Interactive Styling of Virtual Hair

Rui Zhang and Burkhard C. Wünsche

Department of Computer Science, Private Bag 92019, University of Auckland, New Zealand

Email: burkhard@cs.auckland.ac.nz

Abstract

Interactive styling of virtual hair is an important research field since it is essential for creating realistic looking human avatars for use in virtual worlds, computer games and movie special effects. Virtual hair models can contain thousands of hair strands and hence it is important to develop techniques which enable a designer to efficiently modify the hair in a realistic fashion. In this paper we present a hair styling toolset which uses wisps to represent basic units of hair strands and an improved statistical model for hair wisp generation. The toolset provides a convenient way for users to do operations such as create, edit, delete, copy and paste and hence facilitates the quick creation of hair styles while allowing sufficient control for adding individualistic styling details. The styling process is simplified by using a local coordinate system for hair strands in order to define preferred styling (brushing) directions.

Keywords: hair modelling, interaction techniques, spline curves, key strands

1 Introduction

Computer generated realistic virtual humans are required in applications such as the movie industry (CGI – computer generated imagery), computer games, and as avatars for virtual worlds. An important factor for achieving a realistic human appearance is the development of a realistic hair model. Psychological studies have shown that hair is a determining factor of a person's first impression when meeting his or her counterpart [1]. Therefore, the styling of virtual hair is an active field of research in Computer Graphics. Hair styling is challenging since the complex behaviour of each hair strand and the interactions among the hair strands during animation and styling must be controlled in a physically realistic way.

In order to develop efficient styling tools it is important to discuss different approaches for modelling and rendering hair. Popular approaches to model hair are based on polygonal surfaces, noise-based approaches, volumetric textures, strand-based models, wisp-based models, particle models, and models based on fluid flows or vector fields.

Parke introduced a fast and simple way to model hair which uses simple texture mapped polygonal surfaces to capture the shape and appearance of hair [2]. Many applications nowadays still use this approach due to its simplicity. However, because the surface representation does not model the complex geometry of hair strands the specular lighting effects are not correct and the resulting rendered images lack realism.

Perlin proposed a way to synthesize images of visually complex objects including hair by hypertextures [3]. This noise-based modelling approach cannot capture the movement of individual hair strands. Hence this approach is most suitable for short hair where forces such as gravity and friction between hair strands are small such that the hair hardly moves.

In 1989 a volumetric texture hair model was proposed by Kajiya and Kay [4]. The authors introduced "Texels", which are 3-dimensional arrays of parameters approximating visual properties of a collection of micro surfaces. The most important parameter stored in a Texel is the tangent vector, used to calculate the light reflection by an anisotropic reflection model [4]. Instead of using geometries to model hair strands, Kajiya and Kay use texels to represent hair strands and map them onto the surface of a 3D object. The technique works well for short, furry hair since the corresponding texels are simple and can be generated automatically. However, it is not clear how texels can be easily generated for representing more complex hair styles and how this representation could be used to enable interactive styling.

Strand-based models represent every hair strand explicitly. Because of the large amount of hair strands on a head the strands are most frequently represented by connected line segments rather than connected cylinders in order to reduce the consumption of computing resources when modelling hair [5, 6]. This kind of model is suitable for modelling long animated hair strands with a simple style, but it is not practical for modelling complex hairstyles due to the large number of strands which must be moved. It is extremely difficult to achieve animation of complex hairstyles because of the complex behaviour and large number of hair strands. Modelling complex hairstyles using a strand-based model is therefore difficult to achieve in real-time.

The key idea of wisp-based models is to group hair strands into wisps and to define their shape and animation using so-called key strands. The idea is based on the observation that adjacent hair strands tend to form wisps due to static attraction and artificial styling products. Daldegan et al. model the underlying head using triangle meshes and use three key hair strands, one at each vertex of a triangle, to interpolate the hair strands of a wisp [7]. Yang et al. use generalized cylinders to represent hair wisps [8]. Plante et al. proposed an animation method to deal with the interactions among wisps and to simulate complex hair motions [9]. The above three models have the advantage that they make it easy to control hair styling. However, the methods are not effective for controlling complex hairstyles such as curly hair. In 2002 Kim and Neumann proposed a multi-resolution hair modelling system, which can handle fairly complex hairstyles [10]. The model makes it possible to define the behaviour of hair over the entire range from hair wisps down to individual hair strands. Different hair styles can be created rapidly using high-level editing tools such as curling, scaling and copy/paste operations. Subsequently Choe and Ko introduced a statistical wisp model to generate a wide range of human hairstyles [11]. The authors simulate hair deformation by applying physical properties of hair such as gravity and collisions detection and response. The model is capable of handling a wide range of human hair styles but is unsuitable for simulating hair animation due to a lack of real-time performance of their modelling algorithm and failing in collision detection in some cases.

Particles, fluid flow, and vector field models for hair were motivated by the observation that slightly curled hair and fluid flows have similar properties in terms of smoothness and continuity. Stam proposed a particle-based hair model, which simulates a hair strand as a trajectory of a particle shot from the head [12]. Hadap and Thalmann considered hair as fluid flow [13], and Yu proposed a hair model using user controllable vector fields [14]. These approaches provide users an easy way to define and modify simple hairstyles, but fail to handle complex ones.

In conclusion we can say that wisp-based models are the most flexible hair models. Additional advantages are their capacity to create a wide range of different hair styles, control details of a hair style, and support high-level operations such as copy/paste between wisps when designing a hair style. Disadvantages of this approach are the large amount of time needed to handle the interactions between hair strands such as collision detection and difficulties in simulating convincing hair animation. However, for many applications with less animation such as hair styling, these advantages outweigh the disadvantages, and we therefore will use a wisp-based model in our research.

2 A Toolset for Hair Styling

This section first introduces our hair model and then describes the main components of the hair styling toolset and its capabilities.

Our hair model is static and is based on the wisp-based model proposed by Choe and Ko [11]. In order to place our hair strands we use a head model represented by a high resolution triangle mesh. The head model with the scalp region coloured brown is shown in figure 1.

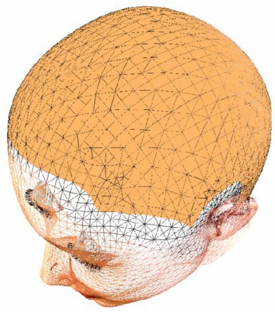


Figure 1: A head model represented by a triangle mesh. The scalp region is coloured brown.

Inspired by Kim and Neumann's interactive hair modelling system [10] we added tools to enable users to manipulate wisps. Hair strands can be grown on the scalp shown in figure 1. Since a scalp can have thousands or tens of thousands of hair strands we need a system to easily control groups of hair strands. The user is able to group several triangles on which to grow a wisp. The size of a wisp (its number of strands) is dependent on the number of selected triangles. The smallest wisp is defined by a single triangle. This level of detail is sufficient for defining a wide variety of hair styles since the triangles are small compared to the scalp's area.

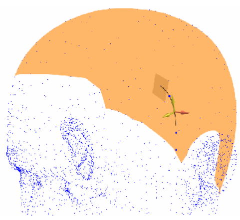


Figure 2: A key strand defining a wisp for a group of four triangles on the scalp. The dark area on the scalp defines the selected triangles. The blue square points are the control points on the key strand.

The geometry of a wisp is controlled using a so-called key strand. A key strand defines the geometry for all strands of a wisp which are then obtained by translating the key strand appropriately. The number of strands for a scalp region depends on the area of the triangles representing it. We use Catmull-Rom splines [15] to represent hair strands since they are smooth (C^1 continuous) and because they interpolate their control points which makes designing a particular hair style more intuitive. An example of a key strand for a group of four selected triangles of the scalp is shown in figure 2.

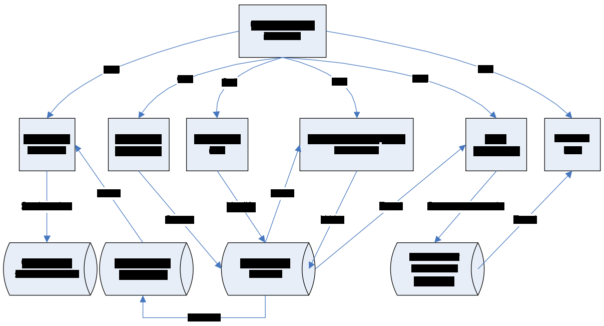


Figure 3: An overview of the components of our hair styling toolset.

The components of our hair styling toolset are illustrated in figure 3. The key strand selection component enables users to choose triangles to grow a wisp or to select an existing wisp for editing. After a group of triangles has been selected it is recorded together with the location of its current key strand. The location of the key strand is determined as the centre of the first triangle of the selected group of triangles (see section 3). The selection of scalp triangles and strand control points has been implemented with the OpenGL "select" mechanism. This enables us to detect whether the projection of a graphical primitive onto the view plane overlaps with a hit region surrounding the mouse location in which case we select the front most primitive.

The key strand generation component enables users to generate one key strand for a selected group of triangles. Users are able to interactively grow a key strand by adding new control points and to modify the 3D shape of a key strand by moving its control points.

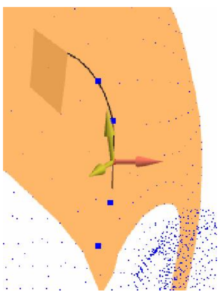


Figure 4: The user interface for changing the 3D positions of a key strand's control points. The red arrow indicates the currently active direction in which the control point can move forward and backward. The yellow arrows indicate the non-active directions.

Users can change the size of a wisp in the key strand edit component by changing the triangles defining the area on which the strands of this wisp grow. In addition users can delete an existing wisp and they can change the 3D shape of a wisp by moving, adding or deleting control points of the key hair strand.

The interface for changing the 3D coordinates of the control points of a key strand is illustrated in figure 4. Since the mouse movements on the screen are in 2D we have to map this into a suitable 3D motion. A common solution in modelling applications is to restrict movements to the coordinate directions, parallel to the view plane or within a user defined plane. We found that in hair styling the preferred hair movement direction depends on a particular style, e.g. "brushing" hair backwards, lifting it up, pulling it down or curling it. We therefore define for each key strand a local coordinate system of styling directions. The coordinate system is represented by three orthogonal arrows and the currently active styling direction is indicated by a red arrow. A new styling direction is obtained by choosing one of the non-active arrows or by changing the local coordinate systems as explained below. Suitable default directions for the local coordinate system at a control point are the curve tangent at that point, the surface normal at the scalp point closest to the control point and the vector perpendicular to these two vectors.

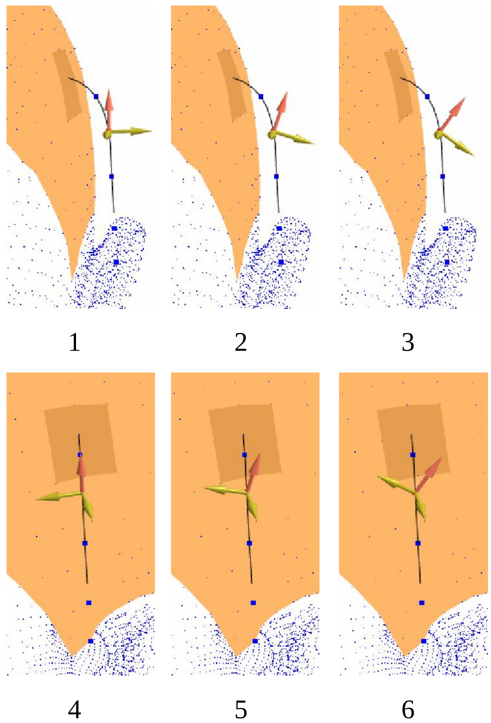
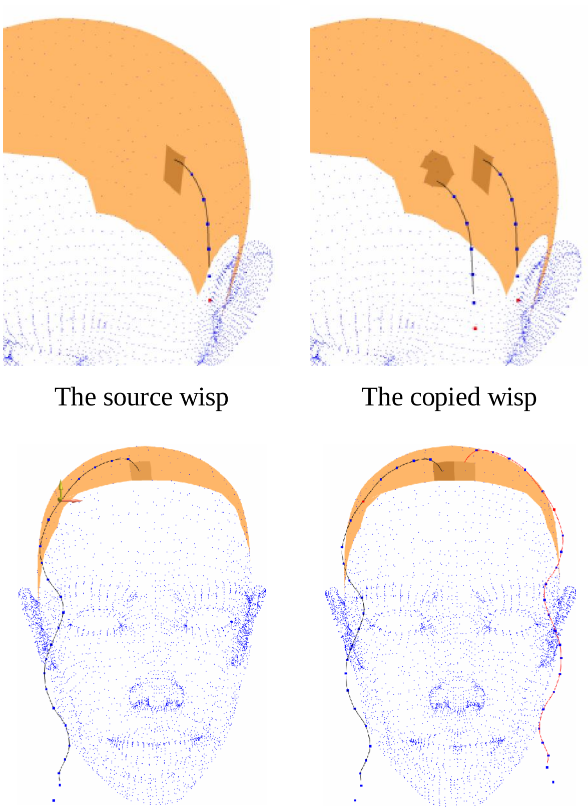


Figure 5: Examples of how the local coordinate system at a control point is changed. The images 1, 2, and 3 show the active arrow being rotated clockwise around the arrow pointing towards the viewer. The images 4, 5, and 6 show the active arrow being rotated clockwise around the bottom yellow arrow.

Since the most suitable styling directions depend on a particular hair style we allow users to adjust the local coordinate system. Users can modify the local coordinate system by rotating the active arrow around one of the non-active arrows as demonstrated in figure 5.

High-level copy/paste and mirror operations between wisps are provided by the key strand copy/paste and mirror component. After selecting the triangles for a wisp, the geometries of the key strand can be copied or mirrored from an existing wisp's key strand by clicking on it as illustrated in figure 6.



The source wisp — The copied wisp

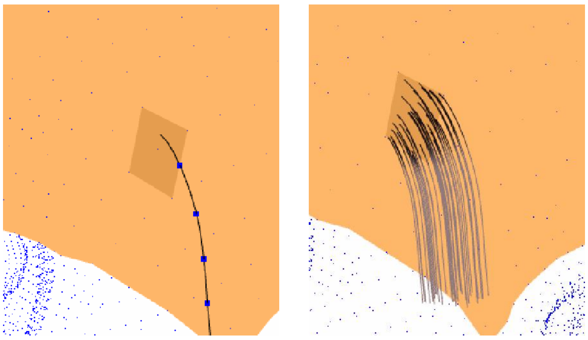
The source wisp — The mirrored wisp

Figure 6: The key strand of the irregularly shaped triangle group in the top right image is an exact copy of the key strand (source wisp) in the top left image. The red key strand in the bottom right image is a mirror version of the key strand in the bottom left image.

The wisp generation component is able to generate all hair strands determined by their key strands and to distribute all hair strands over the scalp uniformly. The geometry of strands is determined using the assumption that the hair strands within one wisp are parallel to each other. The distribution of the hair strands is based on the hair density.

The hair strands within a wisp tend to be similar, although the shapes of the hair strands differ from each other. Choe and Ko observed that the degree of similarity can be controlled by a length distribution, radius distribution and strand variation [11]. The length distribution determines the length variance between the key strand and a member strand within a wisp. The radius distribution controls the distance between the key strand and a member strand within a wisp. Finally the strand distribution gives the shape variation of each strand compared to the key strand. In our implementation we use a length distribution to control the length of each strand and a novel distance distribution, described in section 3, to control the distance between the key strand and a member strand inside of a wisp. We also implemented a strand distribution but found that the hair styles using it were indistinguishable from the ones using just length and distance variations.

Figure 7 demonstrates how a wisp is generated from a key strand.



A key strand — The generated wisp

Figure 7: A key strand (left) and the wisp generated from it (right).

The preview hair component gives users a fast overview of the hair style so that users will be able to adjust the wisps of the hair style quickly.

Using our toolset users can create a wide variety of hair styles within a relatively short period of time. Furthermore details of a particular hair style can be changed easily with the toolset. The toolset is designed such that it can be extended effortlessly in future, e.g. by allowing individually coloured wisps simulating coloured streaks of hair.

3 Implementation

The toolset was implemented in C/C++ using the OpenGL library. The most important part of the hair generation component is the creation of hair strands within a wisp according to the key strand.

The shapes of the member strands within a wisp are determined by their key strand. In our implementation we first use the assumption that the hair strands within one wisp are parallel to each other, and then use a length and distance distribution to make individual member strands different from each other. Both of these distributions are based on the Gaussian distribution.

For the length distribution we define the mean as 95% of the length of the key strand which is computed using a simple first order integration method. The variance of the strands' lengths is user defined depending on the desired hair style. We apply a Gaussian distribution to calculate the length of each member hair strand within a wisp but limit the maximum variation to 5% of the key strand. Hence all hairs within the wisp are within 90-100% of the key strand's length and the distribution of the strands' lengths depends on the desired hairstyle (clean-cut look vs. fringy look).

In order to define the distribution of the distances between a key strand and the strands of the corresponding wisp we first define the strands' root positions using a uniform distribution of points over a triangle such that the density of hair is constant over the scalp. We then define for each control point an offset vector which linearly increases in length for each subsequent control point. The initial offset vector is randomly selected using a uniform distribution over a sphere. In order to maintain the overall shape of the strand the offset vector is defined with respect to a torsion minimising reference frame for the spline curve representing the strand [16].

Note that our implementation offers several advantages over Choe and Ko's one [11], who use random offsets for each control point. This can lead to slightly wavy strands even if the original key strand is uniformly curved. Furthermore by defining the maximum length of the initial offset vector we can produce very smooth hair where the strands are virtually parallel and very fuzzy hair where the distance between hair strands increases at the end of a wisp.

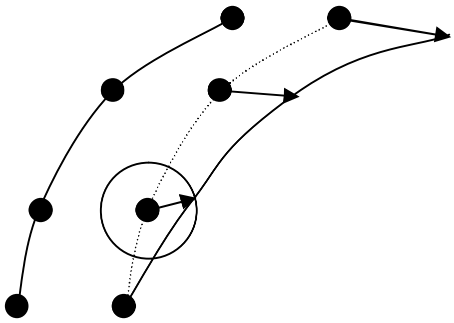


Figure 8: The original key strand (left) and the resulting member strand (right).

Figure 8 illustrates this process. The key strand on the right is reproduced at the new root position. We then define a random offset vector for the first control point subject to a maximum length indicated by the circle in the figure. The offset vector's length linearly increases for subsequent control points. Applying it with respect to the key strand's reference frame generates a new curve of similar appearance.

In order to render hair strands we approximate the Catmull-Rom spline representing them with polylines. This is achieved by computing curve points at equally spaced parameter values using the Catmull-Rom spline equation

$$P(t) = (0.2 * P_1 + (P_2 - P_0) * t + (2 * P_0 - 5 * P_1 + 4 * P_2 - P_3) * t^2 + (-P_0 + 3 * P_1 - 3 * P_2 + P_3) * t^3 / 2$$

where P is a 3D vertex on the spline segment P_1P_2, P_0, P_1, P_2, P_3 are the control points for that segment, and $t \in [0,1]$ is the curve parameter.

4 Results

Our hair styling toolset is capable of creating a variety of moderately complex styles. Depending on the complexity of a new hair style it can take up to several hours for a user without modelling experience to create it. Adjusting the key strands is the most time consuming step when making a specific style. Two examples of completed hair styles created by us are shown in figure 9.

Figure 9: A curly short hair style (left) and a smooth medium length hair style (right) created with our hair styling toolset.

Our hair styling toolset can model real hair styles effectively as demonstrated in figure 10.

Figure 10: Similar hair style generated by the computer (right) and a real human style (left) obtained from [17].

In addition we tested our tool with non-expert users and found that most functions such as wisp copy/paste, mirror, and preview are quite intuitive. However users found that they need to explicitly design the wisp/wisp interactions and it is a little bit difficult to define the directions of key strands. The current version of our toolkit does not perform collision detection between strands/wisps and does not use an explicit physical model and it is therefore difficult to model braided hairstyles.

5 Conclusion

Although the hair styling process can require a couple of hours we found that our toolset enables users to create a variety of hair styles efficiently and effectively. The toolset provides not only high-level functionality such as copy/paste and mirroring of wisps, but also low-level modifications such as changing the number and positions of a key strand's control points in order to modify the shape of a wisp. This was achieved using a novel interaction tool which uses a local-coordinate system for defining "styling directions".

We have introduced a new statistical method to generate strands from a key strand which has the advantage that it maintains consistency of style within a wisp and that it enables users to model smooth, fuzzy and fringy hair. With our density based hair distribution facility the roots of hair strands are distributed evenly over the scalp.

Rendering is performed in real-time using GPU accelerated algorithms and the whole modelling process is interactive.

References

[1] M. Lafrance, "First Impressions and Hair Impressions", Unpublished manuscript, Department of Psychology, Yale University, New Haven, Connecticut. http://www. physique.com/sn/sn_yale-study2.asp, visited on 15th July 2005.

[2] F. I. Parke, "A Parametric Model for Human Faces", PhD thesis, University of Utah, Salt Lake City, UT, UTEC-CSc-75-047, December 1974.

[3] K. Perlin, "Hypertexture", *SIGGRAPH Proceedings*, pp 253-262, 1989.

[4] J.T. Kajiya, T.L. Kay, "Rendering Fur with Three Dimensional Textures", *SIGGRAPH Proceedings*, vol. 23, pp. 271-280, July 1989.

[5] K. Anjyo, Y. Usami, and T. Kurihara, "A simple method for extracting the natural beauty of hair", *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, pp. 111-120, 1992.

[6] R. Rosenblum, W. Carlson, and E. Tripp III, "Simulating the structure and dynamics of human hair: Modelling, rendering and animation", *The Journal of Visualization and Computer Animation*, vol. 2, no. 4, pp. 141-148, October-December 1991.

[7] A. Daldegan, N. M. Thalmann, T. Kurihara, and D. Thalmann, "An integrated system for modelling, animating, and rendering hair", *Eurographics Proceedings*, vol. 12, pp. 211-221, 1993.

[8] X. D. Yang, Z. Xu, J, Yang, and T. Wang, "The Cluster Hair Model", *Graphical Models*, vol. 62, pp. 85-103, 2000.

[9] E. Plante, M. P. Cani, P. Poulin, and K. Perlin, "A layered wisp model for simulating interactions inside long hair", *Proceedings of Eurographics Computer Animation and Simulation 2001*, pp. 139-148, Sep 2001.

[10] T. Kim, U. Neumann, "Interactive multiresolution hair modelling and editing", *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ISBN 1-58113-521-1, ACM Press, New York, NY, USA, pp. 620–629, July 2002.

[11] B. Choe, H. Ko, "A statistical Wisp Model and Pseudophysical Approaches for Interactive Hairstyle Generation", *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 160-170, Mar-Apr 2005.

[12] J. Stam, "Multi-Scale Stochastic Modelling of Complex Natural Phenomena", PhD Thesis, Department of Computer Science, University of Toronto, 1995.

[13] S. Hadap, N. M. Thalmann, "Interactive hairstyler based on fluid flow", *In Proceedings of Computer Animation and Simulation*, pp. 87-100, 2000.

[14] Y. Yu, "Modelling realistic virtual hairstyles", *Pacific graphics*, pp. 295-304, 2001.

[15] E. Catmull, R. Rom, "A Class of Local Interpolating Splines", *Computer Aided Geometric Design*, R. E. Barnhill and R.F. Riesenfeld ed., Academic Press, New York, 1974, pp. 317-326.

[16] J. Bloomenthal, "Calculation of reference frames along a space curve", Graphics Gems Vol. 1, Academic Press, San Diego, CA, USA, 1990, pp. 567-571.

[17] HairBoutique.com – Style #3787, http://gal lery.hairboutique.com/details.asp?ID=3787, visited on 22nd September 2006.