Hair Meshes

Cem Yuksel* Cyber Radiance
Scott Schaefer† Texas A&M University
John Keyser‡ Texas A&M University

Figure 1: *An example hair mesh model and the final hair model generated using this hair mesh and procedural styling operations.*

Abstract

Despite the visual importance of hair and the attention paid to hair modeling in the graphics research, modeling realistic hair still remains a very challenging task that can be performed by very few artists. In this paper we present *hair meshes*, a new method for modeling hair that aims to bring hair modeling as close as possible to modeling polygonal surfaces. This new approach provides artists with direct control of the overall shape of the hair, giving them the ability to model the exact hair shape they desire. We use the hair mesh structure for modeling the hair volume with topological constraints that allow us to automatically and uniquely trace the path of individual hair strands through this volume. We also define a set of topological operations for creating hair meshes that maintain these constraints. Furthermore, we provide a method for hiding the volumetric structure of the hair mesh from the end user, thus allowing artists to concentrate on manipulating the outer surface of the hair as a polygonal surface. We explain and show examples of how hair meshes can be used to generate individual hair strands for a wide variety of realistic hair styles.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations;

Keywords: Hair modeling, hair mesh, volume modeling

*e-mail: cem@cemyuksel.com
†e-mail: schaefer@cse.tamu.edu
‡e-mail: keyser@cse.tamu.edu

1 Introduction

Hair is an extremely important visual component of virtual characters. Therefore, it is crucial to equip artists with powerful tools that can help them sculpt the exact hair model they desire. Unfortunately, realistic hair models may require hundreds of thousands of hair strands, formed into exceptionally complicated geometric structures. The characteristics of individual hairs, the styling products applied to hairs, and the physical forces affecting the hairs all impact the overall look. Hence, a hair modeling tool should be powerful enough to handle a wide range of hair styles, simple enough that the tremendous complexity of the model is hidden from the user, and controllable such that artists can easily express their desired outcome.

Despite a considerable amount of research and a variety of implementations over the past two decades, hair modeling still remains an open challenge; there is no solution that is widely accepted in the graphics industry. To reduce the complexity of hair modeling, almost all existing approaches generate fine details of the hair model through procedural techniques. These procedural tools relieve the burden of dealing with every individual hair strand, allowing artists to concentrate on the overall look of the hair model. Thus, the main modeling effort on the part of the artist is in defining the global shape of the hair.

Even though hair is made up of many thin strands, we often interpret hair models as a surface. Therefore, the shape of this outer surface is important when modeling a particular hair style. Existing hair modeling approaches either define the shape of the hair model indirectly through various parameters or concentrate on the shapes of individual hair strands or bundles. In either case, the outer surface of the hair model is not explicitly defined. For this reason many skilled artists first model the outer surface with standard surface modeling tools and then use this surface as a guide while modeling hairs, attempting to place hairs in positions that will match that surface. This indirect control can be rather time consuming, especially when an artist desires to change the shape of the hair surface later.

In this paper we present a new alternative, *hair meshes*, that aims to bring hair modeling as close as possible to modeling polygonal meshes. Figure 1 shows an example hair mesh and the hair model created using this hair mesh. A hair mesh represents the entire volume of hair with topological constraints that allow us to easily trace the path of hairs from the scalp through the volume. The user has the ability to explicitly control the topological connections within the hair mesh, thus allowing creation of a wide range of possible hair models. A user will typically only interact with the external surface of the mesh volume, using this surface to explicitly control the shape of the hair. Internal vertices of the mesh volume are automatically placed based on the external surface. Since internal vertices do not need to be manipulated directly, artists who are already skilled at polygonal surface modeling can easily model hair with the flexibility and direct control of polygonal structures.

2 Related Work

There is a large body of previous research on virtual hair. In this section we briefly overview most related methods. We recommend the reader refer to Ward et al. [2007a] for a recent, extensive survey of hair methods in Computer Graphics.

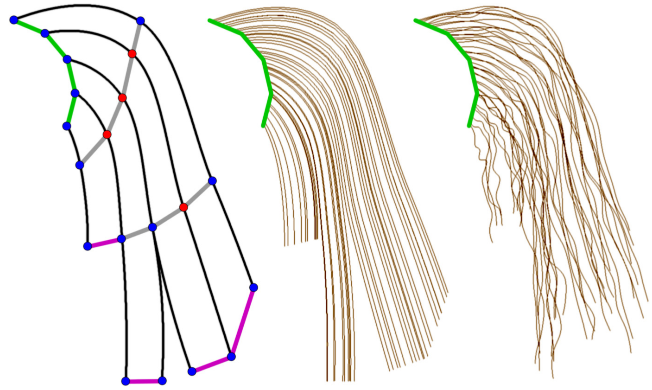
The high geometric complexity of hair and the wide variety of real-world hair styles make hair modeling a challenging task. Therefore, most hair modeling techniques are based on controlling collections of hair strands at once. Perhaps the simplest approach to hair modeling is representing hairs as parametric surfaces (e.g. NURBS) called *strips* [Koh and Huang 2001; Liang and Huang 2003; Noble and Tang 2004]. Using texture mapping with alpha channels, these surfaces look like a flat group of hair strands. Even though these techniques can be improved by adding thickness to this surface [Kim and Neumann 2000], they are very limited in terms of the models these methods can represent and are not suitable for realistic hair modeling.

A common hair modeling technique is to use *wisps* or *generalized cylinders* to control mostly cylindrical bundles of hairs with 3D curves [Chen et al. 1999; Yang et al. 2000; Xu and Yang 2001]. While these approaches are especially good at modeling hair styles with well defined clusters, it is often difficult and time consuming to shape a collection of wisp curves. Even making simple changes to an existing hair model can be exhausting depending on the number of curves to be edited. Multi-resolution approaches [Kim and Neumann 2002; Wang and Yang 2004] can improve the modeling process, yet this improvement depends on the complexity of the hair style and how close the desired hairstyle is to the types supported in the system.

Researchers have also tried using different physically-based techniques to shape hair strands. Anjyo et al. [1992] simulated the effect of gravity to find the rest poses of hair strands. Hadap and Magnenat-Thalmann [2000] modeled hairs as streamlines from a fluid dynamics simulation around the head. Yu [2001] used 3D vector fields to shape hairs by placing vector field primitives, and Choe and Ko [2005] applied vector fields with constraints to shape wisps. While various hair types can be modeled with these approaches, just like other simulation methods, they can be difficult to control in a precise manner.

Capturing a hair model from images [Kong et al. 1997; Grabli et al. 2002; Paris et al. 2004; Wei et al. 2005] is another alternative used to automate the virtual hair modeling process. Even though the recent methods [Paris et al. 2008] are very promising in terms of the visual realism of the results, these methods do not incorporate any artistic control.

Sketch based interfaces are also used for modeling hair [Malik 2005], both for cartoon hairstyles [Mao et al. 2005] and more realistic models [Wither et al. 2007]. Recently, Fu et al. [Fu et al. 2007] proposed a sketch based interface to build a vector field, which is then used to generate individual hair strands. While these techniques are practical for quickly generating a hairstyle, they are very difficult to control for achieving a desired outcome precisely.



(a) Hair Mesh (b) Generated Hairs (c) Styled Hairs

Figure 2: *(a) 2D representation of a hair mesh, (b) hairs generated from this hair mesh, and (c) hairs after procedural styling operations. The green lines correspond to faces of the root layer, the tip layer is colored purple. Blue hair mesh vertices are external, and red are internal.*

Other hair modeling approaches include explicitly modeling individual hair strands [Daldegan et al. 1993; Lee and Ko 2001] or a number of guide hairs [Alter 2004]. An interesting recent approach aims to simulate a real-world hair dressing session using haptic controls and physically-based simulation [Ward et al. 2007b]. Recently, Wang et al. [2009] proposed a method for generating a new hair model based on a given hair model.

3 Hair Mesh Modeling

In our approach, hair modeling begins by defining the outer surface of the hair model. This outer surface is used to create a meshed volume, the *hair mesh* (see Figure 2a). The hair mesh structure is then used to generate individual hair strands (Figure 2b). As in most existing techniques, fine details of the hair strands are subsequently defined through procedural styling operations (Figure 2c), which we will discuss at the end of this section.

3.1 The Hair Mesh Structure

A hair mesh is a 3D mesh describing the volume of space enclosing the hair. It consists of layers of polygonal meshes, which typically contain quadrilateral or triangular faces, but we place no restrictions on the types of polygons used. Let F^k be a set of faces for layer k. We refer to the faces in F^0 as the *root layer* of the hair mesh (highlighted green in Figure 2a) and the polygons in this layer exactly match the scalp model. This is the surface we will grow hairs from and each face in F^0 will correspond to a bundle of hairs.

To create a path for each hair, we place a number of additional layers on top of the root layer, such that each face F_j^k at layer k has a one-to-one correspondence to a face F_j^{k+1} at the next layer. Connecting the two corresponding faces F_j^k and F_j^{k+1}, we form a prism such that these faces are the two base faces of the prism. We refer to the collection of such prisms starting at F_j^0 and connecting to faces F_j^k (where $k \geq 0$) as a *bundle*, F_j. If for any face F_j^k the corresponding face F_j^{k+1} of the next layer does not exist, hair strands of this bundle terminate at layer k. We refer to the termination layer of bundle F_j as n_j and call the surface composed of all of the $F_j^{n_j}$ the

Figure 3: *A hair knot model and its hair mesh. The explicit control of the hair shape provided by hair meshes makes modeling such hairstyles as easy as modeling any other surface.*

tip layer, which is highlighted in purple in Figure 2a. Even though there is a one-to-one mapping between faces at different layers, the faces adjacent to a given face may change from one level to the next (e.g. the mesh can split, separating bundles as in Figure 2a).

Given this correspondence between layers, we can create a path for each hair from the root layer. For each point on the root layer, we compute the barycentric coordinates of that point with respect to the face F_j^0 containing it using mean value coordinates [Floater 2003]. We then trace the path of a hair growing from that point through the volume by applying these barycentric weights at every corresponding face in the bundle up to the tip layer. Finally, we connect these points together using C^1 Catmull-Rom splines [1974] to generate the final hair.

In the simplest case, all layers of a hair mesh have exactly the same topology. However, as illustrated earlier, this is not a requirement. The mesh is valid as long as each face has a corresponding face on all subsequent layers. Therefore, a *vertex* can be connected to one vertex (if the topology is locally the same) or to multiple vertices (if the topology changes) on a neighboring layer.

To keep the hair mesh structure simple, we permit only a one-to-many mapping of *vertices* from one layer to the next layer (note: faces are always a one-to-one mapping), though a many-to-many mapping of vertices can potentially generate a valid hair mesh. This simple restriction ensures many useful properties, such as edges of faces at one layer cannot collapse at the next layer, two faces at any layer can be neighbors (i.e. share a vertex) only if they are neighbors at the root layer, and the genus of the root layer is the same as that of the set of bundles.

3.2 Topological Operations

Users must be given controls that provide a wide variety of modeling operations, but at the same time these modeling operations must preserve the topological constraints on the mesh. The input to our system is a polygonal object that we would like to grow hairs on. This object forms the root layer F^0 of our hair mesh. In the beginning (when the hair mesh has no layers), the root and the tip layers coincide (i.e. $n_j = 0 \, \forall j$).

A user interacting with the mesh will typically model the hair by "growing" the layers out from the root layer, specifying geometric and topological changes in each layer. To perform this modeling, the following operations are supported for creating and modifying the hair meshes:

Face Extrude: This is our primary operation to create new layers. Face extrusions are only permitted from the current tip layer faces, as the extrusions of side faces would not generate valid hair meshes. For each face $F_j^{n_j}$ to be extruded, a new face $F_j^{n_j+1}$ is created, thereby generating a new prism in the hair mesh. This is typically the very first operation we use to extend the root layer.

Face Delete: This operation deletes the face at the tip layer of a bundle, thereby removing the last prism. When the root and the tip layers coincide ($n_j = 0$), a face delete operation is equivalent to deleting a particular face from the root (and thus no hairs will be created for that region).

Layer Insert: We use this operation to create new layers in between two intermediate layers. Though layer insert could be defined as a local operation, to enforce one-to-many mapping of vertices we perform the same operation on all faces *that are topologically connected* in the layer at which the operation is applied. The new layer is inserted before the layer that is selected.

Layer Remove: Similar to layer insertion, layer removal affects all the topologically connected faces of a layer. When the layer to be removed is the tip layer, this operation is identical to face delete(s). The root layer cannot be removed as it would mean deleting the root object.

Edge and Vertex Separate: Vertices and edges shared by more than one face in a single layer can be topologically separated. This topological separation creates multiple edges/vertices that are topologically separated but are geometrically coincident. Subsequent modeling operations may move these points geometrically. If the separated vertex or edge is not at the tip layer, all corresponding vertices or edges above this layer are also separated to ensure one-to-many mapping of vertices.

Edge and Vertex Weld: The weld operation is the inverse of a separate operation, and topologically joins the vertices or edges at the same layer. To respect one-to-many mapping of vertices, this operation can only weld vertices that correspond to the same vertex at the root layer. Furthermore, all corresponding vertices below this layer are also welded.

Face and Edge Divide and Subdivision: Splitting and subdivision of faces can be easily defined over the hair mesh. This includes standard approaches such as Catmull-Clark or Loop subdivision. Any subdivision operation applied to a face must be propagated throughout the entire bundle for that face. In addition, since subdivision may modify adjacent faces, all bundles adjacent to that bundle in the root layer may also be affected. Note that subdivision is supported only on the layers of the hair mesh, not on the quadrilateral faces that form the sides of the prisms in the hair mesh surface (layer insertion provides a similar effect, there).

3.3 Geometrical Operations

The vertices of the hair mesh are described as either external vertices, which lie on the outer surface of the mesh, or internal vertices. This classification is illustrated in Figure 2a, where external vertices are drawn in blue, and internal vertices in red. Note that several

Figure 4: *A simple hair bun modeled using hair meshes.*

topological operations can generate new vertices (both external and internal), or convert vertices between external and internal vertices.

In general, the user can explicitly position all these vertices. However, a large number of internal vertices may be generated during construction of the hair mesh. Since the number of external vertices is proportional to surface area and the interior to volume, the number of internal vertices may dominate the total number of vertices in complex hair models. These vertices are necessary to determine the path of a hair and provide adjacency information for hair bundles. However, these internal vertices are problematic for the user because they lie inside the enclosed volume of the hair mesh, making them hard to see, especially when the hair mesh is visualized as a surface. Therefore, we provide the option to hide these internal vertices and instead place them automatically based on the positions of the external vertices.

Internal vertex placement is a part of the modeling process and is executed every time the user moves or creates a group of external vertices. Some of these operations (moving external vertices) can affect a large number of internal vertices. For this reason, it is critical that the internal vertex placement algorithm be performed very quickly, so as not to interrupt the modeling process.

While many techniques may be used to place internal vertices, we choose a simple constrained quadratic minimization. We will define an error metric in terms of the positions of all mesh vertices, fix the external vertices, and solve for the positions of internal vertices that minimize this metric. Our choice of error metric is motivated by physical properties of hair. Namely, hair strands should have a similar shape as nearby hair strands (later we will apply styling operations to differentiate nearby hairs as explained in Section 3.4).

For each extruded prism between faces F_j^k and F_j^{k+1}, we can approximate the hair direction locally using the edges of the prism in the extrusion direction. Let $V_{j,i}^k$ be the i^{th} vertex of the face F_j^k. Then, an edge of the prism is given by the vertices $V_{j,i}^k$ and $V_{j,i}^{k+1}$, and the hair direction locally along the edge is $V_{j,i}^{k+1} - V_{j,i}^k$. We want to minimize the difference in the local hair direction between adjacent edges along the extrusion direction of the prism.

If we sum over all of the quad faces that form the sides of the prisms in the volume, the resulting minimization is of the form

$$\min \sum_j \sum_{k=1}^{n_j} \sum_i \left\| (V_{j,i}^{k+1} - V_{j,i}^k) - (V_{j,i+1}^{k+1} - V_{j,i+1}^k) \right\|^2 ,$$

subject to the constraint that the external vertices (or any others the user wishes to fix) remain unchanged. We can easily minimize this quadratic, which has a unique minimum, using Conjugate Gradients [Shewchuk 1994]. Since we use the current positions of the internal vertices as a starting point for this minimization, Conjugate Gradients typically converges to a solution in only a few iterations and is quite fast. The red vertices in Figure 2a show the result of a 2D version of this optimization with the blue, external vertices as constraints.

Figure 5: *A futuristic hair bun model and its hair mesh.*

3.4 Hair Styling

Hair mesh modeling can be thought of as an initial stage of modeling hair. The hair mesh defines the overall shape of the hair model and the hair strands we generate conform to that model. However, realistic hair is not always straight and many existing hair modeling techniques can be applied to the hair strands to improve the realism of the hair or reproduce specific hair styles. In our system, all hair modeling operations applied to the hair strands after they are generated from the hair mesh are called *styling* operations.

Procedural hair styling forms one group of such operations. These operations typically deform the hair by moving the vertices of the hair strands using a combination of procedural noise and trigonometric functions with various parameters. The functions can be directly computed using the 3D position of each hair strand vertex. However, this makes the hair style, which is applied using these procedural operations, very sensitive to the initial 3D positions of hair strand vertices. As a result, even minor modifications to the hair mesh may significantly alter the shapes of some hair strands. To avoid this undesired behavior, one can define these procedural operations in the canonical space of a hair strand as in [Yu 2001] or using the barycentric embedding of a hair strand within the hair mesh. In our system we use the later approach.

In addition to procedural operations, we can easily combine our hair modeling system with some previous hair modeling techniques that use wisps. We achieve this by generating wisp curves from the hair mesh similar to generating hair strand curves. In this case, individual hair strands are not directly generated from the hair mesh, but the wisp curves along with a number of parameters are used to populate final hair strands. Note that wisp curves themselves can go through procedural styling operations or explicit user modifications before generating the hair strands. Similarly, we can combine our hair mesh modeling approach with multiresolution hair modeling [Kim and Neumann 2002] by generating first level generalized cylinders using the hair mesh. Higher level generalized cylinders and finally individual hair strands are then generated from the first level generalized cylinders as described in [Kim and Neumann 2002]. This approach replaces the most laborious stage of multiresolution hair modeling (as stated by Kim and Neumann [2002]) with hair mesh modeling.

4 Results and Discussion

To demonstrate the capabilities of our hair mesh modeling approach we present various hair models produced using our system. Figure 1 shows a typical hair model with its hair mesh. While similar hair models can be prepared with many previous techniques, the main advantage of the hair mesh is the ability to control the hair shape by directly manipulating the outer surface.

Figure 6: *A complicated hair bun model and its hair mesh.*

Figures 3, 4, and 5 show different hair models with buns and knots. Such models are very difficult to prepare with most previous techniques, but with hair meshes, modeling these hairstyles are no more difficult than modeling the outer surface using any standard polygonal modeling tool. Figure 6 shows a more complicated bun model. Notice the fine detail of the bun and the explicit control of the hair shape and direction available to the artist.

Depending on the complexity of the desired hair model, modeling using hair meshes can take as short a time as a couple of minutes. For example, the simple hair model shown in Figure 7 can be prepared in a couple of minutes. While preparing a more complicated hair model can take significantly longer, the explicit control provided by hair meshes makes it easy to edit the model and produce the desired variation.

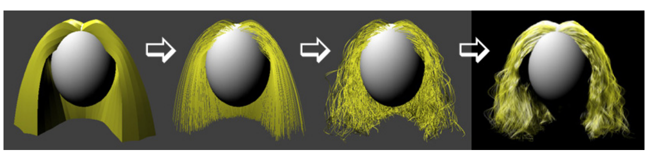


Figure 7: *A simple hair model prepared within a couple of minutes.*

Even though hair meshes are designed to model hair by letting the user specify only the outer surface of the hair, they are also useful when an artist desires to control the hair shape within the hair volume. Figure 8 shows such a complicated hair mesh model where the artist explicitly shapes each hair bundle.

Figure 9 shows an example of combining hair mesh modeling with wisp based hair modeling. Here the hair mesh in Figure 1 is used to generate 200 wisp curves, instead of individual hair strands. Final hair strands are then generated from these wisp curves as in Choe and Ko [2005].

One useful property of our hair mesh modeling approach is the complete separation of large and small scale details. While large scale details that define the global shape of the hair model are controlled using the hair mesh, fine details are introduced during the styling process. Therefore, the same hair mesh can be used to generate different types of hair styles as shown in Figure 10. This technique can also be used for introducing significant style variations within a hair model by generating one set of hair strands from a hair mesh and applying one style, then using the same hair mesh to generate another set of hair strands with a different style applied. By generating multiple sets of hairs with this procedure, style

Figure 8: *A complicated hair mesh model and the hair generated from this hair mesh.*

variations of a hair model can be easily represented. The union of these sets form the final hair model, and the global shape of the hair model is explicitly controlled by a single hair mesh. This feature is especially useful when modeling realistic hairs with rich variations such as frizzy hair and fly-aways. Figure 11 shows such an example. Note that in many previous hair modeling techniques, introducing these frizzy strands can be difficult or even impossible.

We designed the hair mesh modeling approach such that styling operations are reserved for small scale details only and larger details are explicitly modeled using the hair mesh. However, in our system there is no restriction on the user side to forbid using styling operations for large variations as well. When the style variations are exaggerated, the perceived surface of hair formed by the final hair strands can deviate from the surface defined by the hair mesh. This deviation is especially undesirable when the hair mesh is used for explicitly avoiding intersections of hairs with surrounding objects. Note that undesired intersections can also be automatically avoided at the hair strand level using the technique described by Kim and Neumann [2002].

Hair mesh modeling merely provides a high level structure to easily define the global shape of a hair model. Unfortunately, it is not possible to claim for any hair modeling technique that it can produce hair models that cannot be modeled using previous techniques, since theoretically speaking all hair models can be produced by ex-

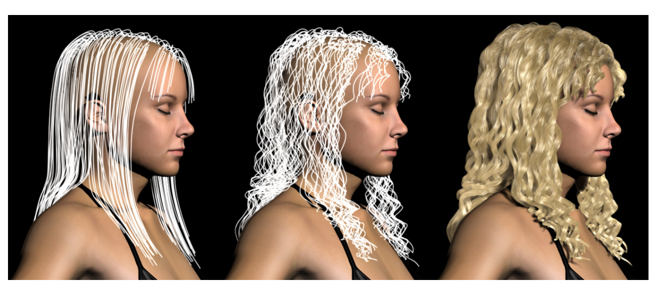


Figure 9: *(Left) wisp curves generated from a hair mesh, (middle) wisp curves after styling operations, and (right) final hair strands generated from these wisp curves.*

Figure 10: *A complicated hair mesh model and two different hairstyles generated from the same hair mesh via different styling operations.*

Figure 11: *Frizzy strands generated directly from the hair mesh as an additional hair group on top of the hairs from Figure 9. The close-up view on the right shows the effect of frizzy strands.*

plicitly modeling the hair strands. Moreover, the real power of hair meshes is not the fact that various different hair models can be prepared with this approach, either. Most existing techniques permit a wide variety of hair styles to be generated. However, the lack of explicit control over the global hair shape makes the existing techniques difficult, if not impossible, to use to achieve the exact hair model one aims for. On the other hand, hair meshes convert the volumetric hair modeling problem to a surface modeling problem. This significantly reduces the high complexity of volume modeling and brings hair modeling closer to standard polygonal surface modeling. As a result, hair meshes offer a familiar interface to experienced modelers and make it very easy for them to sculpt the exact hair models they desire.

We have also tried using hair meshes for simulating hair. We follow an approach similar to that introduced by Chang et al. [2002]. Instead of picking representative hair strands (i.e. guide hairs), we form an articulated rigid body chain directly from the edges of the hair mesh that connect the vertices of one layer F^k to the next layer F^{k+1}. Figure 12 shows example frames captured from our hair mesh simulation system. We observed that physical simulations using hair meshes can produce seemingly natural hair-hair interactions with high performance. The hair mesh in Figure 12 includes 30 rigid body links and 120 chains, and the simulation runs at 92 fps on a 2.14 GHz Intel Core 2 Duo processor with a single thread (note that such a simulation can be trivially multi-threaded).

5 Conclusions and Future Directions

We have introduced hair meshes for modeling hair using polygonal surface tools. Our technique allows an artist to create complex hair styles easily by providing explicit control over the overall shape of the hair surface. By automatically placing internal vertices, the artist can concentrate on the outer surface shape of the hair, which significantly simplifies the hair modeling process without limiting direct control over the hair model shape.

We believe there are more applications of hair meshes than just modeling. An example of using hair meshes for hair simulation is presented in the previous section. Furthermore, real-time rendering of deforming hair such as the animation produced by our hair simulation can be accelerated using hair meshes. Since the hair is completely determined by the geometry of the hair mesh, the hair geometry can be synthesized on the GPU simply by sending the deformed positions of the hair mesh vertices, thereby significantly saving graphics bus bandwidth. Hair meshes may also be used to approximate shadow and ambient occlusion computations.

Acknowledgements

We would like to thank Lee Perry-Smith (www.ir-ltd.net) for providing the models and producing most of the hairstyles in this paper. This work was supported in part by NSF grant CCF-07024099.

References

ALTER, J. S., 2004. Hair generation and other natural phenomena with surface derived control volumes in computer graphics and animation. U.S. Patent 6720962.

ANJYO, K., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 111–120.

CATMULL, E., AND ROM, R. J. 1974. A class of local interpolating splines. In *Computer Aided Geometric Design*, Academic Press, Orlando, FL, USA, 317–326.

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

CHEN, L.-H., SAEYOR, S., DOHI, H., AND ISHIZUKA, M. 1999. A system of 3d hair style synthesis based on the wisp model. *The Visual Computer 15*, 4, 159–170.

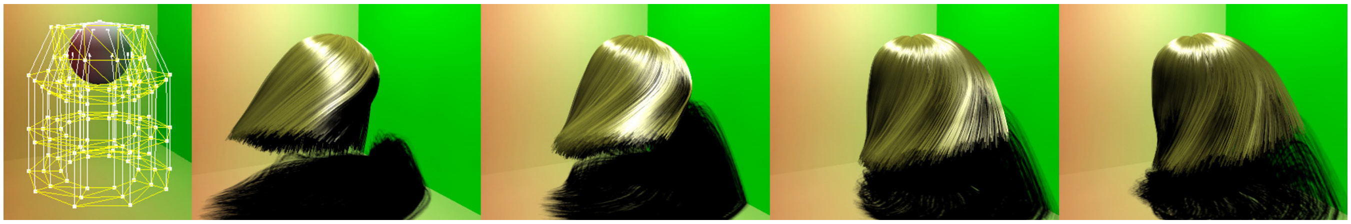


Figure 12: *Sample frames captured from our real-time hair mesh simulation system. The last frame shows the structure of the simulated hair mesh, which has 150 vertices in 5 layers. The simulation runs at 92 fps on a 2.14 GHz Intel Core 2 Duo processor with a single thread.*

CHOE, B., AND KO, H.-S. 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics 11*, 2, 160–170.

DALDEGAN, A., THALMANN, N. M., KURIHARA, T., AND THALMANN, D. 1993. An integrated system for modeling, animating and rendering hair. In *Eurographics '93*, Blackwell Publishers, Oxford, UK, R. J. Hubbold and R. Juan, Eds., Eurographics, 211–221.

FLOATER, M. S. 2003. Mean value coordinates. *Computer Aided Geometric Design 20*, 1, 19–27.

FU, H., WEI, Y., TAI, C.-L., AND QUAN, L. 2007. Sketching hairstyles. In *SBIM '07: Proceedings of the 4th Eurographics Workshop on Sketch Based Interfaces and Modeling*, ACM, New York, NY, USA, 31–36.

GRABLI, S., SILLION, F., MARSCHNER, S. R., AND LENGYEL, J. E. 2002. Image-based hair capture by inverse lighting. In *Proc. Graphics Interface*, 51–58.

HADAP, S., AND MAGNENAT-THALMANN, N. 2000. Interactive hair styler based on fluid flow. In *Eurographics Workshop on Computer Animation and Simulation 2000*, Springer, 87–99.

KIM, T.-Y., AND NEUMANN, U. 2000. A thin shell volume for modeling human hair. In *CA '00: Proceedings of the Computer Animation*, IEEE Computer Society, Washington, DC, USA, 104.

KIM, T.-Y., AND NEUMANN, U. 2002. Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2002) 21*, 3, 620–629.

KOH, C. K., AND HUANG, Z. 2001. A simple physics model to animate human hair modeled in 2d strips in real time. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 127–138.

KONG, W., TAKAHASHI, H., AND NAKAJIMA, M. 1997. Generation of 3d hair model from multiple pictures. In *Proceedings of Multimedia Modeling*, 183–196.

LEE, D. W., AND KO, H. S. 2001. Natural hairstyle modeling and animation. *Graphical Models 63*, 2, 67–85.

LIANG, W., AND HUANG, Z. 2003. An enhanced framework for real-time hair animation. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, 467.

MALIK, S. 2005. A sketching interface for modeling and editing hairstyles. In *SBIM '05: Proceedings of the 2nd Eurographics Workshop on Sketch Based Interfaces and Modeling*, 185–194.

MAO, X., ISOBE, S., ANJYO, K., AND IMAMIYA, A. 2005. Sketchy hairstyles. In *CGI '05: Proceedings of the Computer Graphics International 2005*, IEEE Computer Society, Washington, DC, USA, 142–147.

NOBLE, P., AND TANG, W. 2004. Modelling and animating cartoon hair with nurbs surfaces. In *CGI '04: Proceedings of the Computer Graphics International*, IEEE Computer Society, Washington, DC, USA, 60–67.

PARIS, S., HECTOR M. BRICE N., AND SILLION, F. X. 2004. Capture of hair geometry from multiple images. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2004) 23*, 3, 712–719.

PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2008) 27*, 3, Article 30.

SHEWCHUK, J. R. 1994. An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., Pittsburgh, PA, USA.

WANG, T., AND YANG, X. D. 2004. Hair design based on the hierarchical cluster hair model. *Geometric modeling: techniques, applications, systems and tools*, 330–359.

WANG, L., YU, Y., ZHOU, K., AND GUO, B. 2009. Example-based hair geometry synthesis. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2009) 28*, 3, Article 56.

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, 213–234.

WARD, K., GALOPPO, N., AND LIN, M. 2007. Interactive virtual hair salon. *Presence: Teleoperators and Virtual Environments 16*, 3, 237–251.

WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2005) 24*, 3, 816–820.

WITHER, J., BERTAILS, F., AND CANI, M.-P. 2007. Realistic hair from a sketch. In *International Conference on Shape Modeling and Applications*, IEEE, Lyon, France, IEEE, 33–42.

XU, Z., AND YANG, X. D. 2001. V-hairstudio: An interactive tool for hair design. *IEEE Computer Graphics and Applications 21*, 3, 36–43.

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

YU, Y. 2001. Modeling realistic virtual hairstyles. In *PG '01: Proc. of the 9th Pacific Conference on Comp. Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, 295.