EUROGRAPHICS Workshop on ... (2011), pp. 1–10
M. Chen and O. Deussen (Guest Editors)

Generalized Helicoids for Modeling Hair Geometry

Emmanuel Piuze, Paul G. Kry and Kaleem Siddiqi

School of Computer Science, Centre for Intelligent Machines, McGill University, Canada

Abstract

In computer graphics, modeling the geometry of hair and hair-like patterns such as grass and fur remains a significant challenge. Hair strands can exist in an extensive variety of arrangements and the choice of an appropriate representation for tasks such as hair synthesis, fitting, editing, or reconstruction from samples, is non-trivial. To support such applications we present a novel mathematical representation of hair based on a class of minimal surfaces called generalized helicoids. This representation allows us to characterize the geometry of a single hair strand, as well as of those in its vicinity, by three intuitive curvature parameters and an elevation angle. We introduce algorithms for fitting piecewise generalized helicoids to unparameterized hair strands, and for interpolating hair between these fits. We showcase several applications of this representation including the synthesis of different hair geometries, wisp generation, hair interpolation from samples and hair-style parametrization and reconstruction from real hair data.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

Hair-like patterns, comprised of dense distributions of smooth curves, are ubiquitous in our world. Consider examples arising in nature including fur, grass, feathers or muscle fibers, or those arising in man-made structures, such as optical fibers, cables or thread. Modeling the geometry of such patterns in computer graphics remains a significant challenge due to the extensive variety of possible arrangements of the component strands. Whereas there is an intuitive sense in which hair can be straight, wavy or curly, expressing this mathematically is not trivial. Furthermore, applications constrained by time efficiency or storage requirements demand a suitable level of abstraction. Considering that there are typically more than $100,000$ strands [Rob02] on a human head, it can be impractical to represent each individually.

To simplify the modeling of hair strands, assumptions are often made about their length, their continuity and their local geometry. For example, strands generated through biological processes typically have a coherent orientation and curvature in a local neighborhood. In the computer vision literature, such locally parallel dense curves have been dubbed texture flows [BSZ03]. The coherency is a direct result of the local forces – static, interaction, and external – that apply on a scale larger than each individual strand. This suggests the use of local differential geometric models to express continuity in such patterns.

Motivated by these considerations, we present methods for generating, fitting and interpolating hair patterns using a parametric model based on a class of minimal surfaces called *generalized helicoids*. A generalized helicoid is characterized by three parameters that control curvature in the tangent, normal, and binormal directions to each strand, and a fourth parameter to control elevation angle above a tangent plane [Sav09]. This intuitive representation equips a strand with information not only about its own geometry but also about the “flow” of hair strands in a local volumetric neighborhood. As a result, the model can be used for a variety of applications including hairstyle synthesis and hairstyle reconstruction from sparse samples.

Our major contributions include:

1. The introduction of a generalized helicoid-based framework to model the geometry of a hair strand and its local (volumetric) neighborhood.
2. The synthesis of different hair types by sampling from a generalized helicoid-based representation.

3. The use of the model to “fill-in” a patch of hair between hair strand samples, by interpolation.
4. The use of the model to fit unparametrized hair strand data, which when combined with interpolation allows for efficient hairstyle reconstruction from sparse samples.

2. Related Work

We begin by reviewing related work in the area of modeling the static geometry of hair, or hairstyling. We also refer the reader to review articles on the related topics of hair dynamics and rendering [WBK*07, BHC*08].

2.1. Optimizations

The human scalp is typically populated by more than 100,000 hair strands, that vary in width from 0.05 mm to 0.09 mm [Rob02]. Given this level of resolution, it is necessary to abstract a representation for hair geometry (as well as for rendering and dynamics) particularly in real-time applications. A common strategy is to subsample and select a small number of guide hairs that are representative of the full hairstyle. Additional hair strands can then be interpolated from this subset. These hair strands can be further grouped into larger units to improve the efficiency of subsequent computations, as well as the appearance and texture of the rendered hair. A common approach to grouping involves the formation of wisps, where existing hair strands are locally duplicated, displaced and redrawn, e.g., by using trigonal prisms [WS91]. The wisps are often formed before solving for hair dynamics, as they provide a way to constrain the interactions between larger volumes while requiring fewer pairwise interactions. For simple hairstyles, this optimization is highly convenient at the design, animation, and rendering stages. Consistency is not directly enforced between hair strands belonging to distinct wisps. Cluster-based methods have also been proposed for wisp generation [YXYW00, XY01] and the notion of a skeleton along with a deformable envelope has been used to capture local and global hair motion by Plante et al. [PCP01]. Choe and Ko [CK05] represent wisps as generalized cylinders.

2.2. Explicit versus Implicit Methods

Explicit hair modeling approaches describe hairs as a series of connected geometric primitives. Examples include the use of trigonal prisms [WS91], polylines [KN99, TCC*96] or cylindrical segments [DTKT93] and their cluster-based extensions [CK05, KN02, XY01, YXYW00], the multi-layer wisp model of Plante et al. [PCP01], and super-helices for modeling hair geometry and dynamics [BKMT10, Ber09, BK08, BAC*06]. We note that whereas a super-helix models a single hair strand, the generalized helicoid is a higher dimensional structure that also models the curvature and smoothness of a volumetric neighborhood of streamlines. Pseudo-physical hair systems, where hairs are represented as serial multi-body chains or spring-connected masses rotating about hinges, have also been proposed [Had06], while Kmoch et al. [KBMT09] use elastic rods to describe hair-like objects.

In contrast, implicit approaches describe hair as the output of a function in a continuous medium. A texture-based method was first described by Csuri et al. [CHP*79], while Perlin and Hoffert [PH89] were the first to use volume densities. A textured polygonal mesh is used by Neyret [Ney98], while Yuksel et al. [YSK09] use hair meshes to define the large-scale topology of a hairstyle and generate a refined version by growing hairs along them. In a somewhat different approach, hair geometry can also be modeled by streamlines of a single global fluid flow, which wraps around the scalp and is characterized by sources and vortices [HMT01, Yu01]. This approach is perhaps most similar in spirit to our use of generalized helicoids for incorporating a notion of flow. However, a key difference is that ours is a differential geometric model that is applied locally to capture the geometry of hairs in a neighborhood, and is thus not limited to a single fluid flow for modeling the hairstyle.

2.3. Multiresolution Methods

Multiresolution approaches employ different level of detail computational models depending on the distance from the viewer to the hair, in order to speed up rendering and the simulation of dynamics. For instance, some use a dual representation where the hair is approximated by connected surface patches or strips at large scales, while at smaller scales it is represented by a set of thin cylinders spread over these surfaces [KN00, KH00]. For rendering fur or short hair, Lengyel et al. [LPFH01] use a polyline hair representation for nearby views, while for distant views, concentric texture layers are employed. Others use a hierarchy of generalized cylinders to represent hair [WYZG09, KN02], while Ward et al. [WLJ*03] use a hierarchical representation based on strips, clusters and individual strands.

2.4. Capturing Hair Models from Images

An alternate approach to the modeling of hair geometry involves the capture of real hair data. Perhaps the first such approach is described by Kong et al. [KTN97], where a hair volume is reconstructed from pictures taken from different viewpoints by using a point matching heuristic. Paris et al. [PBS04] and Grabli et al. [GSML02] use a shape from inverse lighting approach to develop a sophisticated method for reconstructing the visible hairs from a small set of images. Similarly, Wei et al. [WOQS05] also use images captured from multiple views to recover the hair geometry. Paris et al. [PCK*08] build on earlier work and simplify the capture of complex features such as concavities and curls. Although they make use of a parameteric model based on structure tensors, it does not immediately translate into a hair modeling framework, where curvature can be controlled in an intuitive manner. Finally, Jakob et al. [JMM09] extract features from hair macrophotographs and uses a sequence of least squares fitting iterations to grow hair strands in a cloud of hair features.

The hair capture methods discussed above show promise in capturing real hair data, but little work has been carried out thus far to parametrize this data so as to facilitate further processing. In fact, the general question of how to obtain a qualitatively faithful reconstruction of a hairstyle from sparse samples remains largely open. Progress on this problem could benefit applications that involve hairstyle editing or transfer. As we shall show, the generalized helicoid-based representation, equipped with algorithms for fitting and interpolation, contributes to this goal.

3. Hair Modeling using Generalized Helicoids

The generalized helicoid is a well-known mathematical object and is the only nonplanar ruled minimal surface. Intuitively, if a curve rotates about a fixed axis and is displaced parallel to this axis such that the displacement rate is always proportional to the angular velocity of rotation, then this generates a generalized helicoid [Che96].

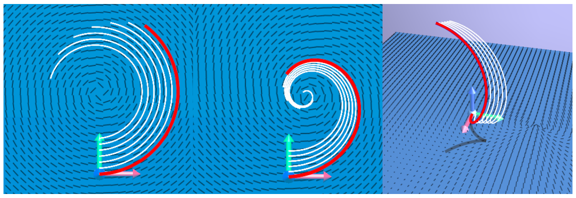
The generalized helicoid model describes the two orientation angles $\theta(x,y,z)$ and $\phi(x,y,z)$ of a 3D flow by a parameter vector $\mathbf{k} = (k_T, k_N, k_B, \alpha)$, in a Cartesian reference frame. In spherical coordinates, these two orientation angles are given by

$$\theta(x,y,z) = \arctan\left(\frac{k_T x + k_N y}{1 + k_N x - k_T y}\right) + k_B z,$$
$$\phi(x,y,z) = \alpha\theta(x,y,z). \qquad (1)$$

Here θ represents the orientation in the xy plane, ϕ represents the elevation angle out of this plane, and α is a constant. As shown by Savadjiev [Sav09], where this abstraction has been used to model white matter fiber tracts of the brain in diffusion MRI, the generalized helicoid has two key mathematical properties. First, the hypersurfaces (x,y,z,θ) and (x,y,z,ϕ) are both minimal surfaces, i.e., surfaces with mean curvature zero. As such, the orientation of the flow is guaranteed to vary smoothly in a local neighborhood. Second, the parameters k_T,k_N, and k_B are each related to a notion of a tangential, normal, and bi-normal curvature, respectively, of the underlying flow field. Continuity for ϕ in Eq. 1 is ensured by counting and adding the number of 2π cycles θ completes.

To gain the intuition behind this model consider a frame field at the origin, with its tangential, normal and binormal components given in spherical coordinates by

$$\mathbf{E}_T = (\cos\phi\cos\theta, \cos\phi\sin\theta, \sin\phi)$$
$$\mathbf{E}_N = (-\sin\theta, \cos\theta, 0) \qquad (2)$$
$$\mathbf{E}_B = (-\sin\phi\cos\theta, -\sin\phi\sin\theta, \cos\phi)\,.$$



(a) $(0.5,0,0,0)$ (b) $(0.5,-0.2,0,0)$ (c) $(0.5,0,0.5,1)$

Figure 1: Varying the parameter vector $\mathbf{k} = (k_T, k_N, k_B, \alpha)$ can produce streamlines that (a) bend and remain in the xy plane, (b) approach a sink, and (c) twist out of plane. The vector field in the xy plane is shown along with a hair strand (red), shadow (grey), and streamlines (white).

Let the tangential component be aligned with the direction of a local flow pattern. The generalized helicoid then describes the fashion in which the frame field must be rotated, in a local neighborhood, to fit the flow pattern at a point (x,y,z). As illustrated in Fig. 1, k_T describes curvature in the tangential direction, k_N provides normal curvature, effectively wrapping the streamlines in a coil-like fashion or fanning them out, and k_B and α allow for local torsion, twisting the streamlines away from or towards the osculating plane. One can observe that flipping the sign of k_T merely results in a reflection of the streamline in the $\mathbf{E}_T$-$\mathbf{E}_B$ plane. This behavior can also be obtained by considering only the magnitude of k_T and instead applying a rotation of π radians about $\mathbf{E}_N$, a strategy we will use in some cases (as described in Section 4.1).

3.1. Tracing Hair Strands

With a generalized helicoid, the geometry of a hair strand is parameterized as $\mathbf{H}(\mathbf{k}, \mathbf{Q}, \mathbf{p}_0, L)$, where $\mathbf{k}$ is the parameter vector, $\mathbf{Q}$ is the coordinate frame at the root of the strand obtained by a rigid body transformation of the frame field at the origin, $\mathbf{p}_0$ is the origin of the root, and L is the length of the hair. An explicit sequence of points on the hair strand is computed as a trace of the frame field described by Eq. 1 (see the red curve in Fig. 1). This representation simultaneously equips the hair strand with implicit information about the curvature of strands in its vicinity (the streamlines in Fig. 1). As such, the generalized helicoid is a hybrid between explicit and implicit representations, with the ability to control hair strand curvature intuitively and efficiently.

To trace a hair strand we use forward Euler integration along the tangential direction $\mathbf{E}_T$ of the frame field, starting from a root location $\mathbf{p}_0$ with the orientations $\theta(\mathbf{p}_0)$ and $\phi(\mathbf{p}_0)$. In all, n steps of length h are taken to produce a hair of length $L = nh$, where points on the hair strand are given by

$$\mathbf{p}_{i+1} = \mathbf{p}_i + h\mathbf{p}_i'$$
$$\mathbf{p}_i' = \mathbf{E}_T\left(\theta(\mathbf{p}_i), \phi(\mathbf{p}_i)\right). \tag{3}$$

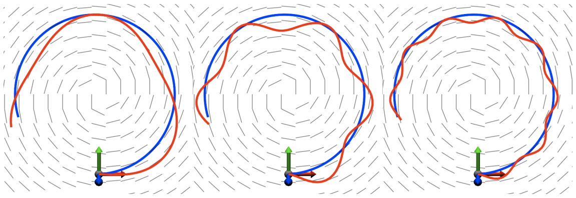
We observe that higher order integration methods are unnecessary due to the fixed hair length and smoothness of the frame field.

3.2. Adding Waviness

With a hair strand parametrized by a generalized helicoid, additional effects can be included for hairstyling. We briefly illustrate a waviness offset effect that is simple to control, efficient to compute (it reuses the precomputed Frenet frame moving along the curve), and couples well with the generalized helicoid tracing algorithm while achieving realism. The basic idea is to preserve the large scale geometry of the hair strand, while adding local helical perturbations. Let the Cartesian coordinates for generating the displacement be replaced by the Frenet coordinates of the generalized helicoid. We parameterize the offset with s and using an amplitude (radius) a and frequency $2\pi b$ map it in the plane spanned by the binormal and normal as follows:

$$\tilde{p}(s) = p(s) + a\cos(bs)\mathbf{E}_B(s) + a\sin(bs)\mathbf{E}_N(s). \tag{4}$$

Fig. 2 shows a hair strand (blue) along with its waviness offset (red) for amplitudes and frequencies of different magnitudes. In practice we fix the amplitude of the helical offset while allowing the frequency to vary. Additionally, stochastic perturbations can be trivially added to the generalized helicoid tracing algorithm to produce realistic variations.



(a) $a = 0.1, b = 0.15$ (b) $a = 0.1, b = 0.3$ (c) $a = 0.05, b = 0.5$

Figure 2: Original hair (blue) and waviness offset (red) with helical radii a and frequencies b.

3.3. Modeling Wisps

The implicit nature of the generalized helicoid model can be leveraged to model wisps very efficiently. The basic idea is to offset the origin $\mathbf{p}_0$ in Eq. 3, to obtain a dense distribution of locally parallel hair strands. Examining Fig. 1, this amounts to tracing additional streamlines that surround a master hair strand. In a sense this can be viewed as a type of "extrapolation", but with essentially no additional computation required. This feature is of particular interest for proponents of the hair wisp model.

Random orientation perturbations can be added to further emphasize the fanning of the wisp. To illustrate, Fig. 3 shows a wisp generated by extrapolating 256 strands around a guide hair by sampling and tracing streamlines from a neighborhood of its root location.

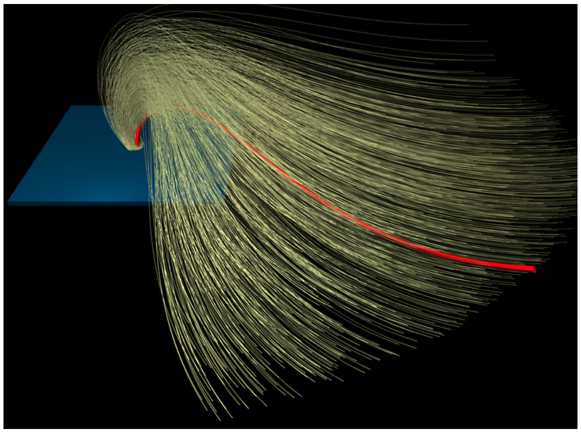
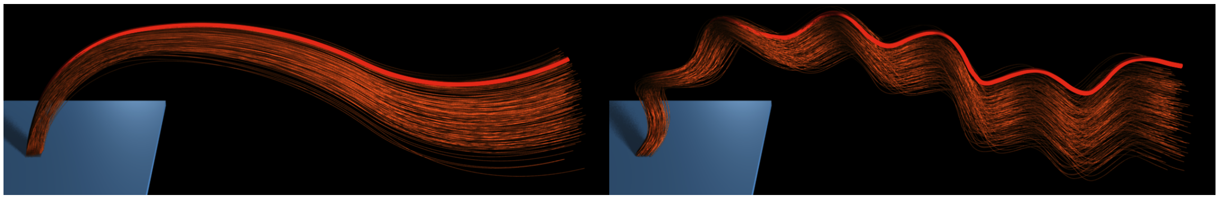


Figure 3: A wisp generated by offsetting the origin $\mathbf{p}_0$ and sampling additional streamlines from the generalized helicoid. The guide strand of the helicoid is shown in red, with the extrapolated hairs comprising the wisp shown in gold.

3.4. Hairs as Piecewise Helicoids

A more powerful and flexible way of representing hairs with complex geometry is to make use of a composite model, which we refer to as a piecewise-helicoid or a *p-helicoid*. A p-helicoid is obtained by concatenating multiple generalized helicoids, aligning them using the Frenet frame constructed by Eq. 2. This characterization of hair strands provides C^∞ continuity along a piece and enforces continuity of the Frenet frame as it moves from one piece to the next, a condition stronger than C^1 continuity.

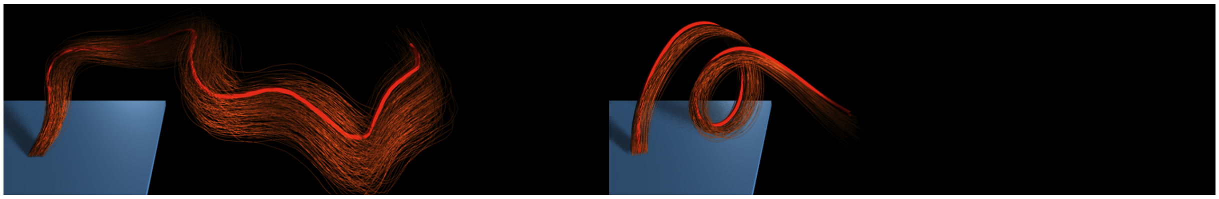
Piecewise helicoids can be used to generate the geometry of more complex hairstyles, and as described above, waviness offsets can be added for additional realism. Fig. 4 and 5 show several examples of p-helicoid based hair strands with waviness effects, along with their extrapolated wisps. Fig. 4 shows a long hair composed of 2 helicoid pieces with varying curvature parameters. Fig. 5 shows a "messy" hair obtained using 3 helicoidal pieces with different curvature parameters and a curly hair composed of 4 helicoidal pieces having similar curvature. In all cases the wisps are comprised of 256 hairs.



(a) A long hair.

(b) A long hair with a waviness offset.

Figure 4: A long hair in red (left) combined with a waviness offset (right). Two helicoid pieces are used. The associated extrapolated wisps are shown in brown.



(a) A “messy” hair.

(b) A curly hair.

Figure 5: A “messy” hair (in red, on the left) using three helicoid pieces, and a curly hair (in red, on the right) using four helicoid pieces. The extrapolated wisps are shown in brown.

4. Hair Interpolation

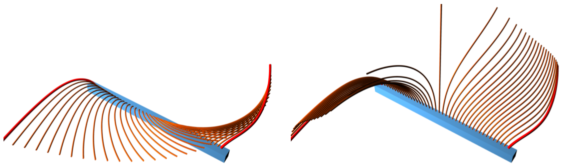
The property that any orientation vector field characterized by Eq. 1 is a generalized helicoid suggests a very natural approach to filling in sparse regions between hair strands or hairstyle completion. The essential idea is to interpolate between the parameter vectors $\mathbf{k}$ associated with each strand by an inverse distance weighting (IDW) scheme to generate new hair strands. This enforces a smooth variation in curvature. We first consider interpolation of single generalized helicoids and p-helicoids on a surface that is described explicitly. We then develop the more general case of interpolating on an arbitrary surface, where new root node locations must first be determined.

4.1. Interpolation Between Single Helicoids

Consider the simple case of interpolating single-piece generalized helicoids on an explicit surface. Let the set of guide hairs, $\mathbf{H}_i$ for $i = 1 \ldots N$, have roots at $\mathbf{x}_i$ and parameters $\mathbf{k}_i$, $\mathbf{Q}_i$, $\mathbf{p}_i$, and L_i. For a new root location sampled at $\tilde{\mathbf{x}}$, a new hair $\tilde{\mathbf{H}}$ is obtained by interpolating the parameters. We use a simple Shepard inverse distance scheme for the interpolation weights, $w_i = \phi_i / \sum_{i=1}^{N} \phi_i$, where $\phi_i = ||\tilde{\mathbf{x}} - \mathbf{x}_i||_{\alpha}^{-1}$. Varying α provides a control of the interpolated hairstyle, either smoothing it out or generating clumps of hair. We use $\alpha = 3$ in all our examples.

Note that if a guide hair has a negative k_T we can rotate its orientation frame $\mathbf{Q}$ by π radians about $\mathbf{E}_N$, and negate k_T. This does not change the hair geometry but improves the interpolation by preventing two geometrically identical guide strands deferring only in the sign of k_T from producing a “straight” helicoid between them.

Fig. 6 (left) illustrates the process of interpolation between two guide hairs (shown in red) along a line drawn between them to generate new hair strands (shown in brown). In this example the two guide hairs are π-rotated copies of one another. Fig. 6 (right) illustrates an alternative approach of interpolating using only the points sampled along the guide hairs. Observe that the generalized helicoid-based interpolation leads to a smoother flow from one guide hair to the other, i.e., one with less curvature variation. In addition, it is many orders of magnitude faster since only the (small number of) helicoid parameters need to be interpolated.



(a) Helicoid-based

(b) Sample Point-based

Figure 6: A comparison of interpolating guide hairs using generalized helicoids (left) and using only points sampled along the guide strands (right). The two guide hairs (red) are π-rotated copies of each other, with $\mathbf{k} = (1, -0.1, 0.3, 1)$.

4.2. Interpolation Between p-Helicoids

The extension of the method to handle guide strands described by p-helicoids on an explicit surface uses the same interpolation scheme and weighting function. However, the new interpolants must now populate not only root locations on the growth surface but also locations above it. This is accomplished with a piece by piece strategy. For the first piece of each new hair strand we use exactly the same strategy as with single helicoids, interpolating $\mathbf{k},\mathbf{Q},\mathbf{p}_0$ and L. Additional pieces of equal length are then computed, in a sequential fashion, such that their Frenet frames are aligned at touching endpoints and the sum of their length is L. Thus, for the additional pieces beyond the first we only interpolate $\mathbf{k}$. The number of pieces in interpolated p-helicoids is determined using the weighting method described in 4.1. Fig. 7 and 8 illustrate that the interpolation between p-helicoid guide hairs on a line, and on a plane, respectively, achieves a smooth variation in curvature between the guide hairs.

We observe that interpolation of p-helicoids is fast, even with our unoptimized implementation. For instance, using a cluster of 9 p-helicoid guide hairs with an average of 4 pieces each, 1000 p-helicoids can be interpolated in 0.67 s using a single 2.4GHz processor. This translates to an average of 0.7 ms per hair, or approximately one minute to interpolate a dense head of hair (90k hair strands). These interpolations can be sped up considerably by computing them in parallel on multiple processors.

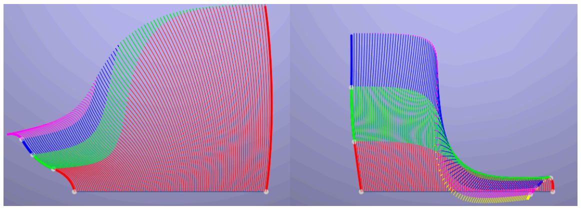

(a) Example 1 (b) Example 2

Figure 7: Two examples of p-helicoid interpolation on a line. Distinct pieces of the p-helicoid interpolants are shown in different colors. Two guide hairs are used, one at each end point (in bold).

4.3. Interpolation on an Arbitrary Surface

To interpolate across guide hairs on an arbitrary surface, root locations for new hairs (follicles) must be determined. The method we use is based on a particle sampling approach. First, an implicit surface is fit to vertices on the initial object model using the thin-plate energy minimization technique described by Turk and O'brien [TO02]. An adapted implementation of Witkin and Heckbert's [WH94] is then used to sample a particle system, driven by repulsion forces and implicit constraints. This method distributes particles across the implicit surface in a near-regular manner, by allowing particles to fission (divide in two) and die. Fixed particles constrain the sampling within a region on the implicit surface. This process is illustrated in Fig. 9.

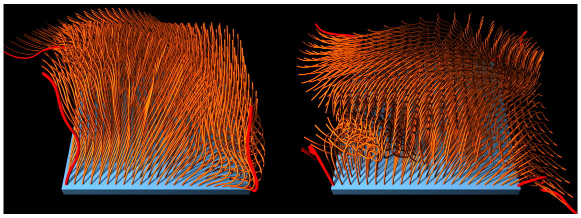

(a) Example 1 (b) Example 2

Figure 8: Two examples of interpolation between p-helicoids on a plane. The four guide hairs are shown in red.

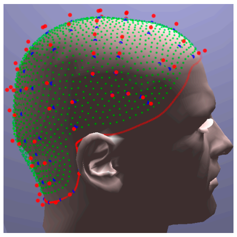


Figure 9: Follicle locations (green) sampled on an arbitrary surface. The scalp contour (red) is defined by fixed particles. The red and blue points represent constraints for the implicit surface.

The normal to the implicit surface serves to orient a coordinate frame **W** at each root node location. The local frame **Q** of an interpolated p-helicoid is obtained by combining this coordinate frame **W** with an interpolation of the coordinate frames of the neighboring guide hairs using the scheme described in Section 4.1. Fig. 10 illustrates the interpolation process on a patch of scalp, using a single helicoid for each strand for simplicity. The method creates a tuft of hair (shown in gold) with smoothly varying curvature between the guide strands (shown in red). One can also apply the interpolation technique to obtain a smooth variation between guide strands of varying geometry, as shown in Fig. 11. In this latter example, long and curly hairs (on the side) transition to a straight hair parting on the forehead. Fig. 12 illustrates the process of creating a “fauxhawk” hairstyle using the methods discussed in this section. The hairstyle was constructed from 20 p-helicoid guide hairs of varying p-helicoid parameters, length, and orientation frames.

5. Hair Fitting

In the examples presented so far we have synthesized hair patterns from scratch using generalized helicoids. There are

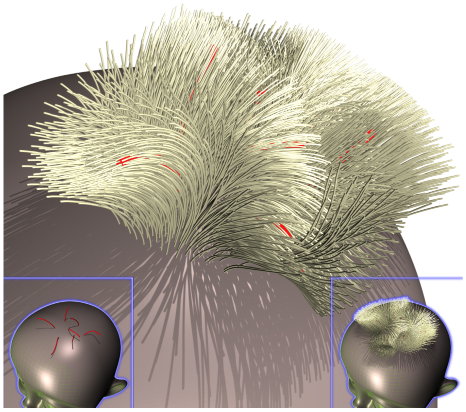
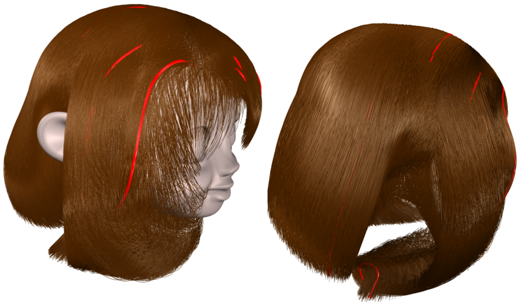


Figure 10: Interpolation of single helicoids on an implicit surface to create a tuft of hair. In this example the guide hairs, shown in red, are generated with random parameter vectors **k** and coordinate frames **Q**. The interpolated strands are shown in gold.



(a) Front view

(b) Back view

Figure 11: Seven guide hairs (red) are interpolated (brown).

other situations, however, where real hair data may be available, described for example by a set of polylines. In such cases it might be desirable to parametrize this data so as to be able to efficiently edit and manipulate it. We address this goal by devising an algorithm for fitting a p-helicoid to a polyline representation of a hair strand. When combined with the methods for interpolation of the previous section it becomes possible to reconstruct a hairstyle, qualitatively, from a sparse set of samples. We provide some striking examples of this using hairstyles from the hair photobooth database [PCK*08].

5.1. The Fréchet Similarity Measure

A key component in the process of fitting a trace of points sampled from a generalized helicoid to a polyline representation of a hair strand is the measure by which the quality of fit is judged. Alt and Godau [AG95] provide an intuitive description of the Fréchet distance, using the analogy of a dog and its handler walking on their respective paths. Both the man and the dog can control their speed independently but can never backtrack. The Fréchet distance between these two curves is the minimal length that a leash must have, over all possible manners that the dog and the handler can move from the starting points of the two curves to their respective endpoints. In the fitting of generalized helicoids to hair data we use a discrete variant of this distance, which is referred to as the Fréchet coupling [EM94]. Returning to the previous analogy, this measure represents the minimum total length of leashes needed for men walking their dogs – one on each point of each curve – such that leashes do not cross over. For two polylines of p and q points, the simplest algorithm for computing the Fréchet coupling has a computational complexity of $O(pq)$. The more familiar discrete Hausdorff distance has the same computational complexity but the Fréchet coupling has the added benefit of being symmetric and considers point ordering [EM94].

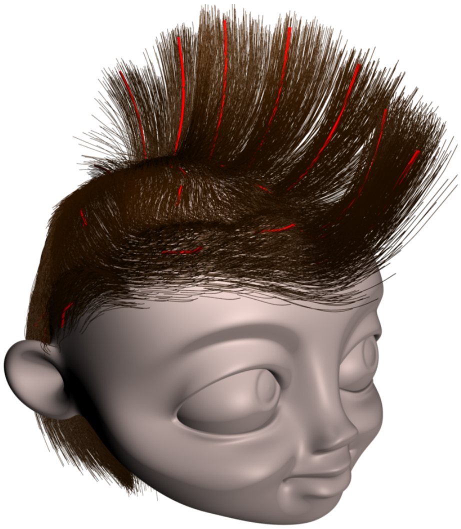


Figure 12: Creation of a hairstyle (brown) from a set of 20 guide hairs (red).

5.2. Fitting Single Generalized Helicoids

In the simplest case we fit a single generalized helicoid to a target hair strand. A parameter vector $\mathbf{k} = (k_T, k_N, k_B, \alpha)$ that generates the trace of a hair is optimized iteratively. The objective function is the minimization of the Fréchet coupling – represented here by the function $\mathcal{C}$ – between a target hair $\mathbf{h}$ with estimated Frenet frame $\mathbf{Q}$ [Ros03] and the trace generated by the helicoid $\mathbf{H}(\mathbf{k}, \mathbf{Q}, \mathbf{p}_0, L)$:

$$\mathbf{k}_0 = \underset{\mathbf{k}}{\text{argmin}} \;\; \mathcal{C}\left(\mathbf{h}, \mathbf{H}(\mathbf{k}, \mathbf{Q}, \mathbf{p}_0, L)\right) \; + \; \lambda ||\mathbf{k}|| \qquad (5)$$

where λ is a constant to regularize the fit by preventing excessively large curvature parameter vectors. L is set as the length of **h**. Since the objective function is not analytical, we use the Nelder-Mead (NM) algorithm [NM65] to minimize numerical errors. This is a gradient-free method that optimizes a function in a multidimensional space and performs well with arbitrary seed points.

5.3. Piecewise Helicoid Fitting

In practice, a single generalized helicoid is usually adequate to fit a hair that contains up to one cusp or inflection point. Clearly it is desirable to extend the fitting process so as to be able to cope with longer hairs with more complex geometry. To this end, we devise a process to fit p-helicoids. Each individual helicoid piece is fit to different pieces of the target strand, following the methods described in Section 5.2. The locations where the target hair is to be fragmented need to be determined and for this purpose, we use the following recursive procedure.

The fitting process is initially carried out on the complete target strand. If the Fréchet coupling measure from the fit to the target strand is small enough, it is accepted. Otherwise, the target strand is split into two pieces. The efficiency of the piecewise fitting depends largely on this distribution of target fragments. The fragmentation locations are determined by a combination of Fréchet and differential heuristics. First, a thresholding is used on the Fréchet coupling between the target and candidate fragments, i.e., a threshold of 100% fits the entire fragment at once. It was determined that 60% provides a suitable trade off between overfitting and avoiding large biases. The objective of the Fréchet thresholding is to provide an upper limit on the fragmentation location. In some cases, a premature acceptance yields inconsistencies in the orientation of a fragment with respect to its fit. In order to ensure that fragments share similar differential characteristics with their fit, a curvature profile is computed on the fragment and the fit, starting from the upper limit from the Fréchet threshold and backtracking to the origin of the fragment. A new fragmentation location is determined such that it minimizes a difference in curvature between the fragment and the fit. In order to prevent excessive backtracking, a threshold is also used on the minimal length that a fragment can have.

Fragment fits are automatically aligned with the Frenet frame of their predecessor, computed using equation 2, and thus preserve C^1 continuity between them. Piecewise helicoid fits for different hairs extracted randomly from unparameterized data in [PCK*08] are shown in Figure 13. Whereas at a fine scale each fit is not exact, they nicely capture the qualitative geometry of the hairstyle, and their representation requires far less information than storing all the points on each target hair strand. This property allows the qualitative reconstruction of a hairstyle from sparse samples of real hair data, as described next.

Because fitting hairs to p-helicoids involves optimization, it is much slower than interpolation. However, for modeling and reconstruction purposes, we generally start with few hair samples and it is ultimately the interpolation of large numbers of hairs which is more costly. With our unoptimized implementation we can fit the data provided by Paris et al. [PCK*08] at approximately 5 hairs per second using a single 2.4GHz processor. The hairs in this dataset have an average of 30 vertices each. The fitting process can be sped up considerably by computing the fits in parallel using multiple processors.

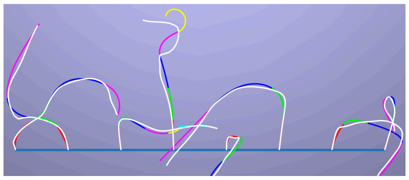


Figure 13: P-helicoid fitting (in color) of unparameterized data (in white) selected randomly from Paris et al. [PCK*08].

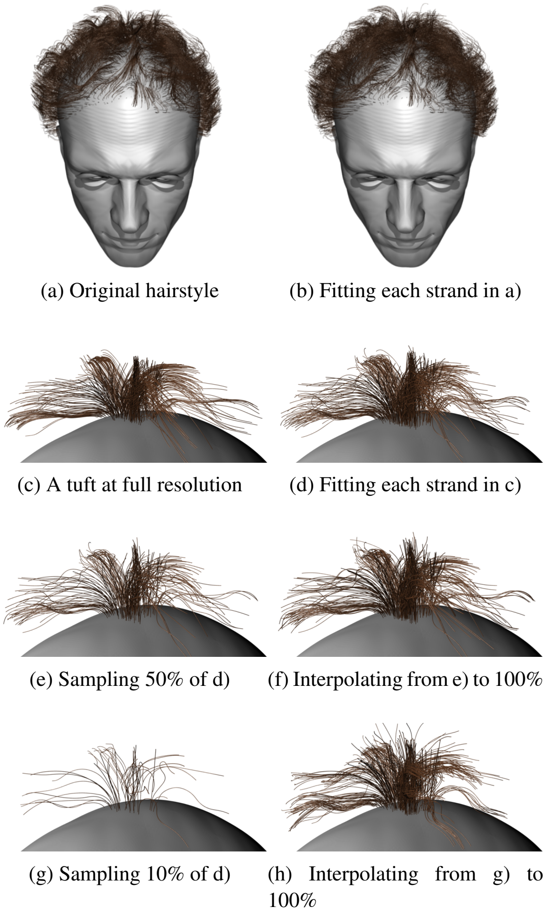
5.4. Application to the Reconstruction of Real Hair

Figure 14 illustrates the use of the methods in this paper to fit and interpolate real hair data from a selected example in the hair photobooth database [PCK*08]. Fig. 14 (top left) shows a sparse sampling of the "straight" hairstyle, along with the individual p-helicoid fits to each hair strand sample (top right). Observe that the fit closely resembles the sampled hair strands. The remaining rows focus on the process of reconstructing a selected tuft at the resolution of the original data (Fig. 14 (c)) by means of subsampling and interpolation. Observe that salient geometrical features of the original hairstyle can be recovered from as little as 10% of the fit to the guide hairs.

6. Conclusion

The generalized helicoid and its piecewise instantiation, the p-helicoid, complement models presently in use for hair geometry by equipping a hair strand with a differential geometric characterization of the curvature of strands in its vicinity. We have described methods by which p-helicoids can synthesize hair patterns with rich and diverse geometry, including wisps with waviness offsets. We have also developed methods to efficiently interpolate between guide hair strands, each parametrized by a p-helicoid, while ensuring a smooth variation in curvature between them. Complete hairstyles can be modeled and existing ones can be reused by applying parameter variations.

Contrasting with other streamline methods, where the hair volume consists of a global fluid flow [HMT01, Yu01], we construct a local fluid flow representation. In a purely static modeling environment this property allows the modeling of messy hairstyles more precisely than with global flow techniques.



(a) Original hairstyle

(b) Fitting each strand in a)

(c) A tuft at full resolution

(d) Fitting each strand in c)

(e) Sampling 50% of d)

(f) Interpolating from e) to 100%

(g) Sampling 10% of d)

(h) Interpolating from g) to 100%

Figure 14: The reconstruction of the "straight" hairstyle from [PCK*08] by fitting and interpolation of p-helicoids.

Although hair interpolated using generalized helicoids is very coherent locally, there is nothing that prevents clumps generated by distinct generalized helicoids from getting entangled. This can make the modeling more realistic for wavy or fuzzy hairstyles. In applications where hair intersection is undesirable additional constraints would have to be incorporated.

Motivated by progress in reconstruction methods for acquiring real hair data [PBS04, PCK*08, JMM09], where individual hairs are described by the vertices of a polyline, we have developed a procedure for fitting a p-helicoid to a hair strand using a measure of similarity based on Fréchet distance. The fitting process can be carried out for a set of sampled hair strands, following which their p-helicoid parametrization can be interpolated densely to fill in the hairstyle. Our results indicate that both the fits and the interpolated strands preserve the qualitative geometry of the hairstyle. In fact, we are able to reconstruct a complex tuft of hair from as little as 10% of the original unparametrized hair data. Thus, the p-helicoid representation may be used to parametrize and then compress a hairstyle, an application that will likely grow in interest as databases containing detailed hair capture data become available.

Whereas the examples in this article focused on human hair, the methods developed are readily applicable to the modeling of other hair-like patterns including, for example, fur, grass and feathers. Individual strands in fur typically have a simpler geometry than human hair and a higher degree of geometric coherency. Similarly, the geometry of the barbs of a feather emerging from its main shaft closely resembles that of a generalized helicoid.

Given the success of the piecewise helical super-helix representation for modeling hair dynamics [BAC*06, Ber09], a second area of research that could prove fruitful is the coupling of these ideas with the p-helicoid representation. Since the p-helicoid captures the geometry of the flow of hair strands in a volumetric neighborhood of a guide hair, it allows for sparsity and hence efficiency. It would be interesting to determine whether the physics of hair, such as the modeling of external forces including wind and contact, or internal properties such as torsion, bending and stretching, can be embedded directly into the p-helicoid parameterization.

A final area of application could be the use of generalized helicoids for hair detail transfer, as developed in [WYZG09], or hairstyle editing. P-helicoids can serve as the skeleton for a variety of hairstyles, where the details are obtained by adding waviness, noise, and possibly other kinds of effects, such as those obtained by varying the curvature parameters to straighten or curl hair, for example.

References

[AG95] ALT H., GODAU M.: Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications 5*, 1 (1995), 75–91. 7

[BAC*06] BERTAILS F., AUDOLY B., CANI M.-P., QUERLEUX B., LEROY F., LÉVÊQUE J.-L.: Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics 25*, 3 (2006), 1180–1187. 2, 9

[Ber09] BERTAILS F.: Linear time super-helices. *Computer Graphics Forum 28*, 2 (2009), 417–426. 2, 9

[BHC*08] BERTAILS F., HADAP S., CANI M., LIN M., KIM T., MARSCHNER S., WARD K., KAČIĆ-ALESIĆ Z.: Realistic hair simulation: animation and rendering. In *ACM SIGGRAPH 2008 courses* (2008), pp. 89:1–89:154. 2

[BK08] BONANNI U., KMOCH P.: Virtual hair handle: A model for haptic hairstyling. In *Proc. Eurographics short papers* (2008), pp. 135–138. 2

[BKMT10] BONANNI U., KMOCH P., MAGNENAT-THALMANN N.: Interaction metaphors for modeling hair using haptic interfaces. *International Journal of CAD/CAM 9*, 1 (2010), 93–102. 2

[BSZ03] BEN-SHAHAR O., ZUCKER S.: The perceptual organization of texture flow: A contextual inference approach. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence 25, 4 (2003), 401–417. 1

[Che96] CHEN B.: A report on submanifolds of finite type. *Soochow Journal of Mathematics 22*, 2 (1996), 117–337. 3

[CHP*79] CSURI C., HACKATHORN R., PARENT R., CARLSON W., HOWARD M.: Towards an interactive high visual complexity animation system. In *Computer Graphics (Proceedings of SIGGRAPH)* (1979), pp. 289–299. 2

[CK05] CHOE B., KO H.: A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics 11*, 2 (2005), 160–170. 2

[DTKT93] DALDEGAN A., THALMANN N., KURIHARA T., THALMANN D.: An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum 12*, 3 (1993), 211–221. 2

[EM94] EITER T., MANNILA H.: *Computing discrete Fréchet distance*. Tech. rep., Technische Universitat Wien, 1994. 7

[GSML02] GRABLI S., SILLION F., MARSCHNER S., LENGYEL J.: Image-based hair capture by inverse lighting. In *Proc. Graphics Interface* (2002), pp. 51–58. 2

[Had06] HADAP S.: Oriented strands: dynamics of stiff multibody system. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 91–100. 2

[HMT01] HADAP S., MAGNENAT-THALMANN N.: Modeling dynamic hair as a continuum. *Computer Graphics Forum 20*, 3 (2001), 329–338. 2, 8

[JMM09] JAKOB W., MOON J., MARSCHNER S.: Capturing hair assemblies fiber by fiber. *ACM Transactions on Graphics 28*, 5 (2009), 164:1–164:9. 3, 9

[KBMT09] KMOCH P., BONANNI U., MAGNENAT-THALMANN N.: Hair simulation model for real-time environments. In *Computer Graphics International* (2009), pp. 5–12. 2

[KH00] KOH C., HUANG Z.: Real-time animation of human hair modeled in strips. In *Proc. Eurographics Workshop on Computer animation and simulation* (2000), pp. 101–109. 2

[KN99] KONG W., NAKAJIMA M.: Visible volume buffer for efficient hair expression and shadow generation. In *Proc. Computer Animation* (1999), pp. 58–65. 2

[KN00] KIM T., NEUMANN U.: A thin shell volume for modeling human hair. In *Proc. Computer Animation* (2000), pp. 104–111. 2

[KN02] KIM T.-Y., NEUMANN U.: Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics 21*, 3 (2002), 620–629. 2

[KTN97] KONG W., TAKAHASHI H., NAKAJIMA M.: Generation of 3D hair model from multiple pictures. In *Proc. Multimedia Modeling* (1997), pp. 183–196. 2

[LPFH01] LENGYEL J., PRAUN E., FINKELSTEIN A., HOPPE H.: Real-time fur over arbitrary surfaces. In *Proc. symposium on Interactive 3D graphics* (2001), pp. 227–232. 2

[Ney98] NEYRET F.: Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics 4*, 1 (1998), 55–70. 2

[NM65] NELDER J., MEAD R.: A simplex method for function minimization. *The Computer Journal 7*, 4 (1965), 308–313. 8

[PBS04] PARIS S., BRICEÑO H., SILLION F.: Capture of hair geometry from multiple images. *ACM Transactions on Graphics 23*, 3 (2004), 712–719. 2, 9

[PCK*08] PARIS S., CHANG W., KOZHUSHNYAN O., JAROSZ W., MATUSIK W., ZWICKER M., DURAND F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Transactions on Graphics 27*, 3 (2008), 30:1–30:9. 2, 7, 8, 9

[PCP01] PLANTE E., CANI M., POULIN P.: A layered wisp model for simulating interactions inside long hair. In *Proc. Eurographics Workshop on Computer animation and simulation* (2001), pp. 139–148. 2

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. In *Computer Graphics (Proceedings of SIGGRAPH)* (1989), pp. 253–262. 2

[Rob02] ROBBINS C.: *Chemical and physical behavior of human hair*. Springer Verlag, 2002. 1, 2

[Ros03] ROSEN K.: *Discrete mathematics and its applications*. McGraw-Hill New York, 2003. 7

[Sav09] SAVADJIEV P.: *Perceptual Organization in Diffusion MRI: Curves and Streamline Flows*. PhD thesis, McGill University, 2009. 1, 3

[TCC*96] THALMANN N. M., CARION S., COURCHESNE M., VOLINO P., WU Y.: Virtual clothes, hair and skin for beautiful top models. In *Proc. Computer Graphics International* (1996), pp. 132–141. 2

[TO02] TURK G., O'BRIEN J.: Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics 21*, 4 (2002), 855–873. 6

[WBK*07] WARD K., BERTAILS F., KIM T., MARSCHNER S., CANI M., LIN M.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 213–234. 2

[WH94] WITKIN A. P., HECKBERT P. S.: Using particles to sample and control implicit surfaces. In *Computer Graphics (Proceedings of ISGGRAPH)* (1994), pp. 269–277. 6

[WLJ*03] WARD K., LIN M., JOOHI L., FISHER S., MACRI D.: Modeling hair using level-of-detail representations. In *Proc. Computer Animation and Social Agents* (2003), pp. 41–47. 2

[WOQS05] WEI Y., OFEK E., QUAN L., SHUM H. Y.: Modeling hair from multiple views. *ACM Transactions on Graphics 24*, 3 (2005), 816–820. 2

[WS91] WATANABE Y., SUENAGA Y.: Drawing human hair using the wisp model. *The Visual Computer 7*, 2 (1991), 97–103. 2

[WYZG09] WANG L., YU Y., ZHOU K., GUO B.: Example-based hair geometry synthesis. *ACM Transactions on Graphics 38*, 3 (2009), 56:1–56:9. 2, 9

[XY01] XU Z., YANG X. D.: V-hairstudio: an interactive tool for hair design. *IEEE Computer Graphics and Applications 21*, 3 (2001), 36–43. 2

[YSK09] YUKSEL C., SCHAEFER S., KEYSER J.: Hair meshes. *ACM Transactions on Graphics 28*, 5 (2009), 166:1–166:7. 2

[Yu01] YU Y.: Modeling realistic virtual hairstyles. In *Proc. Pacific Graphics* (2001), pp. 295–304. 2, 8

[YXYW00] YANG X., XU Z., YANG J., WANG T.: The cluster hair model. *Graphical Models 62*, 2 (2000), 85–103. 2