

# A Generative Sketch Model for Human Hair Analysis and Synthesis

Hong Chen and Song-Chun Zhu

**Abstract**—In this paper, we present a generative sketch model for human hair analysis and synthesis. We treat hair images as 2D piecewise smooth vector (flow) fields and, thus, our representation is view-based in contrast to the physically-based 3D hair models in graphics. The generative model has three levels. The bottom level is the high-frequency band of the hair image. The middle level is a piecewise smooth vector field for the hair orientation, gradient strength, and growth directions. The top level is an attribute sketch graph for representing the discontinuities in the vector field. A sketch graph typically has a number of sketch curves which are divided into 11 types of directed primitives. Each primitive is a small window (say  $5 \times 7$  pixels) where the orientations and growth directions are defined in parametric forms, for example, hair boundaries, occluding lines between hair strands, dividing lines on top of the hair, etc. In addition to the three level representation, we model the shading effects, i.e., the low-frequency band of the hair image, by a linear superposition of some Gaussian image bases and we encode the hair color by a color map. The inference algorithm is divided into two stages: 1) We compute the undirected orientation field and sketch graph from an input image and 2) we compute the hair growth direction for the sketch curves and the orientation field using a Swendsen-Wang cut algorithm. Both steps maximize a joint Bayesian posterior probability. The generative model provides a straightforward way for synthesizing realistic hair images and stylistic drawings (rendering) from a sketch graph and a few Gaussian bases. The latter can be either inferred from a real hair image or input (edited) manually using a simple sketching interface. We test our algorithm on a large data set of hair images with diverse hair styles. Analysis, synthesis, and rendering results are reported in the experiments.

**Index Terms**—Hair modeling, hair analysis and synthesis, flow patterns, generative models, orientation field, texture, nonphotorealistic rendering.

## 1 INTRODUCTION

HUMAN hair is a very complex visual pattern where hundreds of thousands of hairs are grouped into strands and wisps in diverse hair styles. Modeling hair appearance is an important and challenging problem in graphics, digital human animation, and nonphotorealistic rendering. It is evident that hair style also plays a rather significant role in human recognition. For example, it often takes a longer amount of time to recognize a friend in a meeting if that person has changed to a very different hair style. In another case, if we keep somebody's hair the same and replace his/her face, there is a good chance that the image will be identified as the original subject by familiar people, especially when the image is small.

In computer graphics, hair acquisition, modeling, and animation have attracted growing interest in recent years [13], [14], [5], [7], [11], [22]. Hair models in graphics are three-dimensional and typically have  $O(10^4 - 10^5)$  hair strands and  $O(10^6)$  line segments connected to a scalp surface. These models often include hair dynamics, friction, lighting effects, and occlusion relations, which are either input through user interfaces [7], [5], [14], [13] or acquired from multiple (say 30 to 40) views using advanced computer vision techniques [17], [22]. Editing and rendering such a complex 3D model typically takes many hours on a computer.

In computer vision, hair is studied as an oriented (flow) field [12], [26] similar to other texture patterns, such as wood grain, flows, and fingerprints. To our knowledge, there has been no explicit model dedicated to human hair in the vision literature. As each element in the orientation field is a periodic angle in  $[0, \pi]$  (a Riemannian space), in contrast to image intensities (Euclidean space) in conventional Markov random field, special metrics are needed in designing filters and computing the diffusion equations [18], [4]. Some other vision work [20] has studied the structure characteristics in the orientation (flow) field analysis, such as node, saddle, star-node, etc.

In this paper, we present a generative sketch model for human hair analysis and synthesis. Our sketch model is a view-based, two-dimensional, symbolic representation which can be encoded totally in the order of  $O(100)$  bytes. It is extremely parsimonious in comparison to the physically-based 3D hair models in graphics. This sketch representation can be inferred from real hair images rapidly (in a few minutes on a PC) and realistic hair images can be synthesized or reconstructed from the sketch representation in a few seconds (see Fig. 9 and Fig. 15). We also provide a user interface to edit the sketch and to generate different hair styles (see Fig. 16).

Our representation is motivated by the following observations and applications.

1. Although the number of hairs is huge ( $O(10^6)$ ), many hairs are occluded and the visible hairs are often thinner than a pixel. It is neither practical nor necessary to infer the position of each individual hair for general vision purposes. As was argued in

• The authors are with the Department of Statistics and Computer Science, University of California, 8125 Math Science Building, Box 951554, Los Angeles, CA 90095. E-mail: {hchen, sczhu}@stat.ucla.edu.

Manuscript received 21 July 2005; revised 24 Nov. 2005; accepted 28 Nov. 2005; published online 11 May 2006.

Recommended for acceptance by G. Sapiro.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0393-0705.

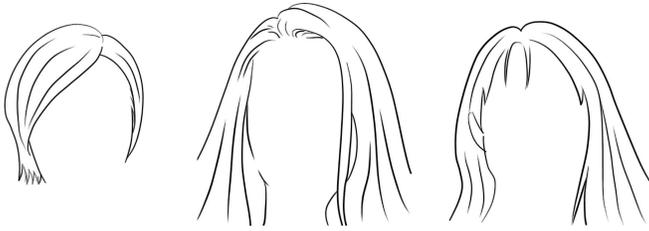


Fig. 1. Three examples of hair sketches drawn by artists.

texture modeling [2] and the information scaling theory [23], human vision perhaps only perceives a general impression of the whole hair style without paying attention to the exact shape of an individual hair. In other words, two hair images are perceptually equivalent if they share some common structural and statistical properties.

2. Artists/painters can capture the essential characteristics of a hair style by a sketch with only a small number (10 to 20) of strokes (See Fig. 1). In this paper, we demonstrate that realistic hair images can be rendered from simple sketches. The reconstructed images (see Fig. 8g and Fig. 15) are different from the original image, but bear similar perceivable structures.
3. Our compact representation is aimed at a number of vision applications:
  - a. extremely low bit image/video compression for telecommunication and video phones,
  - b. hair style recognition and understanding in human-computer interface,

- c. nonphotorealistic rendering or cartoon animation from video images, and
- d. human portraits [6].

Fig. 2 shows our generative model for both hair analysis and synthesis with an optional user interface for hair input and editing. The generative model has three levels. The bottom level is the high-frequency band of the hair image. The middle level is a piecewise smooth vector field for the hair orientation, gradient strength, and growth directions. The top level is an attribute sketch graph for the discontinuities and features in the vector field. Both the vector field and the sketch graph are directed in the hair growth directions. The curves in the sketch graph are divided into five categories of flow primitives in parametric form. Fig. 4 shows some examples for hair boundaries, occluding lines between hair strands, dividing lines on top of the hair, etc. These parametric primitives are essential for generating clear, sharp discontinuities for the orientation field (See Fig. 8d for example). The shading effects are represented in the low-frequency band of the hair image which is modeled as a linear superposition of Gaussian image bases. The color is represented by a color map after a Luv transform. The color map is a mapping from the gray intensity  $[0, 255]$  to a color.

The vector field and sketch graph are computed in a Bayesian framework which maximizes a posterior probability in two steps. The first step includes a greedy sketch pursuit algorithm for constructing the undirected sketch and a diffusion algorithm for computing the orientation field conditional on the sketch graph. The second step infers the growth directions of the sketches and augments the orientation field to a vector field. The hair directions cannot be decided locally and often need global information. We adopt a

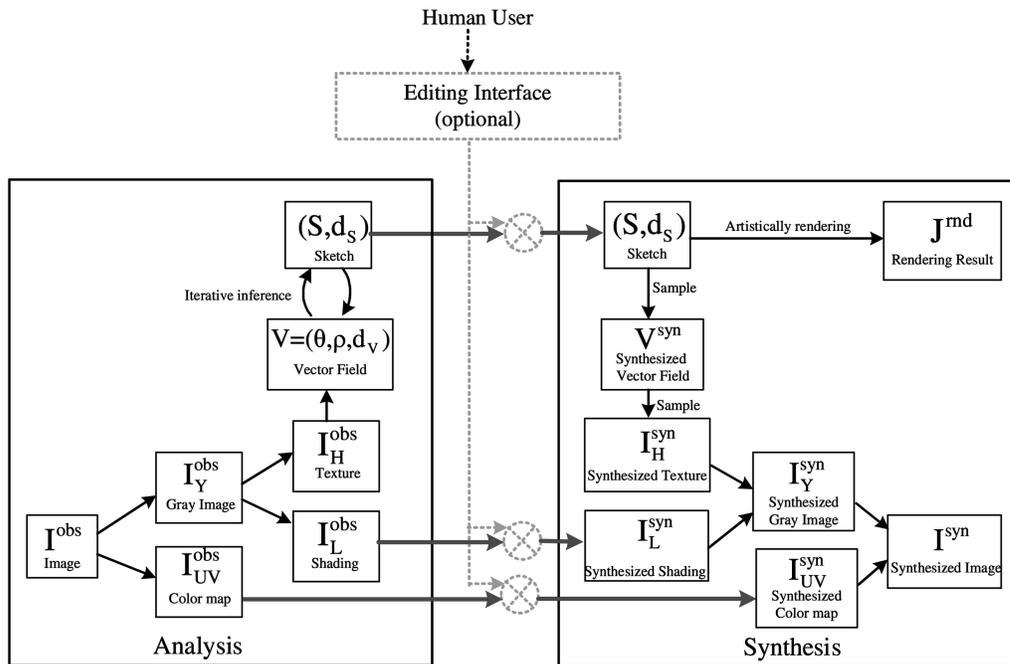


Fig. 2. Overview of our model and algorithm which consists of three modules: analysis, synthesis/reconstruction/rendering, and an optional editing interface. For an input image  $I^{obs}$ , we decompose it into a gray image  $I_Y^{obs}$  and a color channel  $I_{UV}^{obs}$  represented by a color map. The gray image is decomposed into a texture part  $I_H^{obs}$  and a shading part  $I_L^{obs}$  by a low-pass Gaussian filter. From  $I_H^{obs}$ , we compute the vector field  $V$  and the sketch  $S$  with direction  $d_S$ . The synthesis goes from the sketch  $S$  to the vector field  $V^{syn}$  and to the hair image  $I_H^{syn}$ . The latter is combined with the shading  $I_L^{syn}$  and color  $I_{UV}^{syn}$  to produce the final result  $I^{syn}$ . We can render the cartoon sketch in some artistic style  $J^{rnd}$ . The sketch can be input or edited through an interface before entering the synthesis process.

Primitives	(a)	(b)	(c)	(d)	(e)
orientation fields					
vector fields with $(d_l, d_r)$	 (1,1)	 (1,1)	 (1,1) (1,-1)	 (1,1)	 (1,1)
	 (1,-1)	 (1,-1)	 (-1,1) (-1,-1)		 (-1,-1)

Fig. 3. Five primitives for the orientation field  $\Theta$  and 11 primitives for the directed vector field  $\mathbf{V}$  in a dictionary  $\Delta_{\text{sk}}$ . (a) Side boundary. (b) Source (origin) or sink (end) of hair strands. (c) Occluding boundary. (d) Dividing line. (e) Stream line. The line segments and arrows in the primitive windows show the canonical orientations, and the angles may change in  $[-\pi/6, \pi/6]$ . See Fig. 5.

Swendsen-Wang cut algorithm [1] to compute the directions of the sketch graph. We test our algorithm on a large data set of hair images with diverse hair styles. Both analysis, synthesis, editing, and rendering results are reported in the experiments.

Our representation is inspired by the primal sketch model in [10] and the human portrait application in [6]. The latter computes a stylistic cartoon sketch of a human face and hair using an example-based learning method similar to the image analogy work in graphics.

The paper is organized as follows: Section 2 presents the three-level hair model. Section 3 demonstrates that hair images can be synthesized using this generative model. Section 4 discusses the inference algorithm. Section 5 shows some experimental results in both hair analysis, synthesis, editing, and cartoon rendering. Section 6 concludes the work with a discussion of limitations and future work.

## 2 A GENERATIVE SKETCH MODEL OF HAIR

The generative model for both analysis and synthesis is illustrated in Fig. 2. There are three factors contributing to the appearance of hair: 1) hair color, 2) shading effects, and 3) texture. Therefore, a hair image is first decomposed into these three components.

Let  $\mathbf{I}^{\text{obs}}$  denote an observed color hair image. By a Luv transform, we obtain an intensity image  $\mathbf{I}_Y^{\text{obs}}$  and a color channel image  $\mathbf{I}_{UV}^{\text{obs}}$ . The color channel  $\mathbf{I}_{UV}^{\text{obs}}$  is discretized

into a small number of colors and represented by a color map, which maps the gray-scale intensity  $[0, 255]$  of  $\mathbf{I}_Y^{\text{obs}}$  to a color. The intensity image  $\mathbf{I}_Y^{\text{obs}}$  is further decomposed into a low-frequency band  $\mathbf{I}_L^{\text{obs}}$  for illumination and shading with a low-pass Gaussian filter and the remaining high-frequency band is the texture for the hair pattern  $\mathbf{I}_H^{\text{obs}}$ . The low-frequency band is simply represented by a linear superposition of Gaussian image bases plus a mean intensity  $\mu$ ,

$$\mathbf{I}_L^{\text{obs}}(x, y) = \mu + \sum_{i=1}^{K_L} \alpha_i G(x - x_i, y - y_i; \theta_i, \sigma_{x_i}, \sigma_{y_i}) + \text{noise}. \quad (1)$$

Usually,  $K_L = O(10)$  and each Gaussian base is represented symbolically by an ellipses for editing (see Fig. 9). Each ellipse has five parameters, for the center, orientation, and standard deviation along the two axes. The coefficients  $\{\alpha_i\}$  can be positive or negative for highlights and shadows, respectively. The Matching Pursuit algorithm is used to automatically extract the coefficients from the input image. In the editing interface, a user can change the shading by editing the number of ellipses and changing their parameters.

Our study is focused on modeling the texture appearance  $\mathbf{I}_H^{\text{obs}}$  with a three level generative model. A hair texture  $\mathbf{I}_H$  on a lattice  $\Lambda$  is generated by a hidden layer  $\mathbf{V}$ —the vector field for hair growth flow, and  $\mathbf{V}$  is in turn generated by an attribute hair sketch  $\mathbf{S}$  which is a number of sketch curves representing the boundaries of hair strands and wisps with direction  $\mathbf{d}_S$ .

$$\text{Sketch } (\mathbf{S}, \mathbf{d}_S) \xrightarrow{\Delta_{\text{sk}}} \text{Vector field } \mathbf{V} \rightarrow \text{hair image } \mathbf{I}_H.$$

$\Delta_{\text{sk}}$  is a dictionary of sketch primitives shown in Fig. 3. Each primitive is a rectangular window (say  $5 \times 7$  pixels) and some examples are shown in Fig. 4 and Fig. 5.

In the following, we present the three level model in the top-down order as it is shown in the right panel of Fig. 2.

### 2.1 Top Level Representation: The Hair Sketch $\mathbf{S}$

The hair sketch  $\mathbf{S}$  consists of a number of curves denoted by  $C_i$  and each curve represents a long stroke with certain width,

$$\mathbf{S} = (N_C, \{C_i : i = 1, 2, \dots, N_C\}).$$

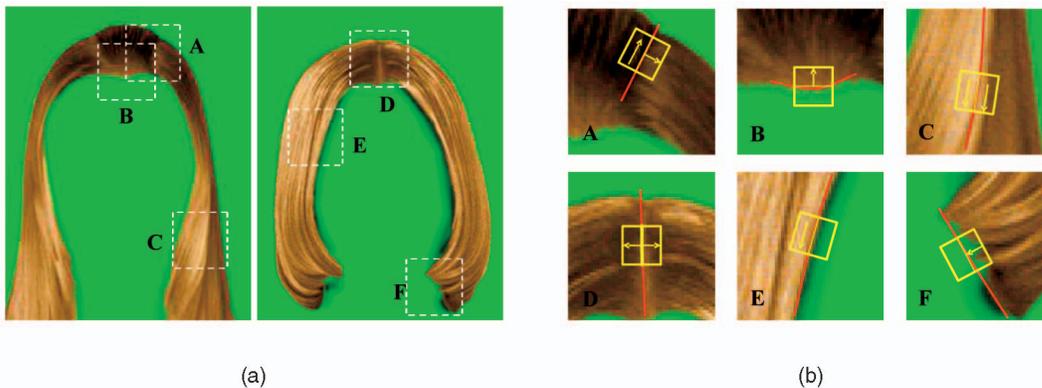


Fig. 4. (a) Windows A-F are six primitive examples. (b) Zoomed-in views of the six windows.

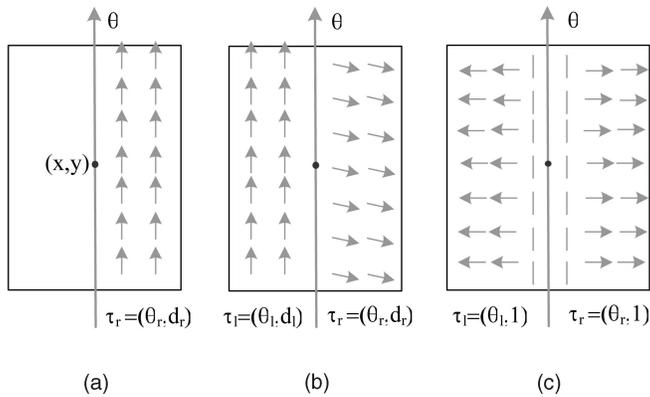


Fig. 5. The windows of three directed primitives. (a) A window of  $5 \times 7$  pixels for a side boundary primitive. (b) A window of  $5 \times 7$  pixels for an occluding boundary primitive. (c) A window of  $6 \times 7$  pixels for a dividing line primitive. Each window has a center  $(x, y)$  and a main axis with orientation  $\theta$ . The left and right side vector fields are specified by the parameters  $\tau_l = (\theta_l, d_l)$  and  $\tau_r = (\theta_r, d_r)$ , respectively. The dividing line primitive is special because its left and right vector fields must grow in opposite directions and there is a middle vector field for the scalp.

These curves are undirected and represent the noticeable structures such as the discontinuities in the hair flow. They consist of a consecutively aligned windows called sketch "primitives." We define five categories of undirected primitives which produce 11 types of directed primitives for the vector field  $\mathbf{V}$ , as shown in Fig. 3. Each primitive specifies the orientations  $\theta_l, \theta_r \in [0, \pi)$  and directions  $d_l, d_r \in \{-1, +1\}$  of the hair flow on the left and right sides of the primitive window. Six examples of the primitives are shown in Fig. 4. These primitives are represented in parametric form for the flow directions and angles on both sides and they are important for generating sharp discontinuities in the vector fields. The five categories of primitives are the following:

1. *Side boundary*: The hair flows along one side and the other side is nonhair. See window E in Fig. 4. Thus, it has only one direction  $d_l$ . Fig. 5a shows an example with a window of  $5 \times 7$  pixels.
2. *Source and sink*: The origin or ending of hair flow. See windows B and F in Fig. 4.
3. *Occluding boundary*: the boundary of two hair strands of different orientations. It often occurs at the places where one hair cluster covers the other. See window A in Fig. 4. Fig. 5b shows its window in the vector field.
4. *Dividing line*: Hair grows in the same orientation but opposite directions. This primitive often occurs at the top the head. See window D in Fig. 4 and Fig. 5c. Note that it has a middle section for the white scalp and the directions  $d_l, d_r$  are fixed, as hair always grow outward from the scalp.

5. *Stream line*: Hair strands with the same orientation and direction but strong contrast in appearance, for example, different dye. See window C in Fig. 4.

Each primitive is represented by a number of variables or denoted by

$$\mathbf{B} = (\ell, x, y, \theta, \rho, (\theta_l, \theta_r)), \quad (2)$$

where  $\ell \in \{a, b, c, d, e\}$  indexes the five primitive types,  $(x, y)$  is the center position,  $\theta$  is the axis orientation in  $[0, 2\pi)$ ,  $\rho$  is the strength of the intensity gradient perpendicular to the axis, and  $(\theta_l, \theta_r)$  are the relative orientation angles at the left and right side vector fields. We discretize the angles so that  $\theta$  has 12-16 orientations. We set  $\theta_l = 0$  and  $\theta_r = 0$  if the orientation on the left side or right side is along the primitive axis. Otherwise,  $\theta_l, \theta_r \in \{\frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}\}$ . Thus, we obtain a dictionary for the undirected primitives.

$$\Delta_{\mathbf{B}} = \left\{ (\ell, x, y, \theta, \rho, (\theta_l, \theta_r)) : \right.$$

$$\left. \ell \in \{a, b, c, d, e\}, (x, y) \in \Lambda, \theta \in \left\{ 0, \dots, \frac{11\pi}{12} \right\}, \forall \theta_l, \theta_r \right\}. \quad (3)$$

The hair growth directions are important for hair understanding and stylistic rendering. It is studied separately from the orientation because the directions have to be inferred from the global image and cannot be decided within local windows. We augment the primitive  $B$  with a direction  $(d_l, d_r)$  specifying the flow directions on the two sides. Thus, we have a sketch primitive dictionary

$$\Delta_{\text{sk}} = \{(\mathbf{B}, d_l, d_r) : \mathbf{B} \in \Delta_{\mathbf{B}}, d_l, d_r \in \{-1, 1\}\}. \quad (4)$$

As Fig. 3 shows, the primitive types (a), (b), and (e) have only one direction variable, the primitive type (d) (the dividing line) is deterministic and, thus, has no direction variables, and only primitive type (c) has two direction variables. For clarity of discussion, we use the uniform notation above.

We denote the direction of the sketch by

$$\mathbf{d}_S = \{(d_{li}, d_{ri}) : i = 1, \dots, N_C\}.$$

We request that all primitives in a sketch curve  $C$  have the same type  $\ell$  and same directions  $\mathbf{d} = (d_l, d_r)$ . Therefore, we denote a directed curve of  $N_{\mathbf{B}}$  primitives by,

$$(C, \mathbf{d}) = (\ell, N_{\mathbf{B}}, \{\mathbf{B}_j : j = 1, 2, \dots, N_{\mathbf{B}}\}, \mathbf{d} = (d_l, d_r)).$$

Fig. 6 shows a curve with a sequence of primitive windows overlapping each other so that the pixels along the curve are covered seamlessly by the primitives.

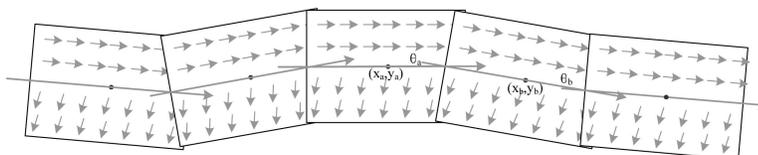


Fig. 6. Pixels along a curve  $C$ .  $C$  consists of  $N_{\mathbf{B}}$  primitives whose windows overlap with neighbors so that the pixels along the curve are covered seamlessly.

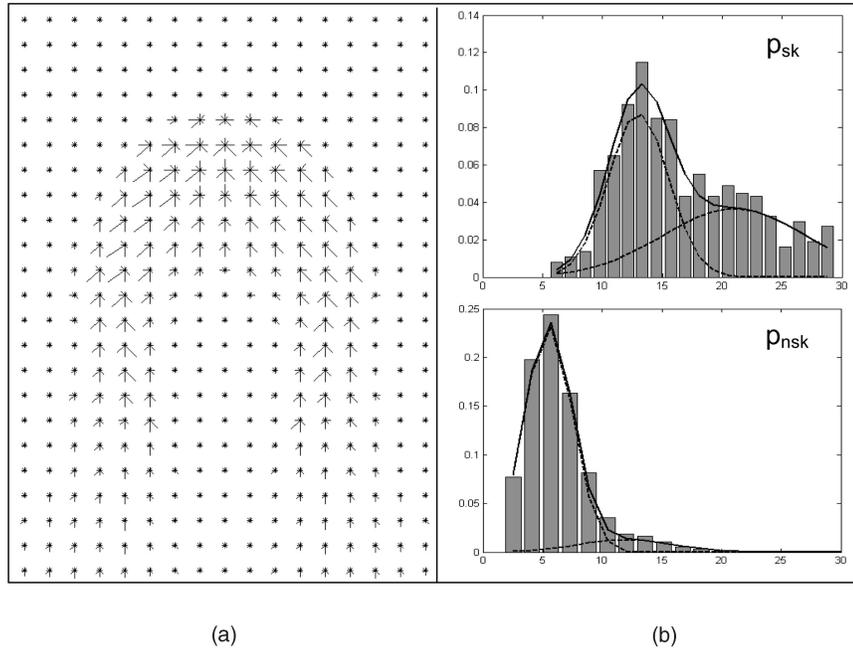


Fig. 7. (a) The empirical point-wise prior probability in the image field  $p_{\text{dir}}(x, y, \theta, d)$  for the flow vector (orientation and direction)  $(\theta, d)$  at each pixel  $(x, y)$ . Longer arrows mean higher probabilities. (b) The empirical prior probability  $p_{\text{sk}}(\rho)$  and  $p_{\text{nsk}}(\rho)$  for the intensity gradient perpendicular to the orientation  $\theta$  for pixels on and off the sketch, respectively. The pixels on the sketch generally have higher intensity gradients and, thus, the histogram has much a heavier tail on the right side.

Suppose the sketch level  $\mathbf{S}$  has a total of  $K$  primitives  $\mathbf{B}_1, \dots, \mathbf{B}_K$  and each primitive  $\mathbf{B}_k$  covers a window  $\Lambda_k$ . The image lattice  $\Lambda$  is divided into two disjoint areas.

$$\Lambda = \Lambda_{\text{sk}} \cup \Lambda_{\text{nsk}}, \quad \Lambda_{\text{sk}} \cap \Lambda_{\text{nsk}} = \emptyset.$$

Pixels in  $\Lambda_{\text{sk}}$  are said to be sketchable or on the sketch and the remaining pixels in  $\Lambda_{\text{nsk}}$  are nonsketchable or off the sketch, according to the terminology used in [10]. The sketchable part is divided as

$$\Lambda_{\text{sk}} = \bigcup_{k=1}^K \Lambda_k.$$

To learn a prior probability model for the sketch level, we collect a set of hair images which are centered in position and normalized in size with manual sketches and vector fields. Then, we compute two types of empirical probabilities (histograms) shown in Fig. 7.

Fig. 7a is a point-wise probability  $p(x, y, \theta, d)$ . At each point  $(x, y)$ , we divide the vector  $(\theta, d)$  (where  $\theta \in [0, \pi)$  and  $d \in \{-1, +1\}$ ) into eight bins (i.e., eight directions in  $[0, 2\pi)$ ) and construct the histogram  $p_{\text{dir}}(x, y, \theta, d)$ . The length of the arrows shows how likely it is that a vector flow will point in a certain direction at a given point.

Fig. 7b shows two empirical probabilities for the intensity gradient  $\rho$  perpendicular to the sketch curves.  $p_{\text{sk}}(\rho)$  and  $p_{\text{nsk}}(\rho)$  are the histograms for pixels on and off the sketch, respectively. Clearly, the gradients are generally larger on the sketch due to high intensity contrast and, therefore,  $p_{\text{sk}}(\rho)$  has a much heavier tail than  $p_{\text{nsk}}(\rho)$ . We fit both histograms by a mixture of two Gaussian distributions.

$$p_{\text{sk}}(\rho) = \omega_{\text{sk}} N(\rho; \mu_{\text{sk1}}, \sigma_{\text{sk1}}) + (1 - \omega_{\text{sk}}) N(\rho; \mu_{\text{sk2}}, \sigma_{\text{sk2}}). \quad (5)$$

We have the following prior probability for the sketch level representation.

$$p(\mathbf{S}, \mathbf{d}_S) = p(N_C) \prod_{i=1}^{N_C} \left\{ \prod_{\mathbf{B}_j \in C_i} [p_{\text{dir}}(x_j, y_j, \theta_{ij}, d_{ij}) p_{\text{dir}}(x_j, y_j, \theta_{rj}, d_{rj})] \prod_{\langle \mathbf{B}_{ja}, \mathbf{B}_{jb} \rangle} G(e(\theta_{ja}, \theta_{jb})) \right\} \prod_{v \in \Lambda_{\text{sk}}} p_{\text{sk}}(\rho(v)). \quad (6)$$

In the above equation,  $p(N_C) \propto e^{-\lambda N_C}$  penalizes the number of curves. For each primitive  $\mathbf{B}_j \in C_i$ , the intensity gradient  $\rho(v)$  on the sketch follows the prior  $p_{\text{sk}}$ , the flow directions  $(\theta_{ij}, d_{ij})$  and  $(\theta_{rj}, d_{rj})$  on both sides follow the prior  $p_{\text{dir}}$ , and  $G()$  is a Gaussian probability on the angle difference  $e(\theta_{ja}, \theta_{jb})$  (see definition in (11)) so that any two consecutive primitives have similar orientations  $\theta_{ja}$  and  $\theta_{jb}$ .

## 2.2 The Middle Level Representation: The Vector Field $\mathbf{V}$

The middle level vector field represents the directed flow of the hair and includes three components  $\mathbf{V} = (\Theta, \rho, \mathbf{d}_V)$  on a lattice  $\Lambda$ .

1. An *orientation field* for the local hair orientation (undirected) in  $[0, \pi)$ ,

$$\Theta = \{\theta(v) : \theta(v) \in [0, \pi), v \in \Lambda\}.$$

2. A *gradient strength field* for the intensity gradient perpendicular to the orientation  $\theta(v)$ ,

$$\rho = \{\rho(v) : \rho(v) = \nabla_{\perp \theta(v)} \mathbf{I}_H, v \in \Lambda\}.$$

3. A *direction field*  $\mathbf{d}$  for the growth direction of the hair at each pixel.

$$\mathbf{d}_V = \{d(v) : d(v) \in \{-1, +1\}, v \in \Lambda\}.$$

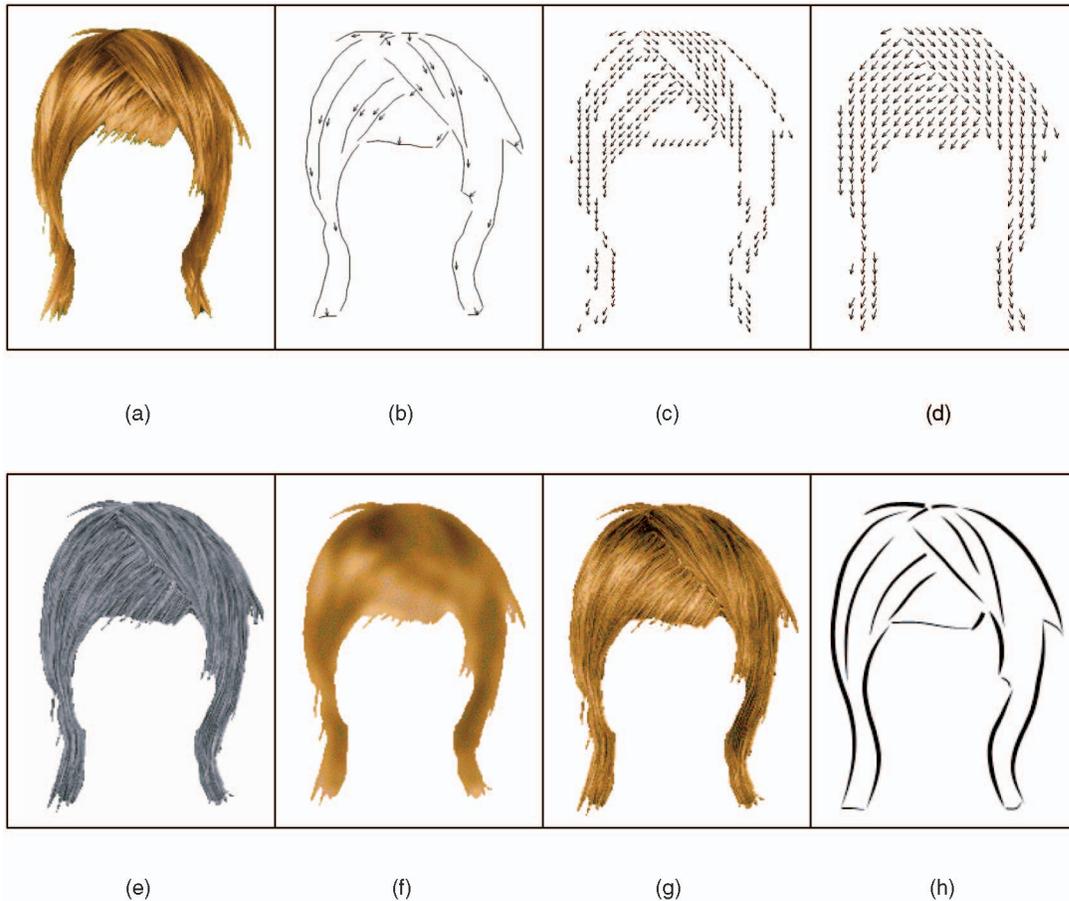


Fig. 8. Example of hair model and inference. (a) is an input color image  $\mathbf{I}^{\text{obs}}$ , (b) is the computed sketch  $\mathbf{S}$  with directions  $\mathbf{d}_S$ , (c) is the sketchable vector field  $\mathbf{V}_{\Lambda_{\text{sk}}}$  generated from  $(\mathbf{S}, \mathbf{d}_S)$ , (d) is the overall vector field  $\mathbf{V}$  after filling-in the nonsketchable part, (e) is the high-frequency hair texture image  $\mathbf{I}_H^{\text{syn}}$  generated from the vector field, (f) is the shading and lighting image, and (g) is the synthesized color image  $\mathbf{I}^{\text{syn}}$  after adding the shading and color. We render an artistic sketch  $\mathbf{J}^{\text{rnd}}$  in (h).

At each point  $v$ , the orientation plus direction decides the flow vector  $\tau(v) \in [0, 2\pi)$  by  $\tau(v) = \theta(v) + \frac{d(v)+1}{2}\pi$ .

The vector field is divided by the sketch graph into two disjoint parts  $\mathbf{V} = (\mathbf{V}_{\text{sk}}, \mathbf{V}_{\text{nsk}})$  for  $\Lambda_{\text{sk}}$  and  $\Lambda_{\text{nsk}}$ , respectively. The generative model from  $(\mathbf{S}, \mathbf{d}_S)$  to  $\mathbf{V}$  is

$$p(\mathbf{V}|\mathbf{S}, \mathbf{d}_S) = p(\mathbf{V}_{\text{nsk}}|\mathbf{V}_{\text{sk}})p(\mathbf{V}_{\text{sk}}|\mathbf{S}, \mathbf{d}_S). \quad (7)$$

$p(\mathbf{V}_{\text{sk}}|\mathbf{S}, \mathbf{d}_S)$  is a Dirac delta function as  $\mathbf{S}, \mathbf{d}_S$  specifies  $\mathbf{V}_{\text{sk}}$  deterministically. Each window  $\Lambda_k$  has a left and a right subwindow with  $\Lambda_k = \Lambda_{lk} \cup \Lambda_{rk}$ , where the vector field is determined by the parameters of the primitive  $\mathbf{B}_k$  and the direction  $\mathbf{d}_k = (\mathbf{d}_{lk}, \mathbf{d}_{rk})$ . Examples are shown in Fig. 5.

$$\begin{aligned} \mathbf{V}(x, y) &= (\theta_{lk}, \rho_k, \mathbf{d}_{lk}), (x, y) \in \Lambda_{lk}, \\ \mathbf{V}(x, y) &= (\theta_{rk}, \rho_k, \mathbf{d}_{rk}), (x, y) \in \Lambda_{rk}, k = 1, \dots, K. \end{aligned} \quad (8)$$

The remaining lattice  $\Lambda_{\text{nsk}}$  corresponds to smooth flow areas where the vector field is “filled-in” from  $\mathbf{V}_{\text{sk}}$  using the following probability:

$$p(\mathbf{V}_{\text{nsk}}|\mathbf{V}_{\text{sk}}) = p(\Theta_{\text{nsk}}, \mathbf{d}_{V_{\text{nsk}}}|\Theta_{\text{sk}}, \mathbf{d}_{V_{\text{sk}}}) \cdot \prod_{v \in \Lambda_{\text{nsk}}} p_{\text{nsk}}(\rho(v)), \quad (9)$$

where  $p_{\text{nsk}}(\cdot)$  is the prior probability shown in Fig. 7b for the gradient strength on pixels off the sketch and  $p(\Theta_{\text{nsk}}, \mathbf{d}_{V_{\text{nsk}}}|\Theta_{\text{sk}}, \mathbf{d}_{V_{\text{sk}}})$  is a smoothness model of the flow field with the sketchable part as its boundary condition.

$$p(\Theta_{\text{nsk}}, \mathbf{d}_{V_{\text{nsk}}}|\Theta_{\text{sk}}, \mathbf{d}_{V_{\text{sk}}}) \propto \exp \left\{ - \sum_{v \in \Lambda_{\text{nsk}}} \sum_{u \in \partial v} \frac{e(\tau(v), \tau(u))^2}{2\lambda_\theta^2} \right\}, \quad (10)$$

where  $\partial v$  is the 4-nearest-neighbor of  $v$  and  $e(\tau(v), \tau(u))$  is the distance between adjacent flow vectors  $\tau(v)$  and  $\tau(u)$  defined on  $[0, 2\pi) \times [0, 2\pi)$ . We adopt the following distance measure metric as in [18], [4].

$$e(\tau(v), \tau(u)) = \sqrt{2(1 - \cos(\tau(v) - \tau(u)))}. \quad (11)$$

Fig. 8 shows an example of the vector field. Fig. 8b is a directed sketch level representation  $(\mathbf{S}, \mathbf{d}_S)$ . Fig. 8c shows the vector field  $\mathbf{V}_{\text{sk}}$  generated by the parameters of the sketch level within the primitive windows. The overall vector field  $\mathbf{V}$  is shown in Fig. 8d after filling in  $\mathbf{V}_{\text{nsk}}$  conditioned on  $\mathbf{V}_{\text{sk}}$ .

### 2.3 The Bottom-Level Representation: The Hair Texture Image $\mathbf{I}_H$

The hair texture image  $\mathbf{I}_H$  is generated by the vector field  $\mathbf{V}$  in the following probability,

$$p(\mathbf{I}_H|\Theta, \rho) \propto \exp \left\{ - \sum_{v \in \Lambda} \frac{(\nabla_{\theta(v)} \mathbf{I}_H)^2}{2\sigma_\theta^2} + \frac{(\|\nabla_{\perp \theta(v)} \mathbf{I}_H\| - \rho(v))^2}{2\sigma_\rho^2} \right\}. \quad (12)$$

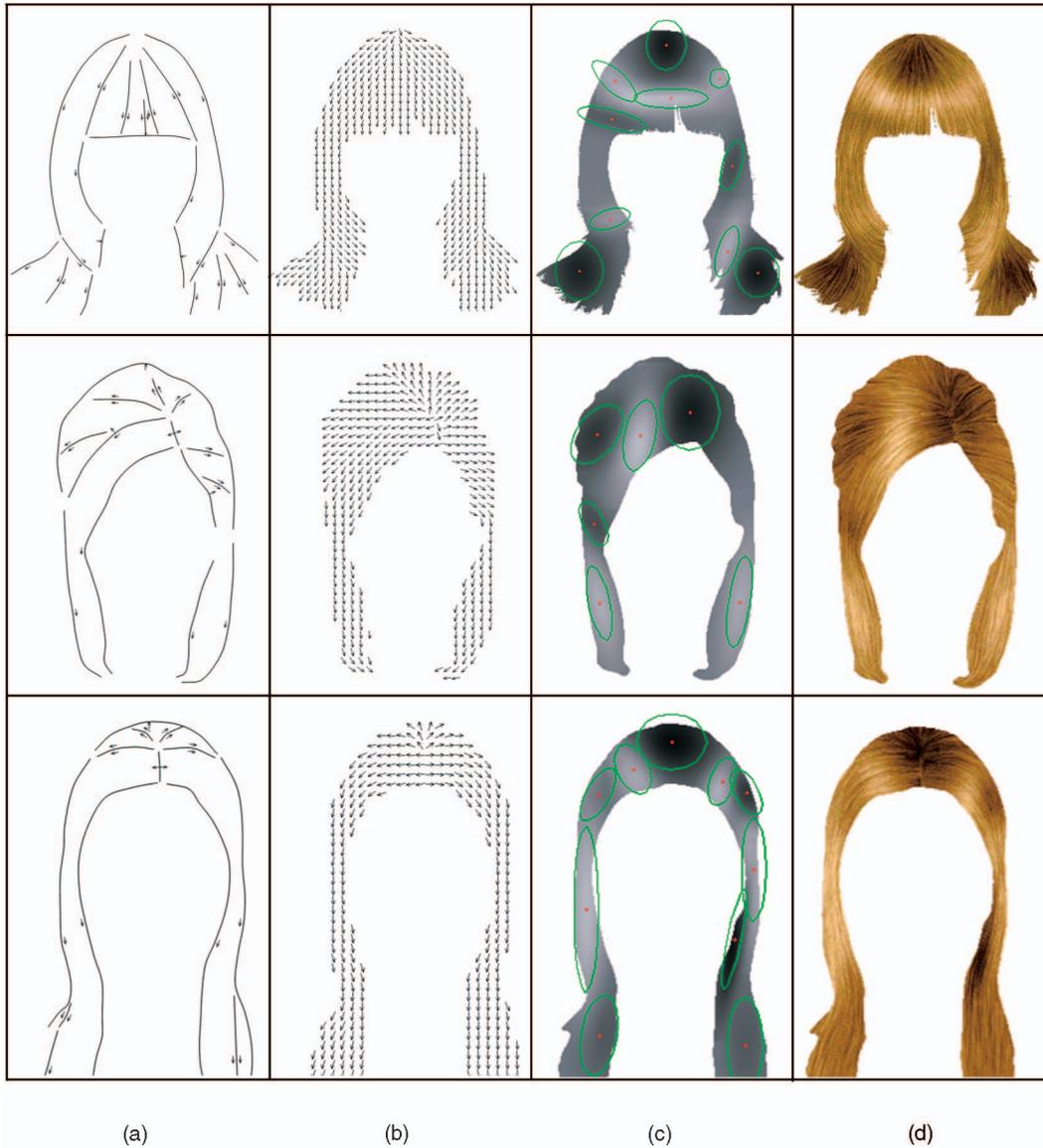


Fig. 9. Three examples of hair drawing and synthesis. (a) Manually input hair sketch  $S$  with directions  $d_S$ . (b) Synthesized vector field  $\mathbf{V}^{\text{syn}}$  given  $(S, d_S)$ . (c) Edited shading maps with a small number of ellipses. (d) Synthesized color images  $\mathbf{I}^{\text{syn}}$ .

Note that the image does not depend on the flow direction  $d_V$ . Intuitively, a hair texture image should have low intensity gradients along the flow orientation  $\theta(v)$ ,  $v \in \Lambda$ , while the gradients in the perpendicular direction  $\perp\theta(v)$  should be close to the gradient  $\rho(v)$  in  $\mathbf{V}$ .

The probability above is an inhomogeneous Markov random field on the image intensity  $\mathbf{I}_H$  modulated by a hidden vector field  $\mathbf{V}$ . This inhomogeneous MRF model has a similar effect to the line integral convolution (LLC) method in flow visualization [3].

Fig. 8e shows an example of the synthesized texture image sampled from the above probability using the vector field in Fig. 8d. With the shading image and color map, it produces a hair image  $\mathbf{I}^{\text{syn}}$  in Fig. 8g.

To summarize the three-level model, we have the following joint probability for the overall representation,

$$p(\mathbf{I}_H, \mathbf{V}, \mathbf{S}, d_S) = p(\mathbf{I}_H | \Theta, \rho) p(\Theta, \rho, d_V | \mathbf{S}, d_S) p(\mathbf{S}, d_S). \quad (13)$$

The three probabilities represent the three level models in (12), (7), and (6), respectively.

### 3 SYNTHESIZING HAIR IMAGES FROM THE GENERATIVE SKETCH MODEL

Following the spirit of analysis-by-synthesis in the texture modeling literature [2], [25], we verify the probability models in the previous section by synthesizing hair images.

Our prior model on the sketch level  $p(\mathbf{S}, d_S)$  is not strong enough for generating hair styles through random sampling, therefore we assume that  $(\mathbf{S}, d_S)$  is either inferred from a hair image in the next section or edited manually through a simple user interface. From  $(\mathbf{S}, d_S)$ , we synthesize a hair image  $\mathbf{I}_H^{\text{syn}}$  in three steps according to the generative model.

1. Synthesizing the vector field from the sketch  $\mathbf{V}^{\text{syn}} \sim p(\mathbf{V} | \mathbf{S}, d_S)$ .
2. Synthesizing the hair texture from the vector field  $\mathbf{I}_H^{\text{syn}} \sim p(\mathbf{I}_H | \mathbf{V}^{\text{syn}})$ .
3. Synthesizing color image  $\mathbf{I}^{\text{syn}}$  by adding a shading image  $\mathbf{I}_L^{\text{syn}}$  to  $\mathbf{I}_H^{\text{syn}}$  and then transferring the gray image to color by the color map. The shading image

is either edited through the interface or inferred from images. One can use various color maps stored for different hairs.

In this section, we briefly mention the sampling processes for Steps 1 and 2.

### 3.1 Synthesis of the Vector Field

Given the directed sketch  $(\mathbf{S}, \mathbf{d}_S)$ , the vector field  $\mathbf{V}_{sk}$  on  $\Lambda_{sk}$  is generated deterministically from the parameters of the sketch primitives according to (8). For  $\mathbf{V}_{nsk}$  on the nonsketchable part  $\Lambda_{nsk}$ , the gradient strengths are sampled iid from the prior model

$$\rho(v) \sim p_{nsk}(\rho(v)), \quad v \in \Lambda_{nsk}. \quad (14)$$

The orientation and direction fields for nonsketchable part are filled in by a diffusion equation derived from minimizing the energy of the smoothness prior in (10). Since the information is propagated from the sketchable part, the pixels near the sketch need to be updated earlier and their values are more confident in the diffusion procedure. Thus, we assign a confidence weight  $w(v)$  for each pixel  $v \in \Lambda$ . At the beginning,  $w(v) = 1$  for  $v \in \Lambda_{sk}$  and  $w(v) = 0$  for  $v \in \Lambda_{nsk}$ . Thus, we modify the energy function in (10) to

$$E(\Theta, \mathbf{d}_V | \mathbf{S}, \mathbf{d}_S) = \sum_{v \in \Lambda_{nsk}} \sum_{u \in \partial v} w(u)(1 - \cos(\tau(u) - \tau(v))). \quad (15)$$

Minimizing the above energy leads to the following diffusion equations for the nonsketchable part:

$$\frac{\partial \tau(v)}{\partial t} = \sum_{u \in \partial v} w(u) \sin(\tau(u) - \tau(v)), \quad \text{for } v \in \Lambda_{nsk}. \quad (16)$$

We compute the diffusion equations by nonlinear Gaussian-Seidel method as in [4] and raise  $w(v)$  in the nonsketchable part gradually at each iteration by a heat diffusion equation with fixed step size.

$$w(v) \leftarrow w(v) + 0.2 \left( \frac{1}{4} \sum_{u \in \partial v} w(u) - w(v) \right), \quad \text{for } v \in \Lambda_{nsk}. \quad (17)$$

The weights converge to  $w(v) = 1$ ,  $\forall v \in \Lambda$ .

### 3.2 Synthesis of Hair Image

The probability  $p(\mathbf{I}_H | \Theta, \rho)$  in (12) has two energy terms. The first is a diffusion term which reduces the gradients along the orientation  $\theta(v)$ , while the second is a ‘‘reaction’’ term which forces the intensity gradient in the direction perpendicular to  $\theta(v)$  to be close to the expected gradient  $\rho(v)$ .  $\rho(v)$  is provided by the vector field. We can sample  $\mathbf{I}_H^{\text{syn}} \sim p(\mathbf{I}_H | \Theta, \rho)$  by the Gibbs sampler, as in texture synthesis [25]. This is quite slow, in practice, so we adopt a much faster method which can generate  $\mathbf{I}_H^{\text{syn}}$  in seconds.

To start with, we define the neighborhood of the inhomogeneous MRF model  $p(\mathbf{I}_H | \Theta, \rho)$ . For each pixel  $v \in \Lambda$ , we trace a 1D curve neighborhood of 3-5 pixels along the orientation  $\theta(v)$  in both directions. The first energy term in  $p(\mathbf{I}_H | \Theta, \rho)$  enforces smoothness on the pixel intensities along the 1D curve neighborhood. This is very similar to the flow visualization algorithm LLC [3]. We initialize the image with white noise and then iterate two steps. The first step is to average the pixel intensities along its 1D curve neighborhood like in the LLC algorithm. The second step is to match the expected gradient strength

along the perpendicular direction. We calculate current gradient strength along the perpendicular direction for the synthesized result of the first step. Then, for each pixel, we scale the local contrast by the rate of the expected gradient strength over the current gradient strength.

Fig. 9 shows three examples of hair synthesis to verify the generative model. Fig. 9a displays the manually input and edited sketches with directions which produce the vector fields in Fig. 9b. Fig. 9c shows the shading image with the ellipses for highlight or dark regions. Fig. 9d is the final synthesized images after a color map transform. Editing and rendering such hair images takes only a few minutes.

## 4 COMPUTATION AND INFERENCE

Given an input image  $\mathbf{I}^{\text{obs}}$ , our goal is to compute the vector field  $\mathbf{V} = (\Theta, \rho, \mathbf{d}_V)$  and the directed sketch  $(\mathbf{S}, \mathbf{d}_S)$  by maximizing the Bayesian posterior defined in Section 2.

$$(\mathbf{S}, \mathbf{d}_S, \mathbf{V})^* = \arg \max p(\mathbf{I}_H^{\text{obs}} | \Theta, \rho) p(\Theta, \rho, \mathbf{d}_V | \mathbf{S}, \mathbf{d}_S) p(\mathbf{S}, \mathbf{d}_S). \quad (18)$$

We choose a two-step greedy algorithm for maximizing the above probability where the two steps minimize different energy terms, respectively.

1. Step 1. Computing the orientation field  $\Theta$  and the undirected sketch  $\mathbf{S}$ . This step can be accomplished by local computation.
2. Step 2. Inferring the growth directions  $\mathbf{d}_S$  and  $\mathbf{d}_V$  for the sketch and orientation field, respectively. The directions have to be inferred with global information on the sketch  $\mathbf{S}$  and are important for hair understanding and cartoon rendering.

The gradient strength field  $\rho$  is computed from the orientation field  $\Theta$  and  $\mathbf{I}_H^{\text{obs}}$  deterministically with  $\rho(v) = \nabla_{\perp \theta(v)} \mathbf{I}_H^{\text{obs}}(v)$ ,  $\forall v \in \Lambda$ .

### 4.1 Step 1. Computing the Orientation Field $\Theta$ and Undirected Sketch $\mathbf{S}$

We transfer the probabilities in (18) into energy functions on the orientation field and undirected sketch,

$$\begin{aligned} (\Theta, \mathbf{S})^* &= \arg \min E(\Theta, \mathbf{S}) \\ &= \arg \min E_1(\mathbf{I}_H^{\text{obs}} | \Theta) + E_2(\Theta, \rho | \mathbf{S}) + E_3(\mathbf{S}). \end{aligned} \quad (19)$$

The three energy functions  $E_1$ ,  $E_2$ , and  $E_3$  are derived from (12), (10), and (6), respectively, by omitting the terms involving the directions  $\mathbf{d}_V, \mathbf{d}_S$ .

$$E_1(\mathbf{I}_H^{\text{obs}} | \Theta) = \sum_{v \in \Lambda} \frac{(\nabla_{\theta(v)} \mathbf{I}_H^{\text{obs}})^2}{2\sigma_\theta^2}, \quad (20)$$

$$\begin{aligned} E_2(\Theta, \rho | \mathbf{S}) &= \sum_{v \in \Lambda_{nsk}} \sum_{u \in \partial v} \frac{e(\theta(v), \theta(u))^2}{2\lambda_\theta^2} - \sum_{v \in \Lambda_{nsk}} \log p_{nsk}(\rho(v)) \\ &\quad - \sum_{v \in \Lambda_{sk}} \log p_{sk}(\rho(v)), \end{aligned} \quad (21)$$

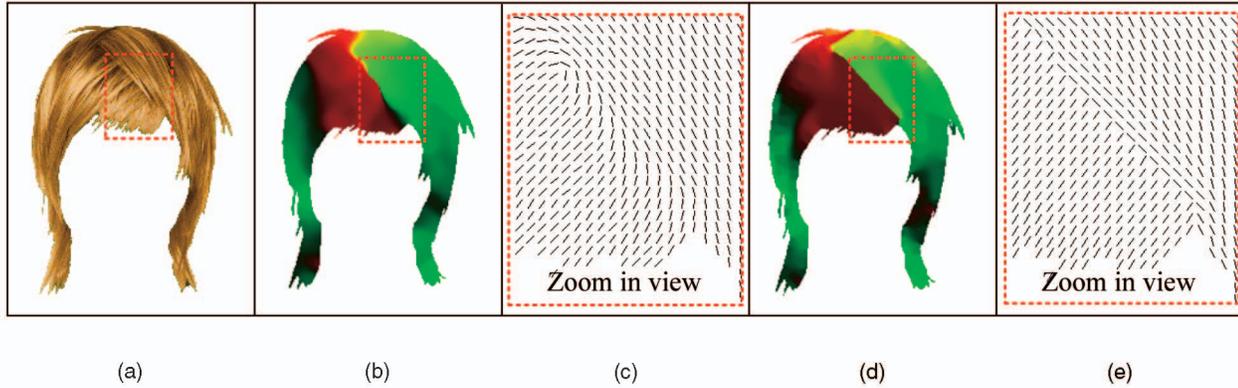


Fig. 10. Comparison of the orientation field computed by diffusion with and without the sketch. (a) The input image. (b) Orientation field  $\Theta$  computed with  $\mathbf{S} = \emptyset$ . We visualize the angle  $\theta \in [0, \pi)$  by color. (c) The zoomed-in view of the orientation field within the dashed window in (b). The orientation field has blurred boundary. (d) Orientation field  $\Theta$  computed together with the sketch  $\mathbf{S}$ . (e) The zoomed-in view of the orientation field within the dashed window in (d). The orientation field has sharp discontinuities along the occluding hair boundary.

$$E_3(\mathbf{S}) = \lambda_{N_C} N_C + \sum_{C_i \in \mathbf{S}} \sum_{\langle \mathbf{B}_{ia}, \mathbf{B}_{ib} \rangle} \frac{e(\theta_{ia}, \theta_{ib})^2}{2\lambda_\theta^2}. \quad (22)$$

In energy  $E_2(\Theta, \rho, \mathbf{S})$ , as the orientations are in  $[0, \pi)$ , we need to renormalize the metric in (11) [18], [4].

$$e(\theta(v), \theta(u)) = \sqrt{\frac{1}{2}(1 - \cos(2\theta(v) - 2\theta(u))).} \quad (23)$$

The second and third terms in  $E_2(\Theta, \rho, \mathbf{S})$  are responsible for pursuing the streamlines. The streamline primitives are distinguished from the nonsketchable part not by their orientations but by their relatively high gradients  $\rho(v)$ . Drawing a sketch curve on the high gradient areas will reduce  $E_2$ .

We initialize  $\mathbf{S} = \emptyset$ . We then add one sketch primitive  $\mathbf{B}_+$  at each iteration  $\mathbf{S}_+ = \mathbf{S} \cup \{\mathbf{B}_+\}$  and recalculate the orientation field by updating  $\Theta$  to  $\Theta_+$ . The primitive  $\mathbf{B}_+$  either starts a new curve  $C_+$  or extends an existing curve  $C_i$  in  $\mathbf{S}$ .  $\mathbf{B}_+ \in \Delta_{\mathbf{B}}$  is selected from a set of primitives in the dictionary  $\Delta_{\mathbf{B}}$  (see (3)) with various locations, orientation, and types so that it achieves a maximum reduction of the energy  $E(\Theta, \mathbf{S})$  until the reduction amount is zero.

$$\mathbf{B}_+^* = \arg \max_{\mathbf{B}_+ \in \Delta_{\mathbf{B}}} E(\Theta, \mathbf{S}) - E(\Theta_+, \mathbf{S}_+). \quad (24)$$

For the current sketch  $\mathbf{S}$  and updated sketch  $\mathbf{S}_+$ , we need to compute the optimal orientation fields  $\Theta$  and  $\Theta_+$ , respectively, by diffusion equations in order to evaluate  $E(\Theta, \mathbf{S})$  and  $E(\Theta_+, \mathbf{S}_+)$ . In the following, we first present the computation of  $\Theta$  given  $\mathbf{S}$ . This is similar to the diffusion of the vector field in the synthesis step, except that the orientation field is influenced by both the sketch and the observed image. Second, we present the algorithm for adding the primitives in a spirit similar to matching pursuit [15].

#### 4.1.1 Computing the Orientation Field by Diffusion

As the sketch  $\mathbf{S}$  determines the orientation field  $\Theta_{\text{sk}}$  in the sketch area  $\Lambda_{\text{sk}}$  ((8)), we only need to compute  $\Theta_{\text{nsk}}$  by minimizing  $E_1(\mathbf{I}_{\mathbf{H}}^{\text{obs}} | \Theta) + E_2(\Theta | \mathbf{S})$  (we omit  $\rho$  for clarity in this section). Therefore, the orientation field  $\Theta_{\text{nsk}}$  should achieve small image gradients  $\nabla_{\theta(v)} \mathbf{I}_{\mathbf{H}}^{\text{obs}}$  and align smoothly with  $\Theta_{\text{sk}}$  in the undirected primitives.

From the image  $\mathbf{I}_{\mathbf{H}}^{\text{obs}}$ , we calculate its intensity gradient

$$s(v) = \sqrt{|\nabla_x \mathbf{I}_{\mathbf{H}}^{\text{obs}}|^2 + |\nabla_y \mathbf{I}_{\mathbf{H}}^{\text{obs}}|^2}$$

$$\theta_o(v) = \arctan\left(\frac{\nabla_y \mathbf{I}_{\mathbf{H}}^{\text{obs}}}{\nabla_x \mathbf{I}_{\mathbf{H}}^{\text{obs}}}\right), \quad \forall v \in \Lambda.$$

We rewrite the gradient at direction  $\theta$  in the following form:

$$\begin{aligned} (\nabla_{\theta(v)} \mathbf{I}_{\mathbf{H}})^2 &= (\nabla_x \mathbf{I}_{\mathbf{H}} \cos \theta(v) + \nabla_y \mathbf{I}_{\mathbf{H}} \sin \theta(v))^2 \\ &= s(v)^2 (\cos \theta_o(v) \cos \theta(v) + \sin \theta_o(v) \sin \theta(v))^2 \\ &= s^2(v) \sin^2(\theta(v) - \theta_o(v)) \\ &= \frac{s^2(v)}{2} (1 - \cos(2\theta(v) - 2\theta_o(v))). \end{aligned}$$

Therefore, the energy function  $E_1(\mathbf{I}_{\mathbf{H}}^{\text{obs}} | \Theta) + E_2(\Theta | \mathbf{S})$  becomes,

$$\begin{aligned} &\sum_{v \in \Lambda_{\text{nsk}}} \frac{s^2(v)}{4\sigma_\theta^2} (1 - \cos(2\theta_o(v) - 2\theta(v))) \\ &+ \frac{1}{2} \sum_{v \in \Lambda_{\text{nsk}}} \sum_{u \in \partial v} \frac{1}{2} (1 - \cos(2\theta(u) - 2\theta(v))). \end{aligned}$$

The equation above can be minimized by the following diffusion equations:

$$\begin{aligned} \frac{d\theta(v)}{dt} &= -\frac{s^2(v)}{4\sigma_\theta^2} \sin(2\theta_o(v) - 2\theta(v)) \\ &- \frac{1}{2} \sum_{u \in \partial v} \sin(2\theta(u) - 2\theta(v)), \quad v \in \Lambda_{\text{nsk}}. \end{aligned} \quad (25)$$

We use the nonlinear Gauss-Seidel method [4] to solve for  $\Theta$ . Once the diffusion equations converge, the energy  $E(\Theta, \mathbf{S})$  measures the goodness of the orientation field  $\Theta$  and sketch  $\mathbf{S}$  and is used in the primitive pursuit algorithm below.

#### 4.1.2 Pursuing the Primitives

In the initial stage  $\mathbf{S} = \emptyset$ , the orientation field  $\Theta$  computed by the diffusion equation has blurry hair boundaries. The sketch  $\mathbf{S}$  plays an important role in computing the orientation field. Figs. 10b and 10c shows a contrasting example of the orientation field computed with and without the primitives.

Suppose  $\mathbf{B}_+ \in \Delta_{\mathbf{B}}$  is a candidate primitive with window  $\Lambda_+$  and  $\mathbf{S}$  is the current sketch. The fitness of  $\mathbf{B}_+$  is evaluated by three energy terms in  $E(\Theta, \mathbf{S})$ .

First,  $\mathbf{B}_+$  should fit well with the current sketch according to energy  $E_3(\mathbf{S})$ . If it starts a new curve, the energy will be increased in  $\lambda_{N_C}$ . Otherwise, it extends a current curve with a smoothness energy  $e(\theta_{ia}, \theta_{ib})^2$ .

Second, adding  $\mathbf{B}_+$  will affect the overall orientation field in the nonsketchable part  $\Theta_{\text{nsk}+}$  through the diffusion equation (25), which minimizes  $E_1(\mathbf{I}_H^{\text{obs}}|\Theta_+) + E_2(\Theta_+, \rho_+|\mathbf{S}_+)$  on  $\Lambda_{\text{nsk}}$ . This is too expensive to compute by running the diffusion equations for each new primitive, so we simply set the orientation fields of the primitive to the current orientation fields  $\Theta$  as an approximation of  $\Theta_+$ .

Third,  $\mathbf{B}_+$  should fit well with the image appearance in the local window. As  $\mathbf{B}_+$  determines the orientation field  $\Theta_+$  within its window  $\Lambda_+$ , it should be aligned well with the minimum image gradient directions in terms of the energy term

$$\sum_{v \in \Lambda_+} \frac{(\nabla_{\theta(v)} \mathbf{I}_H^{\text{obs}})^2}{2\sigma_{\theta}^2}.$$

This energy is part of  $E_1(\mathbf{I}_H^{\text{obs}}|\Theta)$  (see (20)) over window  $\Lambda_+$ . In the following, we shall focus on the computation of

$$\sum_{v \in \Lambda_+} \frac{(\nabla_{\theta(v)} \mathbf{I}_H^{\text{obs}})^2}{2\sigma_{\theta}^2},$$

as other energy terms have been discussed before.

As the dictionary  $\Delta_{\mathbf{B}}$  includes five types of primitives at all locations, orientations, and angle variations, directly computing the summation  $\sum_{v \in \Lambda_+} (\nabla_{\theta(v)} \mathbf{I}_H^{\text{obs}})^2$  for all possible windows is quite expensive. We compute the following quantities once at the beginning and then use them repeatedly in later computation.

At each point  $(x, y) \in \Lambda$ , we compute a matrix (tensor),

$$T(v) = \begin{pmatrix} \nabla_x \mathbf{I}_H^{\text{obs}} \nabla_x \mathbf{I}_H^{\text{obs}} & \nabla_x \mathbf{I}_H^{\text{obs}} \nabla_y \mathbf{I}_H^{\text{obs}} \\ \nabla_y \mathbf{I}_H^{\text{obs}} \nabla_x \mathbf{I}_H^{\text{obs}} & \nabla_y \mathbf{I}_H^{\text{obs}} \nabla_y \mathbf{I}_H^{\text{obs}} \end{pmatrix}. \quad (26)$$

Due to the properties of the tensor, for the window  $\Lambda_0$  which has any constant orientation  $\theta$ , we can compute the summation of  $(\nabla_{\theta} \mathbf{I}_H^{\text{obs}})^2$  by the summation of the tensor,

$$\sum_{v \in \Lambda_0} (\nabla_{\theta} \mathbf{I}_H^{\text{obs}})^2 = (\cos \theta, \sin \theta) \left( \sum_{v \in \Lambda_0} T(v) \right) (\cos \theta, \sin \theta)'. \quad (27)$$

From the definition of the primitives, each primitive is composed of a few (less than three) subwindows with constant orientations. Therefore, we compute and store the summation of the tensor for all kinds of subwindows. Then, the fitness of each primitive can be computed efficiently by combining the fitness energy in the subwindows.

To summarize Step 1, we have the following algorithm for computing the undirected sketch  $\mathbf{S}$  and the orientation field  $\Theta$  from the hair texture image  $\mathbf{I}_H^{\text{obs}}$ :

#### Step 1: Algorithm for the undirected sketch $\mathbf{S}$ and orientation field $\Theta$

0. Given hair texture image  $\mathbf{I}_H^{\text{obs}}$  and primitive dictionary  $\Delta_{\mathbf{B}}$ .

1. Initialize  $\mathbf{S} = \emptyset$ ,  $\Lambda_{\text{sk}} \leftarrow \emptyset$ ,  $N_C \leftarrow 0$ ,  $\Lambda_{\text{nsk}} \leftarrow \Lambda$ .
2. Compute the orientation field  $\Theta_{\text{nsk}}$  by diffusion (25).
3. Compute the current total energy function  $E(\Theta, \mathbf{S})$  in (19).
3. Iterate the following steps.
  4. Birth of a new sketch
  5. For any  $\mathbf{B}_+ \in \Delta_{\mathbf{B}}$ , compute  $\delta(\mathbf{B}_+) = E(\Theta, \mathbf{S}) - E(\Theta_+, \mathbf{S} \cup \{\mathbf{B}_+\})$
  6. Choose  $\mathbf{B}_+^* = \arg \max_{\mathbf{B}_+ \in \Delta_{\mathbf{B}}} \delta(\mathbf{B}_+)$ .
  7. If  $\delta(\mathbf{B}_+^*) \leq 0$ , exit. Output  $\mathbf{S}^*$  and  $\Theta^*$ .
  8. Update  $C_i = \mathbf{B}_+^*$ ,  $N_C \leftarrow N_C + 1$ ,  $\Lambda_{\text{nsk}} \leftarrow \Lambda_{\text{nsk}} \setminus \Lambda_+^*$ ,  $\Lambda_{\text{sk}} \leftarrow \Lambda_{\text{sk}} \cup \Lambda_+^*$ .
  9. Trace the sketch curve  $C_{N_C}$ :
  10. For  $\mathbf{B}_+$  at the two ends of  $C_{N_C}$ , compute  $\delta(\mathbf{B}_+) = E(\Theta, \mathbf{S}) - E(\Theta_+, \mathbf{S} \cup \{\mathbf{B}_+\})$ ,
  11. Choose  $\mathbf{B}_+^* = \arg \max \delta(\mathbf{B}_+)$ .
  12. If  $\delta(\mathbf{B}_+^*) \leq 0$ , go to Step 4.
  13. Update  $C_{N_C} \leftarrow C_{N_C} \cup \{\mathbf{B}_+^*\}$ ,  $\Lambda_{\text{nsk}} \leftarrow \Lambda_{\text{nsk}} \setminus \Lambda_+^*$ ,  $\Lambda_{\text{sk}} \leftarrow \Lambda_{\text{sk}} \cup \Lambda_+^*$ . Goto Step 9.

The sketch pursuit algorithm above resembles the matching pursuit algorithm [15] in signal decomposition with wavelets. The matching pursuit algorithm adds one wavelet base at a time to achieve maximum reduction of reconstruction error. Our primitives are disjoint rather than linear additive. Furthermore, the orientation field is a layer of hidden variables that have to be computed iteratively. All these factors make the computation more complicated than matching pursuit.

## 4.2 Step 2: Computing the Hair Growth Directions

In the second step, we compute the hair growth directions  $\mathbf{d}_S$  and  $\mathbf{d}_V$  for the sketch  $\mathbf{S}$  and orientation field  $\Theta$ , respectively. Suppose  $\mathbf{S} = \{C_i : i = 1, \dots, N_C\}$  has a set of  $N_C$  undirected sketch curves and the primitives on each curve share the same direction labels. The five primitives (a)-(e) have 1, 1, 2, 2, 1 direction variables, respectively. For notation clarity, we pool all the direction variables together and denote,

$$\mathbf{d}_S = (d_1, d_2, \dots, d_m) \in \{-1, 1\}^m. \quad (28)$$

Fig. 11a shows an example with a number of curves. We represent each variable by a node and form an adjacency graph in Fig. 11b. Two nodes  $d_i, d_j$  are said to be adjacent if they belong to two different curves and there is a straight line connecting the two curves without intersecting other curves.

Therefore, the problem of inferring the direction becomes a binary Graph labeling/coloring problem.

### 4.2.1 The Energy Terms Related to Directions

In the maximum posterior probability formulation in (18), the direction is not involved in the image model  $p(\mathbf{I}_H^{\text{obs}}|\Theta, \rho)$ . Only two energy terms contribute to the hair directions. The first term is included in  $p(\mathbf{V}|\mathbf{S}, \mathbf{d}_S)$  and it is the smoothness of vector flow field  $\tau(v), v \in \Lambda_{\text{sk}}$  for the nonsketchable part conditioned on the sketch part  $\tau(v), v \in \Lambda_{\text{nsk}}$  (see (10)). The second term is included in the prior probability  $p(\mathbf{S}, \mathbf{d}_S)$ , where the directions within the primitive windows follow a prior probability  $p_{\text{dir}}(v, \theta(v), d(v))$ . An additional source of information for determining the directions comes from the dividing line primitives whose directions are known.

Given  $\Theta$  and  $\mathbf{S}$ , we transfer (18) to the following energy minimization problem:

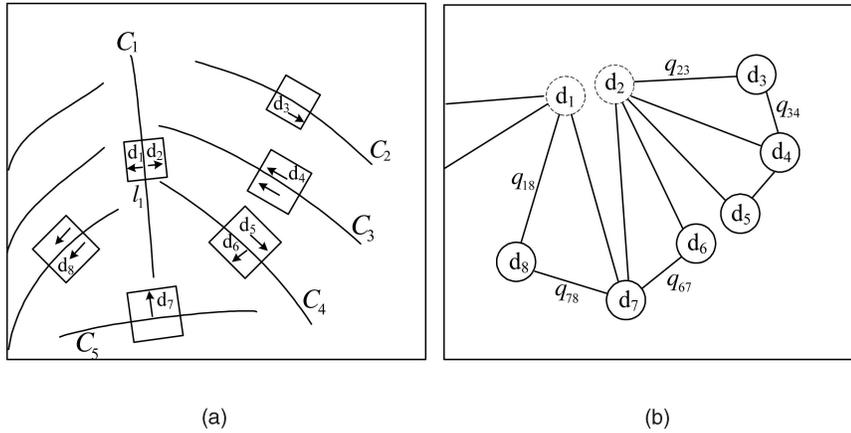


Fig. 11. (a) A portion of a sketch with a number of curves. Each curve has 1-2 direction variables. (b) Each direction variable in (a) is represented by a node/vertex in the adjacency graph. Two adjacent nodes (curves) have a weighted edge for their direction compatibility constraints. The compatibility probability  $q(d_i, d_j)$  is abbreviated by  $q_{ij}$ .

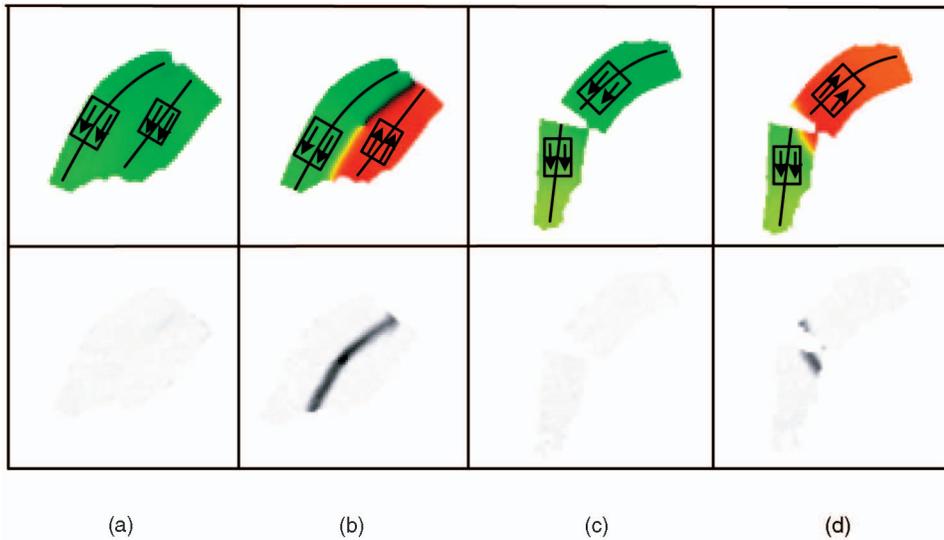


Fig. 12. Direction compatibility between two adjacent streamline curves. Row 1 shows the vector fields in color and row two shows the energy tension in the vector fields with darker spots for high energy. Intuitively, the two curves in (a) and (b) have compatible directions and, thus, lower energies. (b) and (d) are not compatible and have high energies.

$$(\mathbf{d}_S, \mathbf{d}_V)^* = \arg \min E(\mathbf{d}_S, \mathbf{d}_V | \Theta, \mathbf{S}) \quad (29)$$

$$= \arg \min \sum_{v \in \Lambda_{\text{nsk}}} \sum_{u \in \partial v} (1 - \cos(\tau(u) - \tau(v))) - \sum_{v \in \Lambda_{\text{sk}}} \log p_{\text{dir}}(v, \theta(v), d(v)). \quad (30)$$

$d(v)$ ,  $v \in \Lambda_{\text{sk}}$  is determined by  $\mathbf{d}_S$ , and the nonsketchable part  $d(v)$ ,  $v \in \Lambda_{\text{nsk}}$  is computed through the diffusion (25) which minimizes the first energy term above. Therefore, the computation of  $\mathbf{d}_S$  is the key to solving the equation and  $\mathbf{d}_S$  is evaluated by the smoothness of the vector field  $\mathbf{d}_V$  in the above equation.

There are several possible algorithms for solving the graph labeling problem. For example, Belief propagation and graph cut are fast solutions and the Gibbs sampler [8] is a general algorithm. However, in this problem, Belief propagation and Graph cut are not applicable to the general energy functions above and the Gibbs sampler is inefficient because of the strong coupling in the labels. In the following, we adopt a Swendsen-Wang cut algorithm [1] which can simultaneously flip the labels of several nodes.

#### 4.2.2 The Swendsen-Wang Cut Algorithm for Labeling the Adjacent Graph

For each link  $e = \langle d_i, d_j \rangle$  in the adjacency graph, we define a local compatibility probability  $q_{ij}$  which represents how strongly  $d_i$  and  $d_j$  are correlated based on some local measurements. Fig. 12 shows some examples for two streamline directions. These probabilities  $q_{ij}$  are used for proposing clusters (connected components) in the Swendsen-Wang cut method [1]. As a bottom-up proposal probability, the exact formula for  $q_{ij}$  does not matter too much, though a good probability will lead to fast convergence.

Without loss of generality, suppose  $d_i$  and  $d_j$  are the direction variables on the right and left sides of two curves  $C_m$  and  $C_n$ , respectively. We denote the neighborhood between the two curves by  $\Lambda(d_i, d_j)$ . One may use the heuristic method for computing  $\Lambda(d_i, d_j)$  and we define the energy of directional compatibility as

$$E(d_i, d_j) = \sum_{v \in \Lambda_{\text{nsk}} \cap \Lambda(d_i, d_j)} \sum_{u \in \partial v} (1 - \cos(\tau(u) - \tau(v))).$$

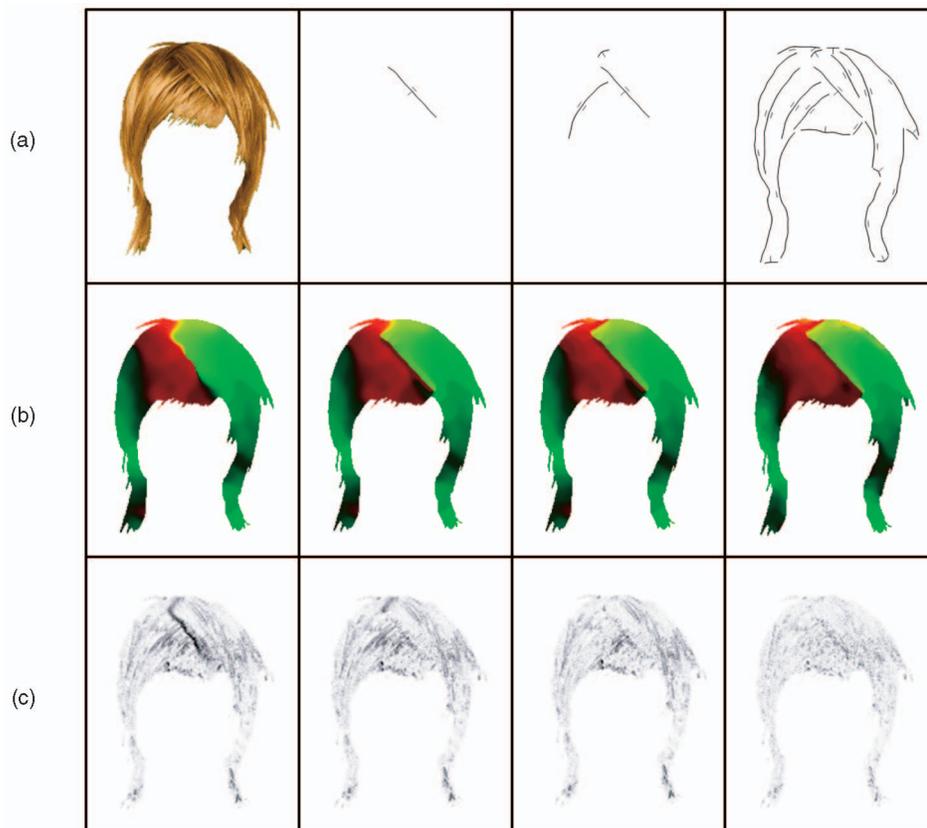


Fig. 13. Sketch pursuit procedure. (a) Inferred sketch. It is a null sketch at the first step. (b) The orientation field. (c) The energy function  $E_1(\mathbf{I}_H^{\text{obs}}|\Theta) + E_2(\Theta, \rho|S)$  at each pixel.

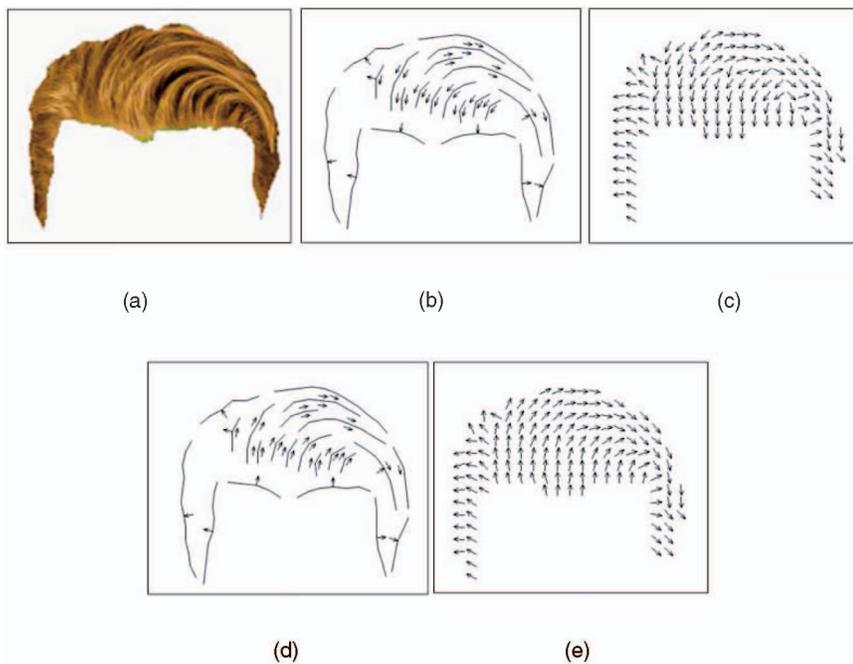


Fig. 14. Inference of hair growth directions. (a) The input image; (b) Sketch  $S$  with direction  $d_S$  computed with the point wise prior  $p_{\text{dir}}$ . Some hairs on the top run in opposite directions. (c) The vector field  $V$  generated by the directed sketch  $(S, d_S)$  in (b). The directions are wrong at some pixels and, therefore, the energy in the vector field is high. (d) Sketch with directions inferred with curve compatibility constraints. (e) The vector field generated by the directed sketch  $(S, d_S)$  in (d).



Fig. 15. Experiments in hair sketching and synthesis on seven hair styles. (a) are the input images, (b) are the computed sketches with growth directions, (c) are the vector fields, (d) are the synthesized images, and (e) are the cartoon renderings in different stroke styles.

As the probability is the same if we flip both  $d_i, d_j$  simultaneously, we normalize the probability,

$$q_{ij} = p(d_i = d_j) = \frac{\exp(-E(d_i = 1, d_j = 1))}{\exp(-E(d_i = -1, d_j = 1)) + \exp(-E(d_i = 1, d_j = 1))}.$$

$d_i$  and  $d_j$  are highly correlated if  $q_{ij}$  is close to 1 or 0 and they are less correlated when  $q_{ij}$  is close to 0.5.

In [1], the Swendsen-Wang cut algorithm generalizes the original Swendsen-Wang algorithm so that it can be applied for general probability models in vision tasks. It samples the posterior probability  $p(\mathbf{d}_S, \mathbf{d}_V | \Theta, \mathbf{S}) \propto e^{-E(\mathbf{d}_S, \mathbf{d}_V | \Theta, \mathbf{S})}$ , where  $E(\mathbf{d}_S, \mathbf{d}_V | \Theta, \mathbf{S})$  is defined in (29). In our experiment, we do not use the simulated annealing schedule which is often needed in optimization. The reason, in our opinion, is that the energy has already rather low temperature and the Swendsen-Wang method is known to mix rapidly in low temperature.

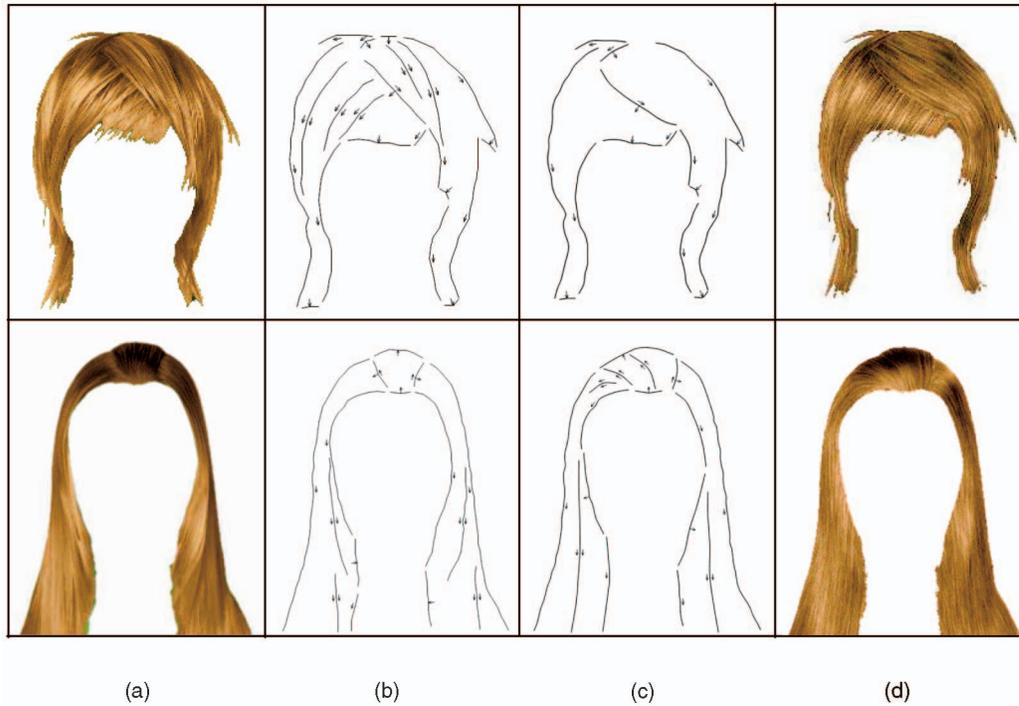


Fig. 16. Examples of hair editing for hair images. (a) Original hair image. (b) Original hair sketch  $S$ . (c) Edited hair sketch. (d) Novel hair styles after editing.

## Step 2: The Swendsen-Wang cut method for direction inference

0. Initialize a labeling  $\mathbf{d}_S = (d_1, d_2, \dots, d_m)$  randomly or uniformly.
1. Iterate the following steps for current  $\mathbf{d}_S$ .
2. For each adjacency link  $e = \langle d_i, d_j \rangle$ ,  
Turn  $e$  off deterministically if  $d_i \neq d_j$ .  
Turn  $e$  off with probability  $1 - q_{ij}$  if  $d_i = d_j$ .
3. Collect a set of connected components  $CP$ .
4. Select a connected component  $V_0 \in CP$  with probability  $q(V|CP)$  (say uniformly).
5. Flip the direction labels in the connected component  $V_0$  from  $\ell$  to  $\ell'$ ,  $\ell, \ell' \in \{-1, 1\}$ .
6. Compute the new labeling  $\mathbf{d}'_S$  and the corresponding  $\mathbf{d}'_V$ .
7. Accept the proposed flip with probability  $\alpha((\mathbf{d}_S, \mathbf{d}_V) \rightarrow (\mathbf{d}'_S, \mathbf{d}'_V))$  as  $\alpha((\mathbf{d}_S, \mathbf{d}_V) \rightarrow (\mathbf{d}'_S, \mathbf{d}'_V)) =$

$$\min \left( 1, \frac{\prod_{e \in \mathcal{C}(V_0, V_{\ell'} \setminus V_0)} (1 - q_{ij}) \cdot \frac{p(\mathbf{d}'_S, \mathbf{d}'_V | \Theta, \mathbf{S})}{p(\mathbf{d}_S, \mathbf{d}_V | \Theta, \mathbf{S})}}{\prod_{e \in \mathcal{C}(V_0, V_{\ell} \setminus V_0)} (1 - q_{ij})} \right).$$

In the above acceptance probability,  $V_{\ell}$  and  $V_{\ell'}$  are the subgraphs with direction labels  $\ell$  and  $\ell'$ , respectively.  $\mathcal{C}(V_0, V_{\ell} \setminus V_0)$  is the cut between  $V_0$  and  $V_{\ell} \setminus V_0$  and  $\mathcal{C}(V_0, V_{\ell'} \setminus V_0)$  is the cut between  $V_0$  and  $V_{\ell'} \setminus V_0$ . We refer to [1] for more details of the Swendsen-Wang cut algorithm.

## 5 EXPERIMENTS

We collected a hair data set containing 300 realistic human hair images extracted from a cosmetic makeup software. The hairs were segmented from human images and pasted on a background of constant color.

For our training examples, we manually labeled the sketches of 20 hair images with distinct styles. From these sketches, we learned the primitives and computed the vector field through diffusion. We learned the two prior probabilities shown in Fig. 7. We also learned a number of parameters, such as the standard deviations used in the representation.

**Experiment 1: Hair sketching and synthesis.** Fig. 8 illustrates the steps in inferring the sketch and synthesis and more results are shown in Fig. 15. A detailed sketch pursuit example is shown in Fig. 13. From the figure, we can see that the energy  $E_1(\mathbf{I}_H^{\text{obs}} | \Theta) + E_2(\Theta, \rho | \mathbf{S})$  is reduced while the new curves are added into the sketch graph.

Our experiments reveal that the sketch level representation plays two important roles in computing the vector field.

1. As Fig. 10 shows, the primitives correct the errors in the initial orientation field  $(\rho, \theta)$ , especially around the discontinuities of flows, which is a well-known problem in filtering or feature extraction in any random fields. On one hand, we need a large window to pool information locally, but, on the other hand, large windows lose spatial resolution. This problem is resolved by fitting a number of parametric primitives in generative windows.
2. The sketch helps to determine the directions which cannot be determined locally. With the sketch, we

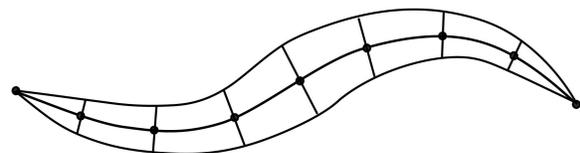


Fig. 17. A brush example for the nonphotorealistic hair rendering.



Fig. 18. Some difficult cases for the current model and inference algorithm. (a) Curly hairs with heavy occlusions of hair strands. (b) Short hairs growing forward. (c) Black hair with weak flow texture.

have high level information of hair structure, such as the dividing line, the source, and sink boundary. Fig. 14 shows a comparison experiment where the compatibility among curves plays an important role in computing the correct directions.

**Experiment 2: Hair editing.** Fig. 16 shows two examples of hair editing. For a given image in Fig. 16a, we compute the sketch graph shown in Fig. 16b. Then, we provide a user interface to edit the sketches. The interface provides a number of operators, such as adding/moving/deleting a curve, changing its directions, editing the shading effects, setting parameters for the intensity contrast, etc. Fig. 16c shows examples of the edited sketch and Fig. 16d displays the novel hair style. These editing steps are more convenient and much faster than the 3D graphics interfaces.

**Experiment 3: Hair rendering.** Rendering cartoon hair is an application of hair analysis and is extremely straightforward using the generative models. All we need to do is replace the sketch curves by stylistic brush drawings.

In nonphotorealistic animation and rendering (NPAR) [9], different drawing styles can be achieved by different types of strokes. In our experiments, we use a simple brush shown in Fig. 17. We sample a number of key points on each sketch curve  $C_i$  and draw a smooth Catmull-Rom spline passing these points as the skeleton of the brush. For each key point on the skeleton, we define a width and get two points along its normal direction. These points determine two Catmull-Rom splines which are the boundaries of the brush stroke. We assume the brush has two narrowed ends. Some examples of hair rendering are shown as  $J^{md}$  in Fig. 8 and Fig. 15. Other styles can be generated with more fancy brushes.

## 6 LIMITATIONS AND FUTURE WORK

The three level generative hair model is extremely compact ( $O(100)$  bytes) and yet it is shown to be effective in synthesizing realistic hair images with diverse hair styles. It also supports convenient hair editing and rendering. The representation is aimed at applications such as extremely low bit image compression in video phones, human hair recognition, and cartoon animation, etc.

The current model is still limited in the following aspects:

1. The model is 2D and, therefore, its vector field does not work well for very curly hairs, as seen in

Fig. 18a. These examples need short sketch curves with occluding relations in the sketch representation. Similar work was studied in a curve process in [21].

2. It cannot work well for very short hairs growing forward (see in Fig. 18b) or black hairs (see in Fig. 18c). The latter have very weak textures for computing the vector field reliably.
3. We work on segmented hair images and we shall integrate the hair sketch with our face sketch work [24] and use the face as context to detect hair.
4. We cannot model more structured hair styles, such as pigtail and braid which will need more sophisticated primitives and sketch graphs.

## ACKNOWLEDGMENTS

This work is supported by grants NIH 5R01-EY013875, NSF IIS-0222967, and a Microsoft gift. The authors would like to thank Dr. Alan Yuille, Dr. Harry Shum, and Dr. Ying Qing Xu for their generous support and many stimulating discussions.

## REFERENCES

- [1] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cuts," *Proc. Int'l Conf. Computer Vision*, pp. 320-327, 2003.
- [2] J.R. Bergen and E.H. Adelson, "Theories of Visual Texture Perception," *Spatial Vision*, 1991.
- [3] B. Cabral and L.C. Leedom, "Imaging Vector Fields Using Integral Convolution," *Proc. 20th Conf. Computer Graphics and Interactive Techniques*, pp. 263-270, 1993.
- [4] T. Chan and J.H. Shen, "Variational Restoration of Non-Flat Image Features: Models and Algorithms," *SIAM J. Applied Math.*, vol. 61, pp. 1338-1361, 2001.
- [5] J.T. Chang, J.Y. Jin, and Y.Z. Yu, "A Practical Model for Hair Mutual Interactions," *Proc. Siggraph/Eurographics Symp. Computer Animation*, 2002.
- [6] H. Chen, Z.Q. Liu, C. Rose, Y.Q. Xu, H.Y. Shum, and D. Salesin, "Example-Based Composite Sketching of Human Portraits," *Proc. Third Int'l Symp. Non-Photorealistic Animation and Rendering*, pp. 95-153, 2004.
- [7] A. Daldengan, N.M. Thalmann, T. Kurihara, and D. Thalmann, "An Integrated System for Modeling, Animation and Rendering Hair," *Proc. Computer Graphics Forum (Eurographics '93)*, pp. 211-221, 1993.
- [8] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 11, pp. 721-741, Nov. 1984.
- [9] B. Gooch and A. Gooch, *Non-Photorealistic Rendering*. A.K. Peters, Ltd., 2001.

- [10] C.E. Guo, S.C. Zhu, and Y.N. Wu, "A Mathematical Theory for Primal Sketch and Sketchability," *Proc. Int'l Conf. Computer Vision*, 2003.
- [11] S. Hadap and N. Magnenat-Thalmann, "Interactive Hair Sytler Based on Fluid Flow," *Proc. Eurographics Workshop Computer Animation and Simulation*, 2000.
- [12] M Kass and A Witkin, "Analyzing Oriented Patterns," *J. Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 362-385, 1987.
- [13] T.Y. Kim and U. Neumann, "Interactive Multi-Resolution Hair Modeling and Editing," *Proc. SIGGRAPH*, 2002.
- [14] D.W. Lee and H.S. Ko, "Natural Hairstyle Modeling and Animation," *Graphical Models*, vol. 63, pp. 67-85, 2001.
- [15] S. Mallat and Z. Zhang, "Matching Pursuit in a Time-Frequency Dictionary," *IEEE Trans. Signal Processing* vol. 41, pp. 3397-3415, 1993.
- [16] M. Nitzberg, D.B. Mumford, and T. Shiota, "Filtering, Segmentation, and Depth," *Lecture Notes in Computer Science*, vol. 662, 1993.
- [17] S. Paris, H.M. Briceno, and F.X. Sillion, "Capture of Hair Geometry from Multiple Images," *Proc. SIGGRAPH*, 2004.
- [18] P. Perona, "Orientation Diffusions," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 457-467, Mar. 1998.
- [19] P. Pérez, A. Blake, and M. Gangnet, "JetStream: Probabilistic Contour Extraction with Particles," *Proc. Int'l Conf. Computer Vision*, pp. 524-531, 2001.
- [20] A.R. Rao and R.C. Jain, "Computerized Flow Field Analysis: Oriented Texture Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 693-709, July 1992.
- [21] Z.W. Tu and S.C. Zhu, "Parsing Images into Region and Curve Processes," *Proc. European Conf. Computer Vision*, June 2002.
- [22] Y. Wei, E. Ofek, L. Quan, and H.Y. Shum, "Modeling Hair from Multiple Views," *Proc. SIGGRAPH*, 2005.
- [23] Y.N. Wu, S.C. Zhu, and C.E. Guo, "From Information Scaling Laws of Natural Images to Regimes of Statistical Models," Preprint 408, Dept. of Statistics, Univ. of California, Los Angeles, 2004.
- [24] Z.J. Xu, H. Chen, and S.C. Zhu, "A High Resolution Grammatical Model for Face Representation and Sketching," *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.
- [25] S.C. Zhu, Y.N. Wu, and D.B. Mumford, "Minimax Entropy Principle and Its Applications to Texture Modeling," *Neural Computation*, vol. 9, no. 8, pp. 1627-1660, Nov. 1997.
- [26] S.W. Zucker, "Early Orientation Selection: Tangent Fields and the Dimensionality of Their Support," *J. Computer Vision, Graphics, and Image Processing*, vol. 32, pp. 74-103, 1985.



computer vision and machine learning.



**Hong Chen** received the BS degree from the College of Electrical and Communication Engineering, Xi'an Jiaotong University in 1996 and the PhD degree from the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University in 2002. He is currently a postdoctoral researcher in the Department of Statistics and the Center for Image and Vision Science at University of California, Los Angeles (UCLA). His research interests include

**Song-Chun Zhu** received the BS degree from the University of Science and Technology of China in 1991, and the MS and PhD degrees from Harvard University in 1994 and 1996, respectively. He is currently an associate professor jointly with the Departments of Statistics and Computer Science at the University of California, Los Angeles (UCLA). He is a codirector of the UCLA Center for Image and Vision Science. Before joining UCLA, he worked at Brown University (applied math in 1996-1997), Stanford University (computer science in 1997-1998), and Ohio State University (computer science in 1998-2002). His research is focused on computer vision and learning, statistical modeling, and stochastic computing. He has published more than 70 papers and received a number of honors, including the David Marr prize in 2003, a Sloan fellow in computer science 2001, a US National Science Foundation Career Award 2001, an US Office of Naval Research Young Investigator Award 2001, and the David Marr prize honorary nomination in 1999. He had a few visiting appointments, including a visiting researcher at Microsoft Research Asia in the summers of 1999, 2000, and 2004, and a research professor at the University of California, Berkeley, Math Science Research Institute Spring 2005. In 2004, he founded, with friends, the Lotus Hill Institute for Computer Vision and Information Science in China as a nonprofit research institute ([www.lotushill.org](http://www.lotushill.org)).

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).