Dynamic Hair Manipulation in Images and Videos

Menglei Chai[*] Lvdi Wang[†] Yanlin Weng[*] Xiaogang Jin[*] Kun Zhou[*]
[*]State Key Lab of CAD&CG, Zhejiang University [†]Microsoft Research Asia

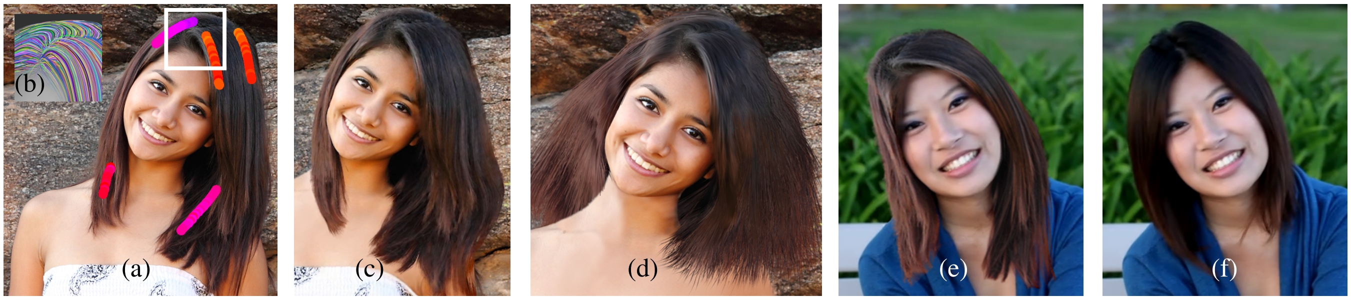


Figure 1: *Given (a) a single portrait image and a few user strokes, we generated (b) a high quality 3D hair model whose visual fidelity and physical plausibility enabled several dynamic hair manipulating applications, such as (c) physically-based simulation, (d) combing, or (e,f) motion-preserving hair replacement in video. Original images courtesy of Asian Impressions Photography.*

Abstract

This paper presents a single-view hair modeling technique for generating visually and physically plausible 3D hair models with modest user interaction. By solving an unambiguous 3D vector field explicitly from the image and adopting an iterative hair generation algorithm, we can create hair models that not only visually match the original input very well but also possess *physical plausibility* (e.g., having strand roots fixed on the scalp and preserving the length and continuity of real strands in the image as much as possible). The latter property enables us to manipulate hair in many new ways that were previously very difficult with a single image, such as dynamic simulation or interactive hair shape editing. We further extend the modeling approach to handle simple video input, and generate *dynamic* 3D hair models. This allows users to manipulate hair in a video or transfer styles from images to videos.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques;

Keywords: hair modeling, image manipulation, video editing

Links: DL PDF WEB VIDEO

1 Introduction

Human hair is so delicate and flexible that people alter it easily, frequently, and in many different ways, unlike any other part of our body. A gentle breeze, a casual brush, or an elaborate hairdo will all reshape the hundreds of thousands of strands into a different arrangement.

Editing hair in a portrait image is, however, far more difficult. Without any knowledge of the underlying 3D structure, traditional image editing tools can hardly perform much complex manipulation on hair. One reason is that in the real world when hair is altered even slightly, some new strands may be exposed while some others become occluded, resulting in a new image with no pixel-level correspondence to the original.

The recent progress on 3D-aware image manipulation [Hoiem et al. 2005; Bitouk et al. 2008; Zhou et al. 2010; Zheng et al. 2012] has demonstrated that, by fitting proper 3D proxies to the objects of interest, a number of semantically meaningful operations that are nearly impossible in the 2D domain now become available. As for hair, Chai et al. [2012] recently proposed a method to generate a 3D hair model from a single portrait image. Such a model visually matches the original image while retaining a strand-based representation, and thus enables some interesting applications such as 3D-aware hair replacement or rendering the portrait from a new viewpoint. However, the model is by no means physically correct: the strands in it are mostly short segments "floating" in a thin shell of the entire hair volume. Such limitation precludes the possibility of more complex and dynamic hair manipulating operations being applied. For example, to correctly simulate the effects of a wind blow or a hair comb, the strand roots must be 1) *fixed on the scalp* and only move with the head rigidly; The strands themselves should also be 2) *smooth* in accordance with hair's physical properties and not have sharp bends; Moreover, they must 3) *preserve the length and continuity* of the real strands in the image as much as possible, so that when locally deformed, they are still continuous strands instead of disconnected fragments. We call a model satisfying these three conditions a *physically plausible* hair model.

A straightforward way to achieve physical plausibility is to "grow" each strand directly from the scalp surface and follow a smoothly changing direction estimated from the image. Unfortunately,

[*]{weng, jin}@cad.zju.edu.cn, kunzhou@acm.org
[†]lvdiwang@microsoft.com

strands generated like this may not correspond well to the real ones in the image due to the limited accuracy of direction estimation and other image artifacts. More specifically, local image orientation estimators alone cannot discriminate hair's direction of growth. Without resolving such directional ambiguity, strands with opposite directions (e.g., those near a hair parting) could be mistakenly connected. To make things worse, such a naïve hair growing process does not even guarantee full coverage of the hair region, i.e., some region containing hair in the image will never be reached by a strand grown from the scalp.

In order to facilitate dynamic hair manipulation in images, we must generate a model that is both *visually* and *physically* plausible.

1.1 Contributions

In this paper we have developed a novel single-view hair modeling algorithm (§ 2). Given a portrait image and a few user annotations as input, it generates a high-quality strand-based 3D hair model that not only visually matches the image, but also possesses important physical plausibility. By utilizing the user's high-level knowledge, we explicitly resolve the directional ambiguity and estimate a dense and consistent 3D hair direction field (§ 2.1), which plays a critical role in hair's *physical plausibility*. We also propose an iterative hair generation procedure (§ 2.2) that uniformly fills the entire hair volume with physically valid strands, thus guaranteeing *visual fidelity*.

The ability to generate visually and physically plausible hair models from single-view images has brought to our attention many new hair manipulation possibilities. For example, we can also reconstruct a *dynamic hair model* from a short single-view video sequence with modest hair motion (§ 3). To achieve this, we start by generating a hair model from a reference frame in the input sequence. This model provides important physical plausibility constraints for estimating a robust motion field from the entire video. Such a field is in turn used to drive the model to produce a complete hair motion sequence that visually matches the original video. Having a consistently changing hair model allows the user to manipulate hair in video, such as changing its appearance and geometric properties, or transferring a static hair model to the video while retaining the original hair motion.

Finally, we demonstrate the quality of the hair models our algorithm generates by applying a few image manipulation effects (§ 4), including interactive combing and physically-based simulation – most of which were previously only available in the 3D hair modeling field.

1.2 Related Work

3D-aware Image Manipulation. Although reconstructing accurate 3D models from a single image is often ill-posed, it has been demonstrated that the use of *3D proxies* can significantly expand the scope of possible image manipulating operations one can perform. Unlike most image-based modeling techniques, such proxies are typically tailored to abstract the common structure of a specific class of objects, but not necessarily fit the true object shapes accurately.

Examples for scenic images include layers of depth images [Oh et al. 2001], plane-based billboards for image pop-ups [Hoiem et al. 2005], cuboid proxies for inserting synthetic objects into the image [Karsch et al. 2011] or editing existing objects in a semantically meaningful way [Zheng et al. 2012].

There are even more effective proxies when it comes to human subjects. Blanz and Vetter [1999] introduced a method to fit a morphable face model to a portrait image. The use of a low-parameter 3D face model can facilitate, e.g., more faithful expression transfer [Yang et al. 2011] and face replacement [Bitouk et al. 2008; Dale et al. 2011]. Similarly, a whole-body morphable model was adopted for parametric human reshaping in images [Zhou et al. 2010] and video [Jain et al. 2010].

This paper is motivated by recent research into single-view hair modeling [Chai et al. 2012]. However, by enforcing physical plausibility through a novel modeling algorithm, we significantly expand the scope of hair manipulating operations in both still images and video.

Image-based Hair Modeling. Despite the existence of many powerful tools, hair modeling from scratch is commonly regarded as a laborious process that involves both in-depth skills and tedious manual work. Yuksel et al. [2009] developed hair meshes, which make hair modeling close to modeling polygonal surfaces. Piuze et al. [2011] introduced generalized helicoids for hair modeling using a hybrid vector field representation. For more references in this area, we recommend the survey by Ward et al. [2007]. On the other hand, most image-based techniques reconstruct a 3D model of real hair from *multiple images* taken either under different lighting conditions [Paris et al. 2004], from different viewpoints [Wei et al. 2005; Paris et al. 2008], at different focal distances [Jakob et al. 2009], or using unconventional imaging devices [Herrera et al. 2012]. Their pipelines and ours share two common steps: estimating some form of an orientation field from the images, and tracing individual hair strands inside a pre-defined volume. However we explicitly resolve the directional ambiguity and develop an iterative hair generation procedure in order to meet our quality requirements.

Bonneel et al. [2009] developed a hair appearance estimation method that can generate a statistically similar 3D hair model from a single photo taken indoors with a flashlamp and a reference gray card. It does not apply to most existing images used in this paper. The generated model is also not expected to match the original image in a pixel-wise sense and thus is unsuitable for editing hair in an image. Luo et al. [2012] noticed that hair orientation is a more robust feature than color during stereo matching and could be used to estimate more accurate hair volume from multiple views. Beeler et al. [2012] proposed a system to capture sparse facial hair and the underlying skin surface simultaneously. The high quality of their results underscores the importance of hair to the realism of virtual avatars.

Compared to the static case, capturing dynamic hair from videos poses an even bigger challenge, and has been an open problem in the field. Yamaguchi et al. [2009] extended existing multi-view approaches [Wei et al. 2005] to handle videos. Zhang et al. [2012] further utilized physically based simulation to refine an initially reconstructed hair sequence. However, without accurate strand correspondence, incorporating input from many viewpoints can overconstrain hair motion, producing excessively smooth results. In contrast, we aim at the application of image/video manipulation and reconstruct a dynamic hair model from a single video sequence.

2 Hair Modeling Algorithm

Given a portrait image and a few user strokes, our algorithm first estimates a dense and unambiguous 3D hair direction field within the hair volume (§ 2.1), then generates 3D hair strands that are compatible with both the direction field and the original image (§ 2.2) through an iterative process, yielding a hair model with both *visual fidelity* and *physical plausibility*.

2.1 Hair Direction Field from Single Images

Interactive Hair Annotation. While existing analysis tools are usually able to give fairly robust estimation of the *nondirectional* local orientation of the hair in a portrait image, discriminating the correct growth direction of individual strands can be very difficult for an automatic algorithm. However, a user can often trivially resolve such directional ambiguity, thanks to the high-level comprehension of human brains.

To utilize this unique capability of users, we develop an interface that allows a user to intuitively draw a few strokes on the image that roughly indicate the growing direction of the hair underneath (Fig. 2(a)). These strokes have two purposes. First, they are used to segment out the hair from other parts of the image using Paint Selection [Liu et al. 2009]. Second, they serve as important clues for resolving the directional ambiguity in a later step.

Nondirectional Hair Tracing. Given the segmented hair image region, we estimate a sparse 2D orientation map using the iterative refinement method described in [Chai et al. 2012]. The basic idea is to calculate an initial orientation map using a bank of oriented Gabor filters, as well as a confidence map. The orientation and confidence maps are then refined reciprocally in an iterative process to eliminate erroneous estimates caused by image artifacts.

From the refined orientation map, we generate a sparse set of 2D curves that correspond to the prominent hair strands in the original image. We follow the 2D hair tracing algorithm described by Chai et al. [2012], but use the arc sampling method [Tan et al. 2008; Beeler et al. 2012] to decide the direction of growth at each curve vertex. This generally produces smoother and more reliable curves than the original re-centering correction approach.

Note that unlike [Chai et al. 2012], the non-directional 2D curves generated here will not become part of the strands in our final hair model. Instead, they serve as important topological and continuity constraints for resolving the hair's directional ambiguity.

Resolving Directional Ambiguity. Determining the per-pixel hair direction from the image directly can be both unreliable due to the imperfection of the imaging process and ambiguous due to the lack of high-level image comprehension. Therefore, we solve for only the directions along the 2D curves generated above where strands are clearly distinguished and orientations robustly estimated. Based on the observation of real hair, we further enforce several constraints: 1) the direction along any smooth curve should be consistent; 2) neighboring curve segments with similar orientations should also have consistent directions; 3) wherever available, the curve direction should be consistent with user strokes.

We represent each 2D curve γ_i as an ordered list of equally distanced vertices, where $\gamma_{i(j)}$ denotes the j-th vertex of γ_i. We can define an intrinsic direction $\tilde{\mathbf{t}}_{i(j)}$ at $\gamma_{i(j)}$ as the normalized vector pointing from $\gamma_{i(j-1)}$ to $\gamma_{i(j+1)}$. The vertex order of a curve is determined arbitrarily during the tracing process, but according to the first constraint above, the direction of hair growth along γ_i can be expressed by $\hat{\mathbf{t}}_{i(j)} = s_i \tilde{\mathbf{t}}_{i(j)}$, where $s_i \in \{-1, 1\}$. Similarly, each stroke drawn by the user can be represented in the same way. But since it inherently has the correct vertex order, we replace the flag s_i with a global weight w_m that controls the influence of annotations. We used $w_\mathrm{m} = 10$ for all the results in this paper.

To encompass the other two constraints, we define a score P for curve γ_i as:

$$P(\gamma_i) = \sum_j \sum_{\gamma_{p(q)} \in \mathcal{N}_{i(j)}} \hat{\mathbf{t}}_{i(j)} \cdot \hat{\mathbf{t}}_{p(q)}. \qquad (1)$$

Here $\mathcal{N}_{i(j)}$ contains vertices from other curves *with resolved directions* (including user strokes) that are within a given distance (70 pixels for the results in this paper) from $\gamma_{i(j)}$. Together, the three constraints translate into the following binary integer programming formulation:

$$\arg\max_{\{s_i\}} \sum_i P(\gamma_i). \qquad (2)$$

Solving Equation (2) is known to be NP-complete [Karp 1972]. But we have found a greedy algorithm works well in our case. The algorithm divides the curves into three disjoint sets: a *resolved set*, a max-heap of *tentative curves*, and the remaining *ambiguous curves*. In the beginning, the resolved set contains only the user strokes, and the tentative heap is empty. The algorithm then simply repeats the following three steps until all curves are resolved: 1) select each ambiguous curve γ_i whose vertices are neighboring to some resolved curves and move γ_i to the tentative heap with a priority equal to $|P(\gamma_i)|$; 2) pop the curve γ_i with the highest priority from the tentative heap, let s_i be the one in $\{-1, 1\}$ that makes $P(\gamma_i)$ nonnegative, and add γ_i to the resolved set; 3) update the priorities of the tentative curves that are neighboring γ_i.

Figure 2(b) and (c) show the 2D curves before and after resolving the directional ambiguity respectively. Note that our algorithm is fast enough to produce timely updates after the user draws a stroke during the interactive annotation step.

3D Direction Field. Assuming a weak perspective camera projection model, we can fit a 3D morphable head model [Blanz and Vetter 1999] for the subject in the image. We then adopt the method proposed by Chai et al. [2012] to estimate the depths of the 2D curves generated previously and further construct an initial hair model. This initial hair model and the head model are rasterized together from both the front and back views to generate two depth maps respectively, which are used to create a closed volume that is guaranteed to match the 2D hair region in the image and enclose the head model. We define the hair volume $H \in \mathbb{R}^3$ as this closed volume *minus* the volume occupied by the 3D head.

Within the hair volume H, we calculate a discrete 3D vector field, denoted as D, by solving a linear least-squares optimization problem in a uniform grid [Fu et al. 2007]. Let $\mathbf{v}_i$ be the 3D directional vector at grid cell i, we minimize the following energy:

$$E_\mathrm{d}(D) = \sum_{i \in H} \|\Delta \mathbf{v}_i\|^2 + w_\mathrm{c} \sum_{i \in C} \|\mathbf{v}_i - \mathbf{c}_i\|^2 , \qquad (3)$$

where Δ is the discrete vector Laplacian operator, C is the set of grid cells where a directional constraint $\mathbf{c}_i$ exists, and w_c adjusts the weight of the constraints. We let $w_\mathrm{c} = 10$ and the resolution of D to be 80^3 throughout the paper.

To define the directional constraints, we project the 2D directed curves generated previously onto the curved boundary surface of H, and store the tangent vectors of the resulting 3D curves in the corresponding grid cells. For the cells near the scalp, we also assign the surface normal as the constraints. Combining these two types of constraints with the Laplacian term, Equation (3) ensures that D matches both the natural direction of hair growth near the scalp and the direction analyzed from the visible hair in the original image, while in the meantime being as smooth as possible in a least-squares sense.

2.2 3D Hair Strands Generation

To generate individual 3D strands for the final hair model, we propose a novel two-step approach. In the first step, a number of initial

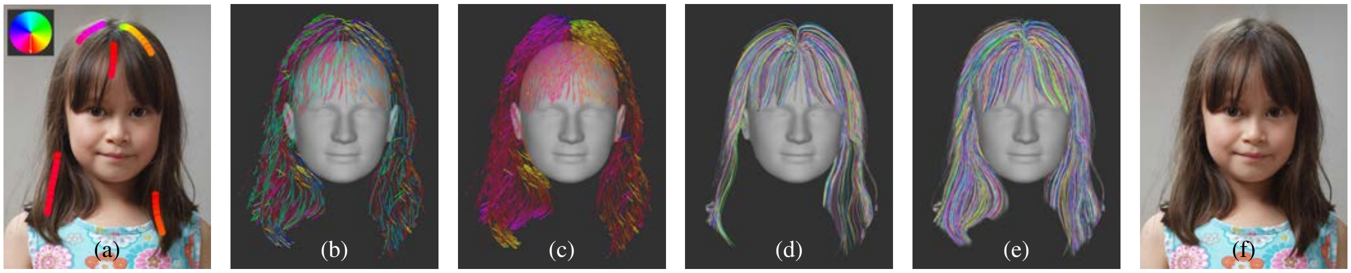


Figure 2: *Hair modeling pipeline. (a) The user draws strokes to roughly indicate the direction of hair growth and segment out the hair region. (b) Initial curves with incorrect direction. (c) Curves with resolved direction. (d) Initial 3D strands. (e) Strands after one iterative refinement. (f) Final rendering result. The direction along each 2D curve in (a–c) is represented by the corresponding color in the hue ring (inset of (a)). Original image courtesy of Wolfgang Lonien.*

hair strands are generated directly from the scalp surface, following the 3D direction field D. Strands generated like this are not guaranteed to fill the entire hair volume H, resulting in discrepancies between the original image and the rendering of the hair model. So we introduce a second step that iteratively synthesizes additional strands and refines the overall hair model.

Initial Hair Tracing. To begin with, we define a default *scalp region* over the 3D morphable head model. We use the mask of the hair region to exclude the facial region from the default scalp region. n_s uniformly distributed locations are first chosen in the parameterized 2D scalp space [Wang et al. 2009] and then mapped back to the 3D scalp surface as hair roots. Starting from these roots, individual strands grow out according to D using a second-order Runge-Kutta method. The step length is set to half of the grid width of D, where directional vectors at non-integer locations are trilinearly interpolated. A strand stops growing if it exceeds a maximum curve length or goes outside of H.

Iterative Refinement. Given the initial set of strands $\mathcal{S}$, we iteratively add new strands to $\mathcal{S}$, aiming to fill the empty space in H with strands that are as visually and physically plausible as the initial ones. Our basic idea is to alternatively repeat these two steps: 1) generate a *candidate strand set* $\mathcal{S}^*$ by tracing strands from seeds randomly sampled within the empty region of H; 2) for each candidate strand $\xi^* \in \mathcal{S}^*$, we check whether there is an appropriate strand $\xi \in \mathcal{S}$ that is partially close enough to ξ^*. If there is, we "cut-and-connect" ξ and ξ^* to form a new *joint strand* ξ' and add it to $\mathcal{S}$. This process continues until no appropriate candidate can be connected to any existing strand.

Let $|\xi|$ denote the number of vertices in strand ξ, and $\xi_{(i)}$ the i-th vertex (from root to tip) of ξ. The cut-and-connect process finds two spatially neighboring vertices $\xi_{(i)}$ and $\xi^*_{(j)}$ from an existing strand ξ and a candidate ξ^* respectively, and generates a new strand ξ' so that

$$\xi'_{(k)} = \begin{cases} \xi_{(k)} & \text{if } k < i, \\ \left(\xi_{(i)} + \xi^*_{(j)}\right)/2 & \text{if } k = i, \\ \xi^*_{(k-i+j)} & \text{if } i < k < i + |\xi^*| - j. \end{cases} \tag{4}$$

Note that the portion of ξ' closer to the root always comes from an existing strand, i.e., $\xi'_{(1)}$ is guaranteed to be on the scalp.

In order to select the appropriate strand $\xi \in \mathcal{S}$ and decide the pair of vertices to cut-and-connect, we need a metric to evaluate the resulting joint strand. Physically and aesthetically, ξ' should be smooth and not have sharp turns at the joint vertex $\xi'_{(i)}$. It should also fill the entire hair volume uniformly so that the final hair model, when rendered from the original viewpoint, closely matches the input image. We therefore define the following energy term for each joint strand ξ':

$$E_{\mathrm{c}}(\xi') = \sum_k e^{-\frac{(k-i)^2}{2\sigma_{\mathrm{c}}^2}} \Delta\xi'_{(k)} + w_\rho \sum_k \rho\left(\xi'_{(k)}\right), \tag{5}$$

where the Laplacian Δ at vertex $\xi'_{(k)}$ weighted by its geodesic distance to vertex $\xi'_{(i)}$ reflects the smoothness near the joint. $\rho\left(\xi'_{(k)}\right)$ measures the *hair density* in a small neighborhood of $\xi'_{(k)}$. For all examples shown in this paper, $\sigma_{\mathrm{c}} = 1$ and the weight of the density term $w_\rho = 0.01$.

For each candidate strand, we enumerate all (if any) of its neighboring vertices in $\mathcal{S}$ within a predefined radius and select only the one that results in the joint strand with the lowest energy. After the optimal ξ' is constructed, we further apply a 1D Gaussian filter on each of the strand vertices within two vertices from the joint to improve the geometric smoothness. The variance of the Gaussian kernel is set to 2 (vertices).

The refinement process usually terminates within 4 to 5 iterations in our experiments, and generates 40% to 70% of the total strands in the hair model (Fig. 2(d–f)).

3 Hair Modeling from Video

Being able to generate a physically plausible hair model from an image further allows us to extend our application to image sequences. In this section, we describe a method to generate a dynamic hair model from a video clip with modest hair motion.

To ensure temporal coherence, we choose to generate a 3D hair model from a user-specified reference frame using our image-based modeling algorithm, and deform that model based on the motion estimated from the video so that it matches the hair in each frame.

Let $\mathcal{S}_r$ be the *base hair model* generated from the reference frame r. We assume that the hair model $\mathcal{S}_i$ at any frame i is just a deformed version of $\mathcal{S}_r$, i.e., $\mathcal{S}_i = \mathbf{T}_i(\mathcal{S}_r)$. Without loss of generality, we also assume $i > r$.

We define $\mathbf{T}_i$ as a combination of a global 3D rigid transformation $\mathbf{M}_i$ and a per-vertex 2D warping $\mathbf{W}_i$ that accounts for the non-rigid deformation. We estimate $\mathbf{M}_i$ by tracking the head motion in the video using a multilinear model [Vlasic et al. 2005; Dale et al. 2011].

The estimation of $\mathbf{W}_i$ is trickier due to the complexity of hair itself. We start by calculating a dense optical flow $\mathbf{V}_i$ from frame $i-1$ to i (the 2D location $\mathbf{p}$ in frame $i-1$ corresponds to $\mathbf{p} + \mathbf{V}_i(\mathbf{p})$

in frame i). Instead of using pixel colors, we match the two sparse orientation fields [Chai et al. 2012], which prove to be a more robust feature for highly specular fiber assemblies like hair [Luo et al. 2012]. Directly tracing the motion trajectories at each pixel from frame r to i yields a 2D warping field between the two frames. Although such a field captures small-scale motion relatively well, the estimation error will propagate and accumulate over time, making it less and less accurate as i increases.

To remedy such an issue, we calculate another 2D warping field $\mathbf{W}_i^\dagger$ from the reference frame r to frame i by first tracking a small number of highly reliable feature points either detected automatically using the Kanade-Lucas-Tomasi approach, or marked manually by the user in the reference frame, and then diffuse the resulting sparse flow to the entire frame in an edge-aware fashion [Lang et al. 2012]. In some difficult cases when motion blur, occlusion, or other image artifacts prevent the feature tracking from finding the correct correspondence, the user could interactively modify any feature point in a frame. Such manual corrections will be propagated smoothly over the entire sequence.

We now define a 2D warping field $\mathbf{W}_i^*$ for frame i by considering both the dense optical flow and the sparse feature correspondence. The displacement vector at 2D location $\mathbf{p}$ from frame r to frame i can be recursively defined as

$$\begin{aligned} \mathbf{W}_i^*(\mathbf{p}) &= 0.5\mathbf{W}_i^\dagger(\mathbf{p}) + 0.5\big[\mathbf{W}_{i-1}^*(\mathbf{p}) + \mathbf{V}_i(\mathbf{p} + \mathbf{W}_{i-1}^*(\mathbf{p}))\big], \\ \mathbf{W}_r^*(\mathbf{p}) &= \mathbf{W}_r^\dagger(\mathbf{p}) = \mathbf{V}_r(\mathbf{p}) = \mathbf{0}, \quad \text{where } i > r. \end{aligned} \tag{6}$$

Let $\mathbf{v}_r$ and $\mathbf{p}_r$ denote the 3D location of a strand vertex of the base hair model and its projection on the X-Y plane respectively. To calculate the deformed vertex $\mathbf{v}_i$ in frame i, we first apply the rigid transformation $\mathbf{M}_i$ to $\mathbf{v}_r$, giving $\mathbf{v}_i' = \mathbf{M}_i\mathbf{v}_r$. This effectively aligns the base hair model with the head in the current frame.

We then calculate the per-vertex 2D displacement for $\mathbf{v}_i'$ as:

$$\mathbf{W}_i(\mathbf{v}_i') = \mathbf{W}_i^*(\mathbf{p}_r) - (\mathbf{p}_i' - \mathbf{p}_r), \tag{7}$$

where $\mathbf{p}_i'$ is the 2D projection of $\mathbf{v}_i'$. Notice that since $\mathbf{W}_i^*$ encodes both non-rigid and rigid deformations, we need to explicitly subtract from it the per-vertex displacement caused by rigid transformation.

Finally, we have $\mathbf{v}_i = \mathbf{v}_i' + \beta\mathbf{W}_i(\mathbf{v}_i')$, where β is a smoothstep function that is 0 for a root vertex and gradually increases to 1 as $\mathbf{v}_i$ becomes farther from the scalp, guaranteeing that hair roots are always fixed on the scalp during the entire video.

4 Results and Discussion

We tested our method extensively on a wide variety of portrait images and video clips downloaded from the Internet. The performance was measured on a PC with an Intel Xeon E5620 processor and an NVIDIA GeForce GTX 680 graphics card. We treat each hair strand as a polyline whose per-vertex color and transparency are sampled directly from the segmented hair layer of the original image by projecting the 3D position of each vertex back to the image plane. Enabling alpha-to-coverage allows us to render tens of thousands of anti-aliased strands in real time without the need for depth sorting.

Note that although the default distance between succeeding strand vertices does not guarantee every pixel in the hair region being sampled, the final rendering result can still faithfully reproduce the original image. This is because the color variation *along* a strand is usually quite smooth.

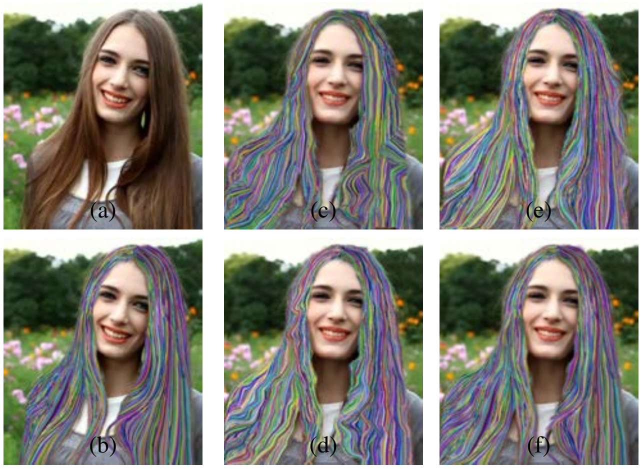


Figure 3: *Hair motion estimation. (a) Reference frame. (b) Base hair model. (c–f) Results of the 60th frame after the reference generated using (c) strands without physical plausibility, (d) feature correspondence only, (e) optical flow only, and (f) our method. Original video courtesy of Children of Heaven Photography.*

In addition to the hair strands, we also reconstruct a 3D head model with an automatically inpainted texture as in [Chai et al. 2012] to deal with the newly exposed facial regions after the hair or the viewpoint have been modified. In some cases (e.g., Figure 1(a), 5 and 6), we further segment out the body layer and the background layer, and fill those occluded regions using a PatchMatch-based interactive inpainting tool [Barnes et al. 2009].

Static Hair Modeling. Figure 1 and 2 show a few 3D hair models generated from still images using our method. The entire hair-modeling pipeline generally takes tens of seconds for a middle-sized image, including user interaction time. Usually 2 to 8 hair annotation strokes are sufficient to produce satisfying results. Depending on the input, the resulting hair models contain 20k to 40k strands. Although there are several parameters throughout the process, they are relatively robust to different inputs, and we use the same set of empirically chosen values for all results in this paper.

We have observed that explicitly resolving the directional ambiguity not only ensures the physical plausibility of hair strands, but also enhances the *visual fidelity* notably, as shown in the comparison between our method and the prior art using non-directional orientation fields (Fig. 4).

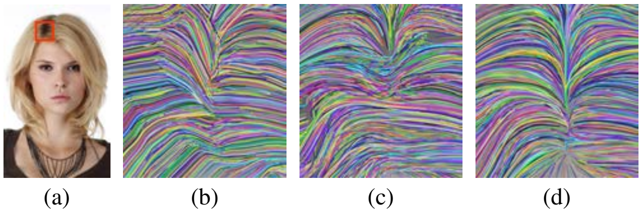


Figure 4: *3D strands of the highlighted region in (a) generated from (b) an orientation map [Chai et al. 2012], (c) a tensor field [Paris et al. 2008], and (d) our unambiguous direction field. Original image courtesy of Lynn Geng.*

Our iterative cut-and-connect algorithm for 3D strand generation has proven significantly superior to simply tracing additional strands "backward" from the empty region, since many such

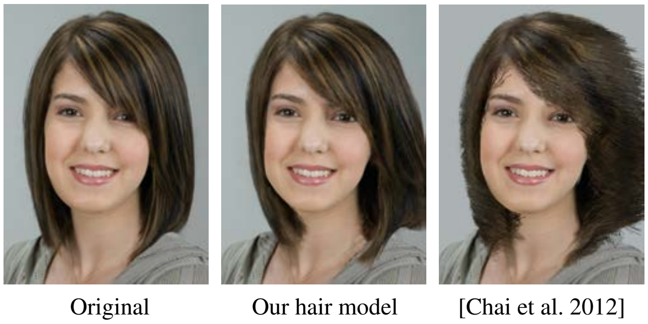


Figure 5: *Physically based simulation of hair in an image. Original image courtesy of Denise Mahoney.*

strands, especially those far away from the scalp in long hairstyles, will never reach the scalp before terminating at the outer boundary of the hair volume. Our method, on the other hand, ensures completeness and uniformity by incrementally looking for physically plausible candidates that can extend the set of valid hair strands.

Dynamic Simulation and Interactive Editing. The hair models generated by our algorithm can undergo much more dramatic modifications than does the prior art because of the physical plausibility. This allows us to borrow some tools directly from the 3D hair-modeling arsenal for image manipulation purposes.

For example, we adopt a simple mass spring model [Selle et al. 2008] to simulate the dynamic behavior of hair under wind and head motion (Fig. 5). The reconstructed 3D head geometry is converted into a signed distance function for detecting hair-head collisions. The velocity field of the wind is represented by a 3D procedural noise [Perlin 2002]. Although our simplified implementation does not currently consider gravity nor more complex strand interactions such as collision or Coulomb friction, the subtle hair movements already notably enhance the perceptual realism of the resulting virtual avatars. Using more advanced models [Daviet et al. 2011] would be interesting for future research.

By precomputing a multi-scale hierarchy of the hair model [Wang et al. 2009], we are able to provide an intuitive stroke-based user interface that allows the user to perform detail-preserving editing of the hair shape at any cluster level, such as combing and cutting (see Fig. 6 and the accompanying video). For more implementation details, please refer to the appendix.

Having a physically plausible hair model is key for both the dynamic simulation and the editing tools to function as one intuitively expects. For comparison, we have performed the same simulation and editing operations to hair models generated by Chai et al.'s previous method [2012]. As shown in the right columns of Fig. 5 and 6, the fragmented and directionally ambiguous hair strands therein tend to cause severe artifacts in the results.

Manipulating Hair in Video. Figure 3 and 7(b) show two dynamic hair models generated from video clips. To generate these models, the user has also manually adjusted a small number of feature points (12 and 6 for the two results respectively). Since such corrections made in one frame can be propagated quickly and smoothly to the remaining frames, this process requires only a few minutes for a normal user without special background.

Based on the generated model, the user can edit the appearance of hair, or replace the original hair model $\mathcal{S}$ with a new one $\mathcal{S}'$ generated from another image. In the former case, the user is provided a simple slider-based UI to transform the hair color in HSV space, or adjust the amount of synthetic highlights rendered using Marschner et al.'s shading model [2003] as in [Chai et al. 2012].

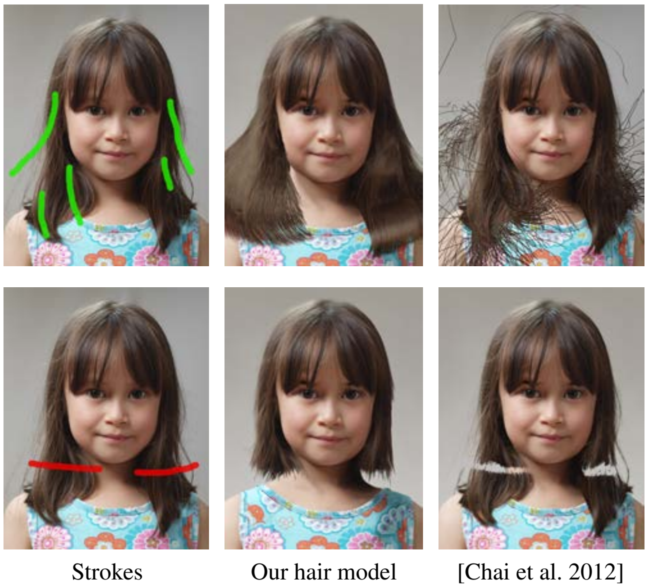


Figure 6: *Interactive editing. Hair in the original image (left column) is combed (upper middle) or cut shorter (lower middle) using an intuitive stroke-based interface. The right column shows the results of applying the same strokes to the hair model generated by [Chai et al. 2012].*

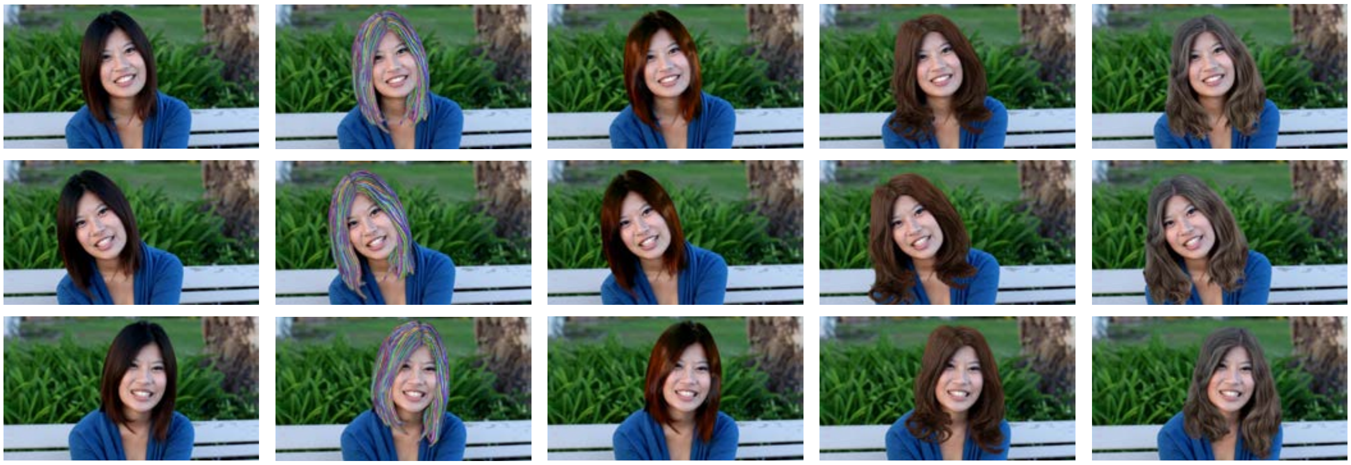
To replace the hair model in the video, we first establish a strand correspondence between $\mathcal{S}$ and $\mathcal{S}'$ by finding, for each ξ' in $\mathcal{S}'$, the geometrically most similar ξ in $\mathcal{S}$. Here we calculate the geometric dissimilarity between two strands by the $\mathcal{L}^2$ distance between their respective feature vectors as defined in [Wang et al. 2009]. The per-vertex relative motion of ξ is then combined with the global rigid transformation to deform the corresponding ξ' for each frame.

Limitations. Our hair modeling algorithm shares similar drawbacks to the previous approach by Chai et al. [2012]: without additional depth knowledge, we cannot accurately estimate the 3D shape of the hair volume, especially for some highly curled or unusual hairstyles. For some complex or messy hairstyles, even the user may find it difficult to determine the correct direction of hair growth. Given only a frontal view, it is also impossible to guess how hair in the back will look. So we let the hair strands beyond the silhouette transit smoothly into a dummy model to minimize the visual discontinuity along the silhouette when the model is viewed from a new angle.

Our current dynamic hair-modeling algorithm can only handle simple video input where the motions of both head and hair are modest, due to the limitation of the motion field estimate. When replacing the hair in video, only the large-scale motion of the original hair can be retained. The detailed inter-strand or strand-body interactions are currently ignored.

5 Conclusion

We have presented an efficient approach for generating high-quality 3D hair models from portrait images. The generated hair models not only visually match the input very well but also possess physical plausibility, meaning that strands have their roots fixed on the scalp,



(a) Original (b) 3D strands (c) Appearance editing (d) Hair replacement 1 (e) Hair replacement 2

Figure 7: *Hair manipulation in video. Each column shows three different frames. Please see the accompanying video for the full sequence. Original video courtesy of Asian Impressions Photography.*

are smooth in accordance with the hair's physical properties, and preserve the length and continuity of the real strands in the image as much as possible. These properties enable us to perform a few hair manipulation operations such as dynamic simulation or hair shape editing in portrait images, which are difficult to achieve with previous portrait editing techniques.

Our single-view hair-modeling algorithm can be extended to create dynamic hair models from video clips with modest hair motion. The physical plausibility of our hair models allows the user to manipulate hair in video, such as changing its geometric properties or transferring a static hair model to the video while retaining the original hair motion. We would like to note that the problem of creating dynamic hair models from videos with large hair motion is far from being solved. We hope our work can stimulate much research interest along this direction.

Acknowledgment

We would like to thank the original image and video owners for letting us use their work, Xiao Li for help on UI design, Xin Tong for inspirational discussions, and the SIGGRAPH reviewers for their constructive comments. This work is partially supported by NSFC (No. 61003145 and No. 61272305) and the 973 program of China (No. 2009CB320801).

References

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. 28*, 3, 24:1–11.

BEELER, T., BICKEL, B., NORIS, G., BEARDSLEY, P., MARSCHNER, S., SUMNER, R. W., AND GROSS, M. 2012. Coupled 3D reconstruction of sparse facial hair and skin. *ACM Trans. Graph. 31*, 4, 117:1–10.

BITOUK, D., KUMAR, N., DHILLON, S., BELHUMEUR, P. N., AND NAYAR, S. K. 2008. Face Swapping: Automatically replacing faces in photographs. *ACM Trans. Graph. 27*, 39:1–8.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH '99*, 187–194.

BONNEEL, N., PARIS, S., PANNE, M. V. D., DURAND, F., AND DRETTAKIS, G. 2009. Single photo estimation of hair appearance. *Computer Graphics Forum 28*, 1171–1180.

CHAI, M., WANG, L., YU, Y., WENG, Y., GUO, B., AND ZHOU, K. 2012. Single-view hair modeling for portrait manipulation. *ACM Trans. Graph. 31*, 4, 116:1–8.

DALE, K., SUNKAVALLI, K., JOHNSON, M. K., VLASIC, D., MATUSIK, W., AND PFISTER, H. 2011. Video face replacement. *ACM Trans. Graph. 30*, 6, 130:1–10.

DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Trans. Graph. 30*, 6, 139:1–12.

FU, H., WEI, Y., TAI, C.-L., AND QUAN, L. 2007. Sketching hairstyles. In *Proc. EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, 31–36.

HERRERA, T. L., ZINKE, A., AND WEBER, A. 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph. 31*, 6, 146:1–9.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph. 24*, 3, 577–584.

JAIN, A., THORMÄHLEN, T., SEIDEL, H.-P., AND THEOBALT, C. 2010. MovieReshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. 29*, 6, 148:1–10.

JAKOB, W., MOON, J. T., AND MARSCHNER, S. 2009. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph. 28*, 5, 164:1–9.

KARP, R. M. 1972. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85–103.

KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic oobject into legacy photographs. *ACM Trans. Graph. 30*, 6, 157:1–12.

LANG, M., WANG, O., AYDIN, T., SMOLIC, A., AND GROSS, M. 2012. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph. 31*, 4, 34:1–8.

LIU, J., SUN, J., AND SHUM, H.-Y. 2009. Paint selection. *ACM Trans. Graph. 28*, 3, 69:1–7.

LUO, L., LI, H., PARIS, S., WEISE, T., PAULY, M., AND RUSINKIEWICZ, S. 2012. Multi-view hair capture using orientation fields. In *Proc. CVPR 2012*, 1490–1497.

MARSCHNER, S., JENSEN, H. W., CAMMARANO, M., WORLEY, S., AND HANRAHAN, P. 2003. Light scattering from human hair fibers. *ACM Trans. Graph. 22*, 3, 780–791.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proc. SIGGRAPH '01*, 433–442.

PARIS, S., BRICEÑO, H., AND SILLION, F. 2004. Capture of hair geometry from multiple images. *ACM Trans. Graph. 23*, 3, 712–719.

PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph. 27*, 3, 30:1–9.

PERLIN, K. 2002. Improving noise. In *Proc. SIGGRAPH '02*, 681–682.

PIUZE, E., KRY, P. G., AND SIDDIQI, K. 2011. Generalized helicoids for modeling hair geometry. *Computer Graphics Forum 30*, 2, 247–256.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Trans. Graph. 27*, 3, 64:1–11.

TAN, P., FANG, T., XIAO, J., ZHAO, P., AND QUAN, L. 2008. Single image tree modeling. *ACM Trans. Graph. 27*, 5, 108:1–7.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIC, J. 2005. Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3, 426–433.

WANG, L., YU, Y., ZHOU, K., AND GUO, B. 2009. Example-based hair geometry synthesis. *ACM Trans. Graph. 28*, 3, 56:1–9.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. C. 2007. A survey on hair modeling: styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 2, 213–234.

WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Trans. Graph. 24*, 3, 816–820.

YAMAGUCHI, T., WILBURN, B., AND OFEK, E. 2009. Video-based modeling of dynamic hair. In *the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, 585–596.

YANG, F., WANG, J., SHECHTMAN, E., BOURDEV, L., AND METAXAS, D. 2011. Expression flow for 3D-aware face component transfer. *ACM Trans. Graph. 30*, 4, 60:1–10.

YUKSEL, C., SCHAEFER, S., AND KEYSER, J. 2009. Hair meshes. *ACM Trans. Graph. 28*, 5.

ZHANG, Q., TONG, J., WANG, H., PAN, Z., AND YANG, R. 2012. Simulation guided hair dynamics modeling from video. *Computer Graphics Forum 31*, 7, 2003–2010.

ZHENG, Y., CHEN, X., CHENG, M.-M., ZHOU, K., HU, S.-M., AND MITRA, N. J. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph. 31*, 4, 99:1–11.

ZHOU, S., FU, H., LIU, L., COHEN-OR, D., AND HAN, X. 2010. Parametric reshaping of human bodies in images. *ACM Trans. Graph. 29*, 4, 126:1–10.

A Implementation of the Hair Editing Tools

A.1 Combing tool

Each *combing stroke* drawn by the user is represented internally as a 2D polyline γ with a per-vertex intrinsic direction $\tilde{\mathbf{t}}$ defined in the same way as for the user-annotated directional strokes in § 2.1. The direction at an arbitrary location *on* γ is calculated by linearly interpolating the directions defined at the two nearest vertices. Given a point p on the image plane, and its closest point p' on γ, we further define a direction vector at p as the direction at p' multiplied by a smooth falloff function which takes the value of 1 when $\|p - p'\|^2 = 0$, and 0 when $\|p - p'\|^2 \geq R$. In other words, a user-drawn stroke defines a local vector field within a certain distance R from it.

For a given hair strand, we examine its vertices in the order from root to tip: if the 2D projection p of a strand vertex falls within the influence region of the combing stroke, we rotate the remaining portion of this strand toward the direction vector defined at p around an axis passing through p and perpendicular to the image plane. As a result, the deformation is length-preserving and does not modify the depth of any strand vertex.

The stroke radius R is user defined. For the results shown in this paper we let $R = 32$ and use a simple hat-shaped falloff function.

A.2 Cutting tool

Intuitively, a *cutting stroke* simply breaks a hair strand at the stroke-strand intresection and removes the part of the strand farther from the hair root. In real world, however, hair stylists frequently use thinning scissors to create fuzzier yet more natural-looking cuts. To simulate such effects, we also define a radius R for a cutting stroke and examine a strand's vertices from root to tip. For any strand vertex $\xi_{(i)}$ that is within the range of the stroke, we set the *probability* of this strand being cut at $\xi_{(i)}$ to be in proportion to $1 - r/R$, where r is the shortest distance between the stroke and the 2D projection of $\xi_{(i)}$, and randomly select a strand vertex to cut.

When $R = 0$ the cutting stroke produces a clean cut. We set $R = 32$ for the result in Figure 6.