight, and therefore completely inextensible wisps.

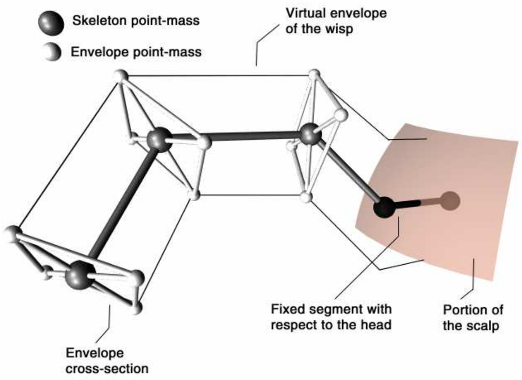

Fig. 1. The elements defining the skeleton and the envelope of a wisp, and their configuration.

In addition to the spring forces along the wisp axis, forces are introduced to simulate a certain resistance to bending, depending of the wisp's thickness. This is achieved with a damped spring for which the force increases linearly with the bending angle. The general configuration of the elements of a skeleton is illustrated in Figure 1.

3.3 Wisp Envelope

The second layer in our model is aimed at modelling radial wisp deformations around the skeleton. Thanks to this layer, the tip of a single swinging wisp will adequately squash and stretch during motion, due to the action of gravity and of air friction forces. Wisp sections will also be adequately compressed during interactions with other wisps or with the character's model.

Since different wisp deformations will occur along the skeleton, we discretize the wisp envelope into a given number of *cross-sections*, each of them being associated with one of the skeleton point-masses. The cross-section lies in the plane bisecting the two skeleton segments adjacent to that point-mass, similarly to what was done for defining the skin envelope control cross-section by Shen and Thalmann [15] (see Figure 1).

A coherent set of 2D frames for the cross sections along a wisp skeleton is needed to position the points modelling the wisp envelope. While designing the wisp skeleton model, we chose to rely on point dynamics only, so such local frames are not provided by the skeleton structure. However, a simple method can be used for recursively generating these frames since we are not modelling twisting effetcs: we attach the first local frame, associated with the first point of the wisp skeleton, to the head. Then, we successively define the other local frames using the quaternion that minimally rotates a given cross section plane onto the next one. This solution completely avoids twisting effects when a wisp moves, which is, as stated earlier, a desirable effect.

The wisp envelope models *radial deformations* only. We model each cross-section using a number of *envelope point-masses*, whose motion is restricted to a given half line of the cross-section plane originating at the skeleton point-mass. Each envelope point-mass thus controls the deformation of the envelope at a constant position over its surface. The envelope point-masses are much lighter than the associated skeleton point-mass, since they represent a small number of strands located near the envelope.

They are linked to the associated skeleton point-mass by non-linear damped springs that model the way the wisp resists to compression, and, to a lower extend, resists to stretching (the latter is due to high friction forces between the hair strands within a wisp). The choice of non-linear springs allows to model the fact that a wisp becomes progressively stiffer during a compression. Envelope point-masses of the same cross section are also linked together by soft, highly damped springs. This ensures a certain conservation of volume, encouraging the wisp to stretch in one radial direction if it is compressed in the other. Figure 1 illustrates the structures of a wisp.

3.4 Hair Strands

A third layer is used for rendering the individual hair strands within each wisp. The idea is to draw hair strand curves at fixed postions with respect with the wisp envelope. The strands will thus move and deform according to the wisp's current local deformations, while ensuring temporal coherence.

The large number of hair strands prevents us from storing much related data for each strand. We rely instead on the pseudo-random number generator for defining the hair stand position inside a wisp while ensuring the required frame-to-frame coherence. Because the number of random values required for modelling the hair is always the same, we can reset the seed of the random number generator at the beginning of every frame, thus providing the necessary coherence. Details on hair strand generation are given in Section 5.

4 Anisotropic Interactions

The idea of clustering hair strands into a number of wisps greatly simplifies interaction processing. In addition to modelling the effect of friction forces between neighboring hair strands of similar orientations, wisps provide an adequate level of detail for approximating other self-interactions inside the hair, and for computing interactions between hair and the character body (or any other obstacle). Since direct modification of velocities and positions has proven useful for processing the interactions of very light material such as cloth [4, 8, 18], we use similar solutions for hair.

4.1 Interactions between Wisps

Interaction detection relies on a standard 3D grid data structure for quickly determining a shorter list of wisp segments susceptible to intersect at a given time step. A mailbox parameter [1], which indicates the last time step when a given pair of such segments has been tested, ensures that each pair is tested only once. Collision detection between two wisp segments is achieved first by comparing their bounding boxes. If they intersect, a collision is detected when at least one of the skeleton or envelope point-masses of a wisp segment penetrates inside the volume of the other one. This volume is defined by its two cross sections.[1] Wisps self-collisions are handled exactly the same way than collisions between two different wisps of hair.

Wisps are highly anisotropic, since they are just a virtual representation for a group of hair strands. While two perpendicular colliding wisps should be compressed in order to avoid intersection, interpenetration has to be allowed between neighboring wisps

[1] Because our wisps segments are mostly as wide as they are long, this simple and efficient intersection scheme has appeared sufficient.

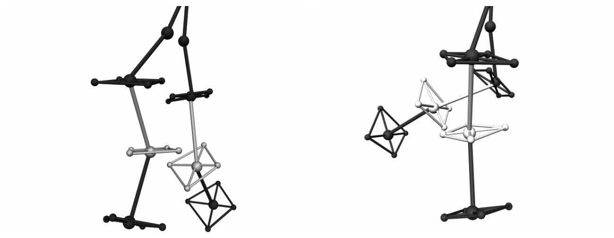

Fig. 2. On the left, the two wisps will interpenetrate, producing viscous friction, while on the right, whey will collide.

of similar orientations and velocities. Another major feature of hair self-interactions is that they are highly dissipative, since the extreme lightness of hair strands causes friction forces to play a very important role. The way we respond to wisp interactions is described next.

Viscous friction between quasi-parallel wisps. If the orientations of the two wisp segments lie in the same plane, and if their relative velocity is also roughly in this plane, wisps are allowed to interpenetrate (see Figure 2). Since the hair strands of the wisps should not go through each other during interpenetration, we eliminate the residual relative velocity lying outside the plane. This is done with a velocity modification.

Suppose a point-mass at position $\mathbf{p_p}$, moving at velocity $\mathbf{v_p}$, has penetrated in a wisp segment whose velocity is $\mathbf{v_s}$ at this same position $\mathbf{p_p}$. Let $\sigma_\mathbf{p}$ and $\sigma_\mathbf{s}$ be the directions of the two wisp skeletons near $\mathbf{p_p}$. The direction $\sigma_\mathbf{p} \times \sigma_\mathbf{s}$ is normal to the plane containing both wisps, and the relative velocity to eliminate is $(\mathbf{v_p} - \mathbf{v_s}) \cdot (\sigma_\mathbf{p} \times \sigma_\mathbf{s})$. We do this by applying to the point-mass the velocity modification:

$$\Delta \mathbf{v_p} = -\left(\frac{(\mathbf{v_p} - \mathbf{v_s}) \cdot (\sigma_\mathbf{p} \times \sigma_\mathbf{s})}{2} \right) (\sigma_\mathbf{p} \times \sigma_\mathbf{s})$$

Since similar modifications will be applied to point-masses of the other wisp at the same time step, this operation sets the velocities of the two wisps outside the plane that contains them, to the average of their initial values.

In addition, the relative velocity of the two wisps in the plane that contains them is submitted to a viscous drag, approximating the friction of the strands against each other. This drag is again implemented through a direct velocity modification, proportional to the hair strand density at position $\mathbf{p_p}$ and to the time step.

Collision between wisps of different orientations. If viscous friction conditions are not met, a very dissipative collision is modelled. We simply eliminate the relative velocities of the point-masses located in the contact area. Similarly to the viscous friction case, we use velocity modifications to set the velocities to the average of their initial values.

4.2 Collisions with the Character

The 3D grid data structure used for optimizing collision detection between wisps is also used to reference the polygons of the character for quickly detecting if a point-mass is close to the character.

Sliding Contact. If a point-mass moves closer to the character's surface than a user-specified threshold, then the part of the relative velocity approching the point-mass to the surface is eliminated. A viscous drag is added between the tangential velocity of the wisp point-mass with respect to the surface and the velocity of the polygon, in order to model viscous friction.

Penetration Reaction. If the character moves fast, the sliding contact mechanism may not be sufficient for preventing some wisp point-masses from penetrating inside the model. Detecting these points is simplified by the specific order in which we process each wisp: A wisp is traversed from root to tip, each skeleton point-mass being processed before its associated envelope point-masses. Since the point-mass at the top of the wisp skeleton is guaranteed to be outside the character, any intersection between a polygon of the character and a radial or axial spring of the wisp model means that the point-mass at the extremity of the current spring has moved into the character.

If collision occurs for a skeleton point-mass, it is moved out from the surface at a distance slightly under the sliding contact threshold. If penetration occurs for an envelope point-mass, it is moved out to the surface of the character along the half line it is constrained to lie on. The sliding contact mechanism is then engaged with the nearest polygon, resulting in a completely inelastic collision.

5 Implementation and Results

5.1 Algorithm for the Animation

As stressed in Section 3, we would like to take into account the effect of local thickness variations of the wisp envelope on subsequent wisp skeleton motion. So rather than decoupling the animation of the two sub-models, we animate all the point-masses within the same loop, enabling an envelope point-mass and a skeleton point-mass to interact through the radial damped spring that connects them.

Due to our interaction processing mechanism, constraints yielding velocity and position modifications have to be allowed. We thus use the following algorithm for the animation:

1. Compute the set of applied forces; these forces include gravity, air resistance, and forces generated by springs;
2. Detect interactions between pairs of wisps, and between wisps and the character's model;
3. Process velocities, using the current velocity values and applied forces; then apply the velocity modifications;
4. Process positions, using the current position and the new velocity; then apply the position modifications.

At each step, all point-masses are processed before going to the next step. A given wisp of hair is processed from root to tip, the envelope point-masses being processed after their associated skeleton point-mass. For envelope points, only forces, velocity modifications and position modifications projected onto the permitted axis, defined in the newly computed cross-section plane, are allowed.

Fig. 3. Our model (left) captures both continuities and discontinuities that can be observed in real long hair motion.

5.2 Individual Hair Strand Generation

In practice, we use only four envelope mass-points for modelling the cross-sections of wisps. A cross-section defines a 2D coordinate system whose origin is the skeleton point-mass, and the four envelope point-masses determine the orthogonal axes and s-caling factors. A set of 2D points, one per hair strand in the wisp, are generated in a circle according to a given distribution. This set is translated and scaled in 3D for each cross-section in the wisp. The 3D points resulting from a 2D point from this set are then linked to form the control points of a Catmull-Rom piecewise cubic curve defining a hair strand. The first and last control points are duplicated so the curve goes through all the control points. In order to prevent any pattern from appearing in the strands of a wisp, a user-specified jittering is applied to each control point, moving it towards its skeleton point-mass. Jittering is also applied to reduce the total length of the strand. Any strand therefore always remains within the wisp envelope.

Wavy hair can be obtained by specifying the number of hair-strand waves as well as their amplitude when the wisp's radius is at rest. Then, if a wisp segment stretches due to gravity for example, the amplitude is scaled down. The waves frequency is scaled too so that the number of waves in the segment is maintained.

5.3 Results

Our results are shown at www-imagis.imag.fr/Membres/Marie-Paule.Cani/hair.html Figure 5.3 (right) shows a frame of a motion tracking session. It illustrates the complex nature of hair motion, where wisps are easily observed. The data acquired from this session was then applied on a synthetic character with three different hair lengths (short, medium, and long). The short hair style is composed of 199 wisps with 4.0 segments per wisp. The medium hair style is composed, of 73 wisps with 9.9 segment per wisp in average. The long hair style, of 56 wisps with 7.7 segments. In the three hair styles, from which a frame is displayed in Figure 4, wisps are usually fairly wide. On average, a wisp segment intersects 17.5 other segments in the short hair style, 21.0 in the medium hair style, and 12.8 in the long hair style.

All three simulations, available from our web site, were computed at 1000 Hz using standard explicit integration, but collision detection was computed at 120 Hz, and wisps self-interactions at 24 Hz. Even with this choice, 92 to 95% of the total computation time was spent on detecting wisps self-interactions (64 to 70% of this percentage) and

Fig. 4. Results with various hair lengths.

wisps collisions with the character (30 to 36%). The total animation times[2] (264 frames) were 3.2 hours for the short hair style and 2.9 hours for the medium and the long hair styles.

These statistics confirm that the complete treatment of wisp interactions is the most significant portion of the total computation time in our simulation of hair motion. Avoiding the simulation at 1000 Hz thanks to implicit integration would probably improve performances, but not improve them as much, since the collision processing cost would remain.

6 Conclusion

Most previous models for hair animation have focussed on animating straight light hair, and resulted into very continuous shapes since almost no interactions were modeled. We have presented an alternative approach that precisely models the deformations and discontinuities due to the multiple interactions that occur in long, possibly wavy, hair.

The main feature of our model is to cluster hair into a number of deformable volumes representing hair wisps. We use a layered model including a skeleton and a deformable envelope for animating each of these wisps. The skeleton captures the global motion and the axial elongation of a wisp, while the envelope models the local radial deformations of its cross-sections. At each animation step, the two layers exchange informations since the envelope, which is positioned around the current position of the skeleton, is used to process interactions.

The method currently handles both straight and wavy hair. Similarly, curly hair could be modelled using a static twisting of the hair strands drawn inside each wisp.

We have seen that our model neglects the effects of dynamic twisting of hair wisps. A more general hair model would be obtained by replacing the standard point-masses used in the wisp skeleton model by a set of oriented particles [13], defining local coordinates systems in which the wisp cross-sections would be defined. This would allow to model a wisp that twists along its axis, which is currently impossible with our model. Moreover, the specific "shape-memory" forces in oriented-particle systems would allow the modelling of different hair styles. However, this solution would increase the computational time.

[2] All statistics were acquired on an SGI O2 with one R12000 processor at 270 MHz.

Acknowledgments

We would like to thank Arash Habibi of Université Louis Pasteur, for his help in the early stages of this work. This research has been made possible through an NSERC Industry Partnership Grant with Taarna Studios, inc. It also benefited from a grant from the "Centre Jacques Cartier".

References

1. J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics '87*, pages 3–10, August 1987.
2. M. Ando and S. Morishima. Expression and motion control of hair using fast collision detection methods. In *Image Analysis Applications and Computer Graphics. Third International Computer Science Conference. ICSC '95 proceedings*, pages 463–470, 1995.
3. K. Anjyo, Y. Usami, and T. Kurihara. A simple method for extracting the natural beauty of hair. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 111–120, July 1992.
4. D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, July 1998.
5. M.-P. Cani-Gascuel, A. Verroust, and C. Puech. Animation and collisions between complex deformable bodies. In *Graphics Interface '91*, pages 263–270, June 1991.
6. J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 243–252, July 1989.
7. A. Daldegan, N. Magnenat-Thalmann, T. Kurihara, and D. Thalmann. An integrated system for modeling, animating and rendering hair. In *Eurographics '93*, volume 12, pages 211–221, 1993.
8. M. Desbrun, P. Schroeder, and A. Barr. Interactive animation of structured deformable objects. In *Graphics Interface '99*, pages 1–8, June 1999.
9. S. Hadap and N. Magnenat-Thalmann. Interactive hair styler based on fluid flow. In *Computer Animation and Simulation '00*, pages 87–100, August 2000.
10. S Hadap and N. Magnenat-Thalmann. Modeling dynamic hair as a continuum. 2001. to appear in Eurographics 2001.
11. T-Y. Kim and U. Neumann. A thin shell volume for modelling human hair. In *Computer Animation '00*, pages 121 – 128, May 2000.
12. C. Koh and Z. Huang. Real-time animation of human hair modeled in strips. In *Computer Animation and Simulation '00*, pages 101–112, August 2000.
13. J.-C. Lombardo and C. Puech. Oriented particles: A tool for shape memory objects modelling. In *Graphics Interface '95*, pages 255–262, 1995.
14. R.E. Rosenblum, W.E. Carlson, and E. Tripp. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, 1991.
15. J. Shen and D. Thalmann. Interactive shape design using metaballs and splines. In *Implicit Surfaces '95*, pages 187–196, April 1995.
16. Z.-C. Shih and H.-D. Guo. The modeling and animation of human hair. *The Journal of Information Science and Engineering*, 11(3):465–488, 1995.
17. D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8(6):41–51, November 1988.
18. P. Volino and N. Magnenat-Thalmann. Accurate collision response on polygonal meshes. In *Computer Animation '00*, pages 179–188, May 2000.
19. Y. Watanabe and Y. Suenaga. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics and Applications*, 12(1):47–53, January 1992.