Modeling Realistic Virtual Hairstyles

Yizhou Yu
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 West Springfield Avenue, IL 61801, USA
www-faculty.cs.uiuc.edu/∼yyz
yizhouy@acm.org

ABSTRACT

In this paper we present an effective method for modeling realistic curly hairstyles, taking into account both artificial hairstyling processes and natural curliness. The result is a detailed geometric model of hairs that can be rendered and animated via existing methods. Our technique exploits the analogy between hairs and a vector field; interactively and efficiently models global and local hair flows by superimposing procedurally defined vector field primitives that have local influence. Usually only a very small number of vector field primitives are needed to model a complicated hairstyle. An initial model of hair strands is extracted from the superimposed vector fields by tracing their field lines. Random natural or artificial curliness can be added to the initial model through a parametric hair offset function with a randomized distribution of parameters over the scalp. Techniques for shearing and clustering are also designed to improve the overall appearance of the hair model. Our technique has been successfully applied to generate a variety of realistic hairstyles with different curliness and length distributions.

Keywords: Hair Modeling, Deformations, Vector Fields, Offset Functions

1 Introduction

Modeling human characters poses some of today's most challenging problems in computer graphics. Modeling and designing natural hairstyles is one of them since a good hairstyle is crucial to a person's appearance. Realistic representation of human hair presents problems in all aspects of computer graphics technologies, i.e. rendering, shape modeling and animation [10, 6, 11, 12, 20, 14, 13, 1, 4, 3]. We classify hairstyling as a shape modeling problem. Most previous work on hairs only considers creatures and furry objects [10, 6, 16, 13] with short hair. There are basically two different techniques for modeling long hairs.

The Wisp model has been used with some success in some of the papers [26, 4, 3]. It typically models the hair by assigning a set of example hairs to the patches or vertices on the scalp, and then procedurally fill in the hairs in-between using interpolation and a certain level of randomness. However, specifying an example hair for each surface patch on the scalp and modeling new example hairs not in the original set need significant amount of user interaction, and can be laborious and tedious. The other approach models hairs by tracing particle trajectories through motion vector fields [25, 7]. [25] considers the motion field as the inducing force field and emphasizes the dynamic appearance of hair instead of complex static hairstyles. [7] needs to simulate the dynamics of a flow vector field to obtain particle trajectories. Controlling the dynamics to produce desirable hairstyles remains difficult. A few hair modeling plug-ins [23, 24] have been commercially available for 3D Studio MAX and the Alias/Wavefront Maya modeling system. They adopt either of the two approaches. However, a survey of the features available in these commercial packages is beyond the scope of this paper. In a word, efficiently modeling and designing complex wavy hairstyles from long curly hair remains a tough problem.

The most important thing in a hairstyle lies in the way hair strands curve and deform. The causes for hair deformation can be summarized as natural curliness, artificial hairstyling processes and deformation under external forces such as gravity, collision and static charges. According to [9], the degree of natural curliness is indicated by the curvature of a hair strand when it is not under any external forces. In an actual hairstyling process, many artificial procedures are provided by a hairdresser, such as perming, combing, shearing and the application of cosmetics. The effects of perming and cosmetics are most influential to cause long-term artificial hair deformation. They overcome natural curliness, and make hairs hard to model. Lastly, hairs bend under gravity and collision. The degree hair strands bend shows the synergy between hair pliability and these external forces. This effect is easily reversible because a hair can resume its original shape as these forces disappear.

Because of artificial hairstyling and external forces, hair strands do not grow and curve completely randomly in all directions, but follow certain global as well as local flow patterns. For example, long hair is usually draped down, and nearby strands usually form a cluster and deform in the same way. However, there is still small amount of randomness that distinguishes strands from each other in the way they curve. Other essential factors to hair appearance lie in the natural and physical properties of hair, including hair color, width, pliability, and volumetric appearance.

A hairstyle modeling method for our purpose should be computationally tractable, while considering all the factors for hair deformation as well as other properties. A novel technique dealing with the intrinsic factors as well as the artificialities is desired. It should also have a friendly interface to both rendering and animation[12, 20, 1, 4] since we would lose much of the natural beauty of hair if we excluded the potential to apply any of these technologies.

In this paper, we introduce a series of novel techniques for modeling and designing hairstyles. These techniques include

- an approach to simulate both global and local hair flows using vector field primitives,
- a generic offset function to simulate natural curliness and random small-scale deformations,
- methods for shearing and clustering.

The techniques we introduce here have a number of advantages: *i) the approach is extremely expressive, and can efficiently produce a variety of realistic hairstyles using a small number of parameters; ii) The user has full control of the design process, therefore, is able to produce novel hairstyles from imagination; iii) They have a good interface to rendering and animation; iv) The algorithms are efficient.*

1.1 Overview

Previous approaches either model individual hairs explicitly [1, 26, 20, 4, 3] or model all hairs collectively as a volume with a density at each point [10, 18]. We take the explicit approach here, considering the enormous variations of hairstyles. The input to our method is a polygonal model of a head including the scalp and face. Our overall virtual hairstyling process has the following multiple steps: *1) select a region on the scalp for hair growth; 2) specify a length distribution for hairs growing from the region; 3) select a set of vector field primitives interactively and specify their corresponding rigid transformations to produce a global vector field; 4) extract an initial model of the hairs from the vector field; 5) edit each strand using an offset function to produce random curliness; 6) pull together nearby hairs to form clusters; 7) render the generated hairs.*

The rest of the paper is organized as follows. Shearing techniques are presented in Section 2. Hair modeling by vector field primitives is introduced in Section 3. Adding additional random curliness is discussed in Section 4. We briefly talk about rendering in Section 5, and present results in Section 6.

2 Shearing

We first need to define the region on the scalp where hair should be grown (Fig. 4(a)). The contour of this region is generated by linearly interpolating the 2D polar coordinates of a few points, which have been interactively selected in a flattened map of the scalp. The center of this map aligns with the top center of the scalp. Once specified, the same region can be repeatedly used for multiple hairstyles.

Shearing is the starting point of hairdressing. Every hairstyle needs a corresponding hair length distribution. The length of a particular strand is determined by the location of its root on the scalp. We use the vector between the root of the strand and the centroid of the scalp to define this location in a 3D polar coordinate system. Hairdressers sometimes cut hair with respect to some reference planes[28, 22]. For example, they would cut lower part of the hair to a horizontal plane at the bottom of the neck, and pull upper part of the hair straight up and cut it to a horizontal plane on top. At other times, the hair would be cut to have a smoothly varying length[28, 22].

According to these two shearing styles. We designed two representations for hair length distributions. The first one is based on a BSP tree. The plane defined for an intermediate node of the BSP divides a region on the scalp into two smaller ones. The plane at a leaf node of the BSP actually defines a reference plane for hair-cutting. All hairs rooted in the region corresponding to that leaf node should be cut to the reference plane given at the same leaf. This is done by setting the length of a strand to be the length of the shortest path outside the scalp between its root and that reference plane. So part of the shortest path may be curved and lying on the scalp. We actually use a sphere to approximate the shape of the scalp to accelerate this calculation. A hair length distribution based on this BSP representation is shown in Fig. 4(b)-(c).

The second representation is a linear interpolation model based on the length specified at six pre-defined locations on the head. They are the intersections between the head and the three axes of a canonical coordinate system centered at the centroid of the head. There is not really hair growing on the front and bottom sides of the head. But the specified length, which can be negative, at the two intersections on these two sides will be needed for linear interpolation.

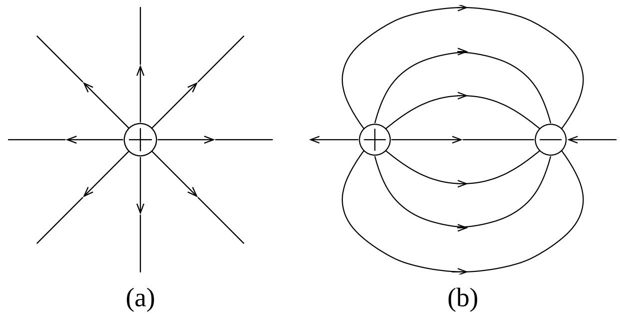


Figure 1. (a) An electric field induced by a positive point charge; (b) an electric field induced by two opposite point charges.

The six points can be connected to construct eight triangles with an overall diamond shape. Every point on the scalp can be projected onto the surface of this diamond shape using the centroid of the scalp as the center of projection. For a particular strand, we can locate which of the triangles the projected point of its root belongs and determine its length by linearly interpolating the length at the vertices of the triangle. This model corresponds to the second shearing style.

3 Modeling Hair Flow with Vector Field Primitives

We decide to use vector fields to simulate both global and local hair flows caused by perming, combing and cosmetics because both have clear orientation at each point and both belong to volumetric data. Vector fields are particularly useful in modeling the volumetric appearance of hair. In computer graphics, there has been considerable amount of previous work on visualizing vector fields with volume rendering techniques [2], and applying the concept of vector fields to nonphotorealistic rendering[8]. Motion vector fields have also been applied to hair modeling and animation in [25] and [7]. However, [25] has emphasis on hair motion by repeatedly tracing particles with a fixed life-time through a motion field which changes over time; and [7] only adopts a 2D flow vector field without the capability of composition for complex styles. These methods require either synthesizing or simulating the dynamics of a motion field. This section independently develops an intuitive and flexible technique to model complex 3D curly hairstyles using static 3D vector fields. In a word, we need to work out how to interactively design a vector field that looks very much like natural hair when visualized. The use of static vector fields for hair modeling removes the complexity of simulating the dynamics of a motion field, and provides easier and more direct control of the modeling process.

A 3D *vector field* can be considered as a mapping from a domain $\Omega \subseteq R^3$ to the 3D vector space. The returned 3D vector from this mapping specifies a unique orientation and strength of the field for every nonsingular point in the domain Ω. There may be singular points, such as sources and sinks, in the vector field with infinite magnitude and not well-defined orientation. If we put two vector fields together, the vector corresponding to a point in the superimposed field should be the summation of the two vectors in the original fields. A *field line* in a vector field is a 3D curve such that the tangential direction at a certain point on the curve is collinear with the field orientation at that point. Fig. 1(a) and (b) show the field lines of electric fields induced by a positive point charge, and by two opposite point charges, respectively.

3.1 Vector Field Primitives with Local Influence

It is difficult to generate a complete vector field for a complicated hairstyle at once. To have easy control and editing of hair flows globally as well as in some local regions, we need to decompose the global vector field into multiple primitives with each one responsible for one feature in the hair flows. For example, we can use a field primitive to bend hairs towards the back of the head. We also need an intermediate level of local control. For example, hairs on the forehead may curve quite differently from those on the back of the head. But we definitely do not want to edit the shape of each individual strand. *Vector field primitives* can satisfy these requirements. They are procedurally defined simple vector fields with each one having a rigid transformation between the world coordinate system and its local coordinate system. The rigid transformation can be interactively edited to change the position and orientation of a vector field primitive. We can easily edit the global vector field, therefore global hair deformation, by inserting new vector field primitives, deleting existing primitives, or modifying the rigid transformations of existing primitives.

The orientation and strength of a vector field primitive at a certain point is provided by two separate (procedurally defined) functions. Usually we require that they are continuous functions. To achieve local influence, the strength of a field primitive is actually a product of the inherent field strength and a spatial term that diminishes with increasing distance from a reference point in the field. In practice, we have used three common functions for the spatial term, namely, uniform, inverse power and a smoothly decreasing function connecting two constants. The inverse power function is defined as

$$IP(r) = \frac{s_0}{(r - r_0)^e} \qquad (1)$$

where r is the distance from the considered point to the reference point, s_0, r_0 and e are constant parameters. This function creates a singular point at $r = r_0$ with infinite magnitude. Hairs close to this singular point are deformed dominantly by the field primitive creating this singular point. If a singular point is not desirable, the inverse power function can be replaced with an exponential function which decays when r increases.

The smoothly decreasing function is defined as

$$SD(r) = \begin{cases} s_0, & r \leq r_0; \\ 0.5(s_0 + s_1) + \\ \quad 0.5(s_0 - s_1)\cos(\frac{r - r_0}{r_1 - r_0}\pi), & r_0 < r < r_1; \\ s_1, & r \geq r_1 \end{cases} \tag{2}$$

where a cosine function connects two constant parameters s_0 and s_1. Either s_0 or s_1 can be set to zero. Local influence can be achieved by setting s_1 to zero. The product of Eq. (1) and (2) provides another possible spatial term which has infinite magnitude at $r = r_0$, and finite (possibly zero) magnitude at $r = r_1$.

The locality of a vector field primitive can also be defined on the scalp. We can associate a list of local regions on the scalp with a field primitive so that only hair strands growing from these regions can be affected by this field primitive. The union of these local regions is called the *domain* of the field primitive.

3.2 Vector-Tracing for Hair Extraction

A number of vector field primitives superimposed together define a global field which forms the base for all hairs. The superimposed field obviously follows the definition of a vector field. We still need to extract hair strands from this vector field by tracing field lines. The starting point of a hair strand is always at its root on the scalp. The roots of all hairs are randomly distributed in the region of the scalp where hair should be grown. At every step during tracing, we need to evaluate each field primitive whose domain covers the root of the considered strand, and sum up all the vectors from these evaluations. Then the considered hair strand extends by a certain step size along the direction of the accumulated vector. This is repeated until the hair has reached its predefined length. We can either use a small constant step size or an adaptive step size that dynamically changes according to the local curvature of the hair strand being traced. So we can see that hair strands are actually represented as 3D piecewise linear curves.

Two properties of the extracted hair strands can be summarized as follows when an infinitesimal step size is used.

1. If at a certain point P, each of the vector field primitives is continuous with respect to both magnitude and orientation, and the magnitude of the superimposed vector field is nonzero, then the hair curve passing through P has C^1 continuity at P.
2. If the vector at a certain point P in a vector field is uniquely defined, there are no hair curves intersecting each other at P.

The first property indicates that most of the hair curves traced out have C^1 continuity which guarantees smooth appearance. However, there exist singular points in a vector field where the magnitude of the field becomes zero and the orientation is not well-defined. Fortunately except for field sources and sinks, it is unlikely that a hair curve passes such a singular point.

The second property indicates that hair curves usually do not cross each other and they form nonintersecting layers, which is essentially a volumetric representation and improves realism by letting us sense the thickness of the hair.

We need to further consider hair-scalp collision during vector-tracing. To detect whether a strand has gone underneath the scalp, we generate a spherical depth map centered at the centroid of the scalp from the set of polygons representing the scalp and face. At each step of vector-tracing, the depth value of the currently considered hair vertex is compared with the corresponding one in the spherical depth map. In case it is smaller, we increase the depth of the vertex until it moves outside the scalp.

An example of a vector field and the extracted hair model from the vector field are shown in Fig. 4(d)-(f).

3.3 Examples of Vector Field Primitives

We introduce a few important vector field primitives in this section. Some of them were actually inspired by electromagnetic fields[27]. In the following, we adopt some names from electromagnetic fields. But the definition of those fields have been slightly changed. In most situations, the strength of the field can be redefined to incorporate either Eq.(1) or (2). Although each of these field primitives is quite simple itself, multiple superimposed field primitives can lead to a very complicated vector field whose field lines have the shape of desired natural hairs. Declarations of new field primitives can be integrated into our framework quite conveniently.

- **Electric Field of a Point Charge** The orientation of any point in this field is either pointing to or pointing away from the point charge which is also the reference point of this field. This field can be used to attract hair to or repel hair from a certain point. Similarly, we can define an electric field of a line segment.

- **Electric Field of an Ellipsoidal Charge** We can define this field by introducing a family of ellipsoids in a canonical coordinate system as follows.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = r^2 \tag{3}$$

where r is a variable parameter and a, b and c are constants. Any fixed value of r defines an ellipsoid from this family. The orientation of a point in this field is collinear with the normal direction of the ellipsoid from the above family and passing through this point. The particular value of r for this ellipsoid is used to define the spatial term in field strength at that point. A very thin ellipsoidal field primitive can function as a hair divider (Fig. 3(a)).

- **Magnetic Field Induced by a Linear or Arc Current** The field lines of a linear current form concentric circles around and perpendicular to the line segment inducing the field. The orientation of the field at a point is tangential to the circle passing that point. This field can wrap hair around it and therefore simulate the perming effects of a hair roller (Fig. 3(b)). If we tune the orientation at each point so that it is a little bit away from the tangential direction of a circle passing that point, spiral hair strands can be obtained. The distance r in the spatial term is determined by the shortest distance from the considered point to the line segment. Similarly, we can define a magnetic field induced by an arc current.

- **Field Simulating a Comb** If we induce a field around a short line segment by defining the orientation at any point in the field to be parallel to the line segment, we can simulate the effects of a comb.

- **Field Primitives for Volumetric Waves** To design regular volumetric wavy structures in a local region, we introduce a field primitive defined in a pyramid with infinite height and with its apex at the centroid of the scalp. Any hair intersecting this pyramid will be affected and become wavy. The orientation of a point in this field is collinear with the vector connecting the point and the centroid of the scalp. The strength of the field is an angular sinusoidal function which means the period of the wave becomes smaller when it gets closer to the centroid of the scalp. By applying this field, the phases of the hair waves within the pyramid are forced to be synchronized no matter where the roots of the strands are (Fig. 3(c)). Section 4 will introduce techniques to obtain random asynchronous hair waves.

3.4 Modeling Pliability

Every hair strand has a certain amount of strength to resist external attempts and maintain the path of its original sweep. Therefore, the actual path of a hair should deviate from the field lines of the vector field. Suppose we know the original sweep of a hair. By default, the sweep is a straight line. Without any external influences, the hair should extend along the tangential direction of its sweep at each point. To model hair pliability, we need to revise the above vector-tracing. At every step, in addition to accumulate the vectors from all field primitives, we also consider the tangential direction of the original hair sweep. Therefore, the final direction for hair growth at each point is a weighted average of the tangential direction and the accumulated vector.

In general, the original sweep of a hair is a smooth curve passing through a sequence of control points. For the reason mentioned above, we need to use tangential directions at points on the sweep during vector superposition instead of 3D positions of the points themselves. So we designed a technique to directly estimate the tangential direction of a point on the sweep without explicitly solving the curve representing the sweep. We assume the actual arc length of the part of the sweep between two consecutive control points is proportional to the distance between the two points. The tangential direction of any point on this part of the sweep is obtained by linearly interpolating the tangential directions at these two control points. The tangential direction at a control point can be estimated in advance by numerical differentiation from itself and the preceding and following control points in the sequence. Special cases for the first and last control points need to be considered separately. Thus, the estimated tangential directions of the original sweep of a strand can be used during hair tracing to model pliability. As a result, we obtain an updated sweep for each strand because of the influence of the vector field. As a by-product, we can also estimate the original sweep itself by tracing the tangential directions. The accuracy becomes better with more control points. This is analogous to solving ordinary differential equations using Euler's method[19].

Externally defined curves can be imported into our framework by specifying the imported curves as the original sweeps of hair strands. These curves further deform under the influence of the vector field. As mentioned, we have to use the tangential directions of the curves during vector-tracing. An example of an imported curved hair sweep is illustrated in Fig. 3(d). We can view this part as the interface between the Wisp model and our framework.

4 Generating Random Curliness with an Offset Function

Given the hair flow pattern from the previous section, we still need to add natural curliness of hair strands and even some local randomness in the sweep of each strand to create an overall natural appearance. To achieve this, we first define a specific offset function for hair strands in a canonical coordinate system, then reconstruct curly hairs by modulating the hair sweeps from vector fields with this function.

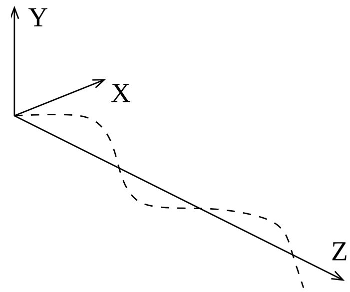


Figure 2. The canonical coordinate system for the hair offset function.

4.1 Offset Function

We assume that the underlying hair strand is a straight half line coincident with the positive z-axis of the canonical coordinate system for the offset function, and the origin is at the root of this strand. The offset function is a two-component function that returns the offsets along both $x-$ and $y-$ axes given a z value(Fig. 2).

According to [15], there are three primary types of natural hair waves, namely, a uniplanar wave, a dished wave and a helix. A uniplanar wave looks very much like a planar sinusoidal wave, while a dished wave looks like a sinusoidal wave which has been mapped onto half of a cylindrical surface with the axis of the wave lengthwise along the cylinder. [15] further presented the author's research results on the main reason for the formation of these waves, which is the sequence of periodical contraction and relaxation of small-scale muscles that are in control of the various follicle configurations. Please note that an artificial helix can also be shaped with a hair roller during perming[28] by first wrapping hairs around the roller and then removing the roller along its symmetric axis.

We can see that all three types of hair waves can be easily represented with our offset function although we do not have to be restricted to these types when synthesizing wavy hairs. The uniplanar wave can be represented with a sinusoidal offset for the x-axis and a zero offset for the y-axis. A helix can be represented with a sinusoidal offset for the x-axis and a cosinoidal offset for the y-axis. The dished wave can be represented with a sinusoidal offset for the x-axis, but the wave for the y-axis is a more complicated function which always returns a positive offset value and whose period is at most half of the period of the wave for the x-axis.

Although a variety of parametric or procedurally defined functions can be used, we designed a specific class of offset functions with a fixed number of parameters as follows.

$$\text{Wave}(t) = \text{Mag}(t)\sin(2\pi(Rt + P_0)t + \phi_0) + Bias \quad (4)$$

where t is the function variable; R, P_0, ϕ_0, and $Bias$ are constant parameters; and

$$\text{Mag}(t) = A + Bt\exp(-\alpha t) + C(1 - \exp(-\beta t) + D\exp(\gamma(t - t_0)) \quad (5)$$

where A, B, C, D, α, β, γ, and t_0 are all constant parameters. We can see this is basically a sinusoidal wave with variable period whose initial value is P_0, and with variable magnitude whose initial value is A. If $R > 0$, the period of the wave becomes smaller as we move closer to the tip of the hair, which is a common phenomenon exhibited by real wavy hairs. The last three terms in Eq.(5) allow us to vary offset magnitude at different parts of a hair strand. The second term reaches a maximum at $t = 1/\alpha$; the third term becomes close to C with sufficiently large t; and the last term increases exponentially especially when $t > t_0$ which can be used to model curly features at the end of a strand. With two sets of different parameters, this function can model offsets along both x- and y-axes. All the examples shown in this paper use this parameterization of the offset function.

4.2 Editing Hair Sweeps with the Offset Function

To modulate a hair sweep from vector fields with the above offset function, we need to consider the hair sweep as the parametric axis for the variable t in the offset function. For a certain point p on the sweep, its corresponding t value is the accumulated arc length between p and the root of the sweep. Thus, we can obtain a pair of offset values for each point on the sweep.

We still need to define a local coordinate system at each point on the sweep to impose the offsets. The z-axis of the local system at a point p is always the tangential direction of the sweep at p. We use the vector between the point p and the centroid of the scalp as an UP vector for the local system. Then both x- and y- axes can be derived from the UP vector and the z-axis. So p is originally at the origin of this local system. Its new location is always on the local xy-plane and is determined by the pair of offsets returned from the offset function. [21] presents a similar idea to edit details of a 2D curve.

4.3 Random Curliness

To achieve natural appearance, obviously we should not edit all hair sweeps with the same set of offset parameters. On the other hand, random hair curliness is far from white noise. The author of [15] observed smoothly spatially varying initial phase, ϕ_0 in Eq.(4), in hair waves. During perming, usually a local cluster of hair is wrapped around a roller, which suggests that hair waves from that cluster share similar shapes. In practice, we set up a multi-scale grid over the scalp in a polar coordinate system, and generate smoothly varying random parameters for the offset function using Perlin's noise[17] given the mean and standard deviation of each parameter for each scale.

4.4 Hair Clustering

Because of cosmetics and static charges, nearby hairs tend to form clusters. This effect is partially dealt with in Section 4.3 when multiscale noise is synthesized. Hairs growing from the same cell in the highest resolution grid have the same parameters for the offset function, therefore tend to be close to each other. Here we introduce an additional offset to further reinforce this effect when the previous treatment is not enough. One hair is chosen to be the representative of the hairs growing from the same cell in the grid. Each vertex on other hairs from the same cell is forced to move by a certain distance towards its corresponding vertex on the representative hair. Hair clusters become obvious after this step.

Fig. 5(a)-(c) show the difference the offset function can make on hair appearance. Fig. 5(d) demonstrates the effectiveness of the hair clustering step.

5 Rendering

Hair rendering is not the focus of this paper. However, we need rendering to produce visually appealing results. Most of the time, the diameter of a hair strand is smaller than the size of a pixel. Antialiasing can be considered mandatory to generate reasonable results. Usually supersampling is used. Some of the rendering algorithms [12, 20] model the segments of hair strands as extremely thin cylinders. This is actually not necessary. We can directly render hair strands as a set of line segments with one-pixel width in a super-resolution image which will be filtered down to an appropriate resolution regarding the diameter of the strands[1]. We adopt the latter scheme. The length of each line segment is on the order of 1mm to make curved strands look smooth. Each individual hair strand is shaded using the model in [10]. Self-shadows among the strands is generated from the modified shadow buffer algorithm presented in [12] which already produces good results. The algorithm in [14] can also be applied for better shadowing effects.

6 Results

We have applied our techniques to modeling a variety of hairstyles with different degrees of curliness, and obtained satisfactory results. Some of the hairstyles are shown in Fig. 6. They look natural and realistic. All the vector field primitives are defined as functions in the C++ programming language. It is rather easy to insert new field primitives by writing short C++ functions. All the hairstyles shown in this paper were produced from eight vector field primitives. We need to use multiple instances of the same primitive. Each instancing needs a new rigid transformation, including a translation and a rotation, between the world coordinate system and the primitive's local system, which can either be interactively adjusted by the user or be randomly generated. The user also needs to specify the parameters in Eq. (1) or (2) to define the new primitive's influence region.

The number of instances of field primitives used by each hairstyle in Fig. 6 is summarized as follows: 13 for Fig. 6(a), 12 for Fig. 6(b), 12 for Fig. 6(c), 273 for Fig. 6(d), 2 for Fig. 6(e), and 11 for Fig. 6(f). The rigid transformations of the 273 vector field instances for Fig. 6(d) were actually randomly generated and distributed over the scalp. They are instances of the field primitive induced by a linear current. For the other five hairstyles, the time for user interaction is typically between one and two hours. Obviously, it only took a few minutes to instantiate the two primitives for Fig. 6(e).

For each local cluster of hairs, the parameters in Eq. (4) are randomly generated. The mean and standard deviation of each parameter at each scale are fixed for all hairs, and are specified by the user. For the results shown here, we use two levels of scales to generate the random parameters. The final hair waves in Fig. 6(a) and in Fig. 6(e) actually have the same average magnitude, 5mm. However, hair waves in Fig. 6(e) look more random because they have a much shorter period and a much larger standard deviation for the initial phases of the waves. The hairs in Fig. 6(c) have an average magnitude of 1mm to give a smoother look.

Each hairstyle in Fig. 6 has about 50,000 hairs. It took between 30 seconds and 4 minutes on a Pentium III 733MHz processor for our program to go through the multiple modeling stages and output the final hair deformation. The actual amount of time depends on the density and length of the hairs. The worst-case time complexity of our program is $O(NM)$ where N is the number of vertices in the hair model and M is the number of vector field primitives because the step on deforming hairs with offset functions only has linear complexity, $O(N)$. Since each hairstyle typically has a small number of field primitives, this complexity does not impose any serious problems. Since each primitive may even have its own local influence region, this complexity can be easily reduced with a space partitioning scheme. The 273 primitives for Fig. 6(d) are actually mutually exclusive with each hair influenced by only one primitive, which drops the time complexity for this particular example to $O(N)$.

7 Conclusions and Discussions

In summary, we have presented and demonstrated a new method for modeling realistic curly hairstyles, taking into account both artificial hairstyling processes and natural curliness. It exploits the analogy between hairs and a vector field; interactively and efficiently models global and local hair flows by superimposing procedurally defined vector field primitives that have local influence. An initial model of hairs is extracted from the superimposed vector fields by tracing their field lines. Random natural or artificial curliness can be added to the initial model through a parametric hair offset function. With our approach, a user can design realistic hairstyles fairly easily either according to his imaginations or according to some real examples.

Our approach is targeted at designing virtual hairstyles. One possible future research direction is combining image-based modeling techniques [5] with our approach to digitize real hairstyles. But we are concerned about two issues with such a hybrid approach. First, images only capture the appearance of the visible parts of the hairs. It is hard to figure out the location on the scalp each hair is growing from. There is also not enough information available to model volumetric hair appearance which is particularly useful for hair animation. Second, the width of a hair is most likely smaller than that of a pixel. It is difficult to distinguish the data for each individual hair strand from images.

Another interesting problem is modeling braided hair. Braided hair has more complicated deterministic structures, but less random structures. However, it is still possible to define braiding patterns procedurally.

Acknowledgments

This research was supported by the Faculty Startup Funds from University of Illinois at Urbana-Champaign. The author wishes to thank the reviewers for their valuable comments.

References

[1] K. Anjyo, Y. Usami, and T. Kurihara. A simple method for extracting the natural beauty of hair. In *Proc. of SIGGRAPH'92*, pages 111–120, 1992.

[2] B. Cabral and L.C. Leedom. Imaging vector fields using line integral convolution. In *Proc. of SIGGRAPH'93*, pages 263–272, 1993.

[3] L.-H. Chen, S. Saeyor, H. Dohi, and M. Ishizuka. A system of 3d hair sytle synthesis based on the wisp model. *The Visual Computer*, 15(4):159–170, 1999.

[4] A. Daldegan, N.M. Thalmann, T. Kurihara, and D. Thalmann. An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum(Eurographics'93)*, 12(3):211–221, 1993.

[5] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1993.

[6] D. Goldman. Fake fur rendering. In *Proc. of SIGGRAPH'97*, pages 127–134, 1997.

[7] S. Hadap and N. Magnenat-Thalmann. Interactive hair styler based on fluid flow. In *Computer Animation and Simulation 2000. Proceedings of the Eleventh Eurographics Workshop*, 2000.

[8] P. Haeberli. Paint by numbers: Abstract image representations. In *Proc. of SIGGRAPH'90*, pages 207–214, 1990.

[9] R.R. Ogle Jr. and M. J. Fox. *Atlas of Human Hair(Microscopic Characteristics)*. CRC Press, 1999.

[10] J. Kajiya and T. Kay. Rendering fur with three dimensional textures. In *Proc. of SIGGRAPH'89*, pages 271–280, 1989.

[11] W.M. Kong and M. Nakajima. Visible volume buffer for efficient hair expression and shadow generation. In *Proc. of IEEE Computer Animation*, pages 58–65, 1998.

[12] A.M. LeBlanc, R. Turner, and D. Thalmann. Rendering hair using pixel blending and shadow buffers. *Journal of Visualization and Computer Animation*, pages 92–97, 1991.

[13] J. Lengyel. Real-time hair. In *Proc. of Eurographics Workshop on Rendering*, pages 243–256, 2000.

[14] T. Lokovic and E. Veach. Deep shadow maps. In *Proc. of SIGGRAPH'00*, pages 385–392, 2000.

[15] A.G. Lyne and B.F. Short. Biology of the skin and hair growth. In *Proc. of a Symposium held at Canberra, Australia*, 1964.

[16] G. Miller. From wire-frame to furry animals. In *Proc. of Graphics Interface'88*, pages 138–146, 1988.

[17] K. Perlin. An image synthesizer. In *Proc. of SIGGRAPH'85*, pages 287–296, 1985.

[18] K. Perlin and E.M. Hoffert. Hypertexture. In *Proc. of SIGGRAPH'89*, pages 253–262, 1989.

[19] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge Univ. Press, New York, 1988.

[20] R.E. Rosenblum, W.E. Carlson, and E. Tripp. Simulating the structure and dynamics of human hair: Modeling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2:141–148, 1991.

[21] M.P. Salisbury, S.E. Anderson, R. Barzel, and D.H. Salesin. Interactive pen and ink illustration. In *Proc. of SIGGRAPH'94*, pages 101–108, 1994.

[22] M.T. Scali-Sheahan. *18 Men's Styles*. Milady Publishing Company, Albany, NY, 1994.

[23] Shag (plugin for 3d studio max). home.abac.com/ddag/hair.html.

[24] Shave (plugin for lightwave). www.joealter.com/software.html.

[25] J. Stam. *Multi-Scale Stochastic Modeling of Complex Natural Phenomena*. PhD thesis, University of Toronto, 1995.

[26] Y. Watanabe and Y. Suenaga. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics and Applications*, 12(1):47–53, 1992.

[27] E. Weber. *Electromagnetic Fields (Theory and Applications)*. John Wiley & Sons, Inc., New York, 1950.

[28] K. Young. *28 Styles for Student Practice with Basic Cutting and Styling Guides*. Milady Publishing Company, Albany, NY, 1992.

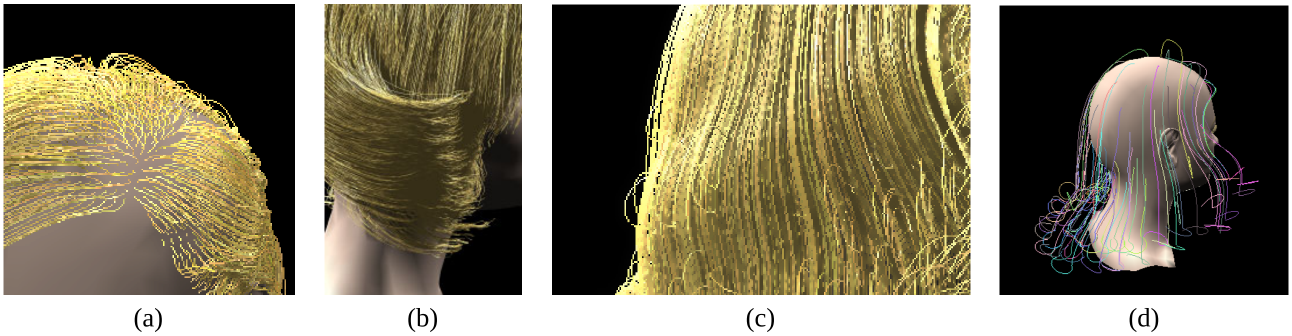


Figure 3. (a) As shown here, a thin ellipsoidal vector field primitive can be used for dividing hairs; (b) the magnetic field induced by a linear or arc current can be used as a perming roller to wrap hair around; (c) synchronous volumetric wave produced from the last primitive introduced in Section 3.3; (d) an imported curved hair sweep is integrated into a vector field to create a curly tip for the hairs.

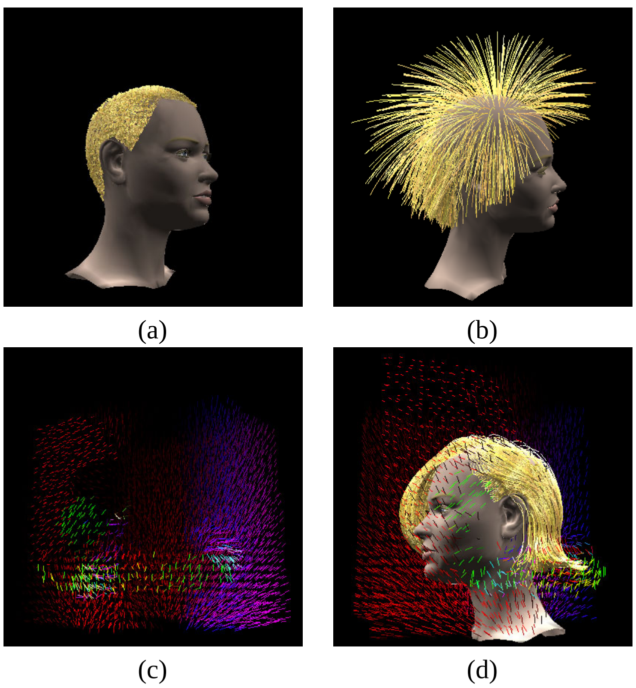


Figure 4. (a) A polygonal head model and the selected region on scalp for hair growth; (b) a length distribution represented with a BSP tree; (c) a visualization of a vector field generated from twelve field primitives. The orientation at each point is pseudo-colored with brightness indicating field strength; (d) the initial hair model extracted from the vector field in (c).

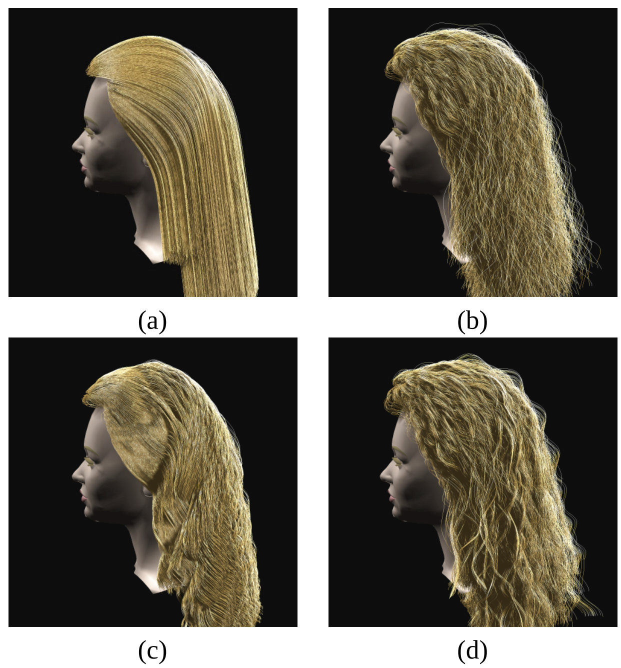
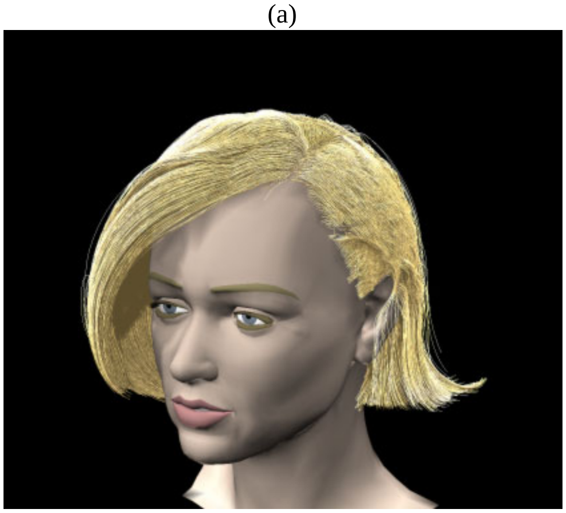


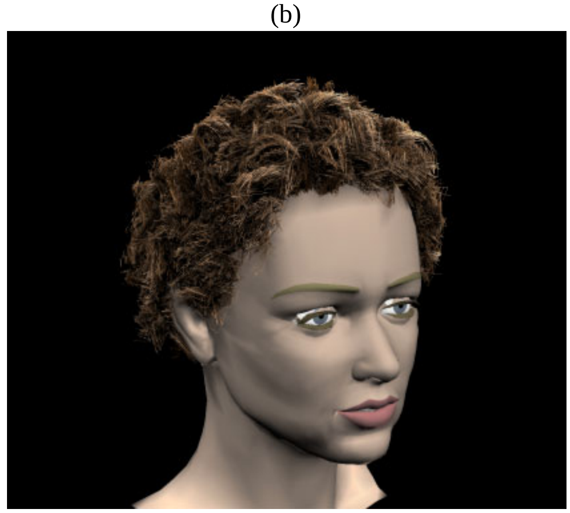
Figure 5. (a) A simplistic initial hair model extracted from a vector field with only two primitives; (b) an improved version after modulating the intial model in (a) with random offset functions at a fine scale; (c) another version by modulating the model in (a) with random offset functions at a coarse scale. (b) and (c) look quite different because of different offset functions; (d) the appearance of the model in (b) is improved by further clustering nearby hairs;

(a)

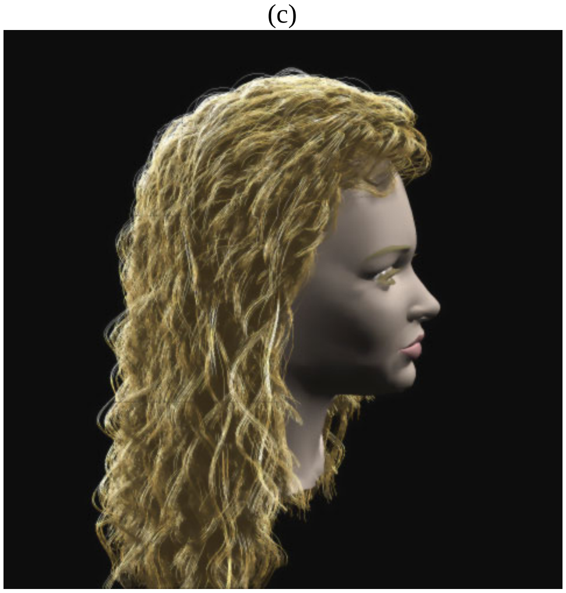
(b)



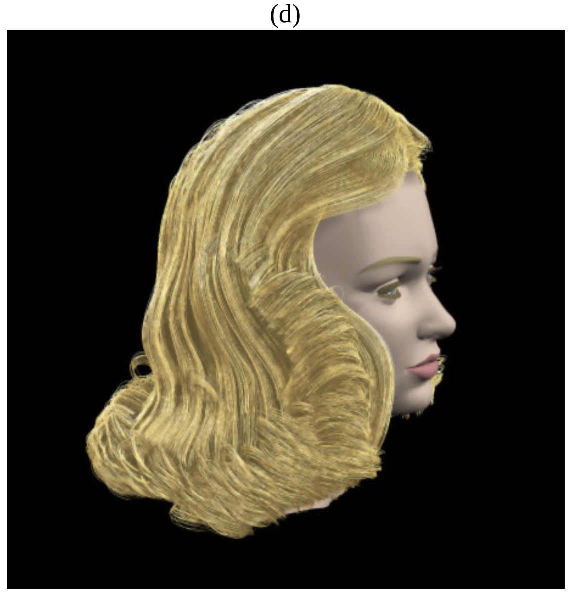
(c)



(d)



(e)



(f)

Figure 6. A list of six hairstyles produced from our techniques. These hairstyles have different length distributions and different levels of curliness. (f) is produced from the curved hair sweep in Fig. 3(d).