To appear in ACM TOG 31(4).

Single-View Hair Modeling for Portrait Manipulation

Menglei Chai[*] Lvdi Wang[†] Yanlin Weng[*] Yizhou Yu[‡] Baining Guo[†] Kun Zhou[*]

[*]State Key Lab of CAD&CG, Zhejiang University [†]Microsoft Research Asia [‡]The University of Hong Kong

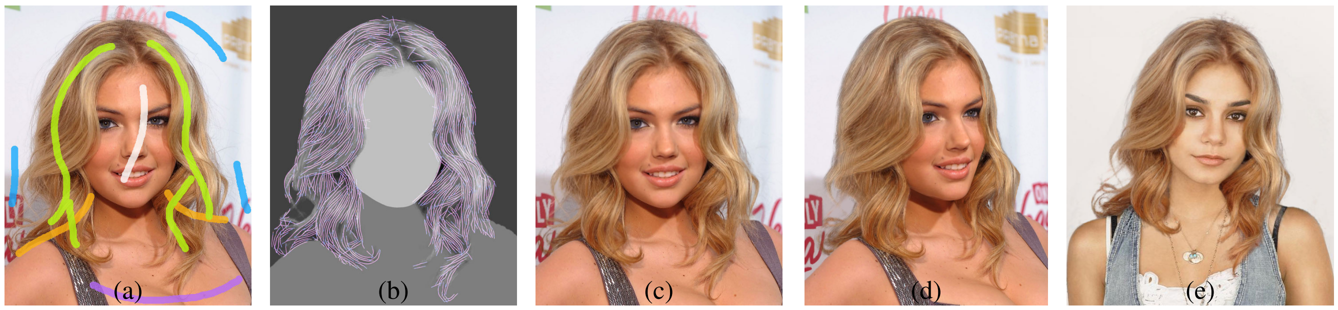

Figure 1: *Given a portrait image and a few strokes drawn by the user as input (a), our method generates a strand-based 3D hair model as shown in (b), where a fraction of reconstructed fibers are highlighted. The hair model can be used to convert the input portrait into a pop-up model (c) which can be rendered in a novel view (d). It also enables several interesting applications such as transferring the hairstyle of one subject to another (e). Original images courtesy of Getty Images (a) and Andrew MacPherson (e).*

Abstract

Human hair is known to be very difficult to model or reconstruct. In this paper, we focus on applications related to portrait manipulation and take an application-driven approach to hair modeling. To enable an average user to achieve interesting portrait manipulation results, we develop a single-view hair modeling technique with modest user interaction to meet the unique requirements set by portrait manipulation. Our method relies on heuristics to generate a plausible high-resolution strand-based 3D hair model. This is made possible by an effective high-precision 2D strand tracing algorithm, which explicitly models uncertainty and local layering during tracing. The depth of the traced strands is solved through an optimization, which simultaneously considers depth constraints, layering constraints as well as regularization terms. Our single-view hair modeling enables a number of interesting applications that were previously challenging, including transferring the hairstyle of one subject to another in a potentially different pose, rendering the original portrait in a novel view and image-space hair editing.

Keywords: strand tracing, portrait pop-ups, hairstyle replacement

1 Introduction

Images featuring *people* as their subjects are always of great interest. The urge to synthesize better looking, or simply funnier portraits has motivated much successful research in computer graphics and computer vision in the past decades. For example, studies on data-driven face geometry and facial element detection can already robustly recover a 3D face model from a single image [Blanz and Vetter 1999] or generate an animated face avatar from video [Pighin and Lewis 2006]. Yet notably, *hair*, which plays a unique role in depicting a person's character, has been constantly omitted or replaced by ad-hoc geometry. However, handling hair in a more careful way would be beneficial in multiple aspects. It would significantly enhance the quality and realism of a synthetically rendered face. It would also enable a number of applications that involve hair, such as transferring a person's hairstyle in one photograph to another person in a second photograph and generating better looking hair for virtual avatars in games and interactive programs.

Unlike most other parts of the human body which can be well approximated locally by a surface, human hair is composed of hundreds of thousands of thin fibers grouped together in widely different and possibly very complicated ways to form various hairstyles, and is known to be very difficult to model or reconstruct [Ward et al. 2007]. It usually requires a number of images taken from a wide range of viewing angles as well as a significant amount of image processing and visual computing to reconstruct a complete 3D hair model. Such a level of complexity prevents widespread adoption of the underlying technology.

In this paper, we focus on applications related to portrait manipulation and take an application-driven approach to hair modeling. Portrait manipulation has its unique requirements and restrictions. The most important requirement is that a re-rendered portrait must be photorealistic. That is, its quality should be comparable to a real photograph. An implication is that re-rendered hair should still maintain its resolution in the original portrait especially when hair strands are visible there. However, a portrait is typically taken from a near-frontal view and it is typically unnecessary for the view angle of any re-rendering to deviate much from such a view. This means hair visible from a near-frontal view should be modeled more carefully than the rest.

To enable an average user to achieve interesting portrait manipulation results, we develop a single-view hair modeling technique with modest user interaction to meet the aforementioned requirements. Without sufficient information to reconstruct a completely accurate 3D model, our goal is a plausible high-resolution strand-based 3D hair model adequate for portrait manipulation. Given a single portrait photo, we first fit a 3D morphable head model [Blanz and Vetter 1999] to the person in the photo and let the user annotate the hair region with a few strokes (§ 3.1). Within this region our method traces a sparse set of 2D hair strands (§ 3.2) using a novel

[*]weng@cad.zju.edu.cn, kunzhou@acm.org
[†]lvdiwang@microsoft.com, bainguo@microsoft.com
[‡]yizhouy@acm.org

high-precision strand tracing algorithm, which explicitly models uncertainty and local layering at hair intersections. The depth of the traced strands is solved through an optimization, which simultaneously considers depth constraints, layering constraints as well as regularization terms. To further make the reconstructed strands denser, we synthesize extra hair strands that are invisible in the original image due to occlusion (§ 3.3).

Our single-view hair modeling technique enables a number of interesting portrait manipulation applications that were previously challenging. With modest user input, we show how to convert a single image into a "2.5D" portrait pop-up which can be rendered from novel viewpoints (§ 4.1). The hair model extracted from one image can be used to replace the hair in another image, even when the two people have different head poses (§ 4.2), facilitating interesting usage such as virtual hairstyle try-ons. We can also edit the geometry or appearance of the hair model for image-space hairstyling (§ 4.3).

2 Related Work

Portrait manipulation is of great interest to many researchers. While methods working in the 2D domain have achieved compelling results in face beautification [Leyvand et al. 2008], face swapping [Bitouk et al. 2008] and face attribute enhancement [Joshi et al. 2010], there is a trend recently in utilizing *3D shapes* for more complex portrait manipulations, such as reshaping of human bodies [Zhou et al. 2010; Jain et al. 2010], face replacement [Shlizerman et al. 2010; Dale et al. 2011], face component transfer [Yang et al. 2011] and generating face animations from large image collections [Shlizerman et al. 2011]. Image manipulations performed with respect to these specialized 3D proxy meshes can utilize the semantic structures contained in the input and better resolve problems caused by ambiguities and occlusions in a single image. Our approach also follows this trend and constructs a strand-based 3D hair model for portrait manipulation.

Hair modeling is an extensively studied problem in computer graphics (see [Ward et al. 2007] for a comprehensive survey). Our work is most related to image-based hair capture methods [Paris et al. 2004; Wei et al. 2005; Paris et al. 2008; Jakob et al. 2009]. Given multiple hair images taken from different viewpoints or under different illuminations, these methods can reconstruct a full 3D hair model which can be rendered in a CG scene. Our input, however, is a single image without any knowledge about the geometry and illumination of the scene. Instead of trying to recover a full hair model, we generate a plausible high-resolution strand-based 3D hair model from the single input image, which suffices for many portrait manipulation applications. Bonneel et al. [2009] proposed a method to construct a parametric hair appearance model from a single photograph, which is used to render a synthetic hair model that visually matches the input image in a statistical sense. The method assumes the photo was taken indoors with a flash on the camera together with a reference gray card. It is thus not applicable to most images used in this paper which are downloaded from the internet. The method also does not recover hair geometry.

Single-view 3D modeling is in general an ill-posed problem. There exists some work focusing on modeling specific objects and scenes. Hoiem et al. [2005] generate photo pop-ups for outdoor images containing ground, walls or sky. Rivers et al. [2010] proposed a 2.5D model for cartoon images which allows a small number of given views of a 2D cartoon to be smoothly interpolated as if it is a 3D object. Öztireli et al. [2011] proposed a method that extracts a 2.5D model of a bilateral symmetric object from a simple user sketch.

3 Single-View Hair Modeling

Without sufficient information to reconstruct a completely accurate 3D hair model, the goal of our single-view hair modeling is a plausible hair model adequate for portrait manipulation. Thus, the estimated hair model should possess the following properties: *First*, we should be able to reconstruct the hair region in the original image from the hair model. The visual difference (if any) should be as small as possible. *Second*, when the hair model is rendered from a novel viewpoint, it should still look like the original hairstyle. *Third*, due to occlusions, most of the actual hair strands are invisible from the single input image (e.g., those behind the head and beneath the outermost layer). These strands should be hallucinated in a reasonable way so that there are no annoying holes when the hair is rendered from a novel viewpoint.

3.1 Preprocessing

User-assisted segmentation: To simplify the task, we assume the input portrait consists of four basic layers: hair, face, body and background, which are processed independently for future steps. Due to their complex appearances, Lazy Snapping [Li et al. 2004] is adopted to perform image segmentation and extract the visible regions of these layers with some user interaction. Typically, a few strokes are sufficient to obtain all these layers.

Hair matting: Since hair is translucent and the hair region in an image often has a complicated boundary, the segmentation result obtained from the previous step is approximately correct but not sufficiently accurate. We further apply matting to improve the accuracy and obtain an alpha channel. We simply expand and shrink the segmented hair region to compute a trimap automatically. Based on this trimap, closed-form matting [Levin et al. 2008] is applied to obtain the final hair region mask and foreground color values.

Head mesh fitting: We assume a weak perspective camera projection model and fit a 3D head model to the input image using Yang et al.'s method [2011]. Following the morphable head model [Blanz and Vetter 1999], we collected a head model database and computed around 100 principal components for the database using Principal Component Analysis. The basic idea of the fitting procedure is to compute a new head shape S_{new} by calculating a series of coefficients β for the principal components:

$$S_{\text{new}} = \overline{S} + V \cdot \beta, \qquad (1)$$

where $\overline{S}$ is the average head shape vector and V is the matrix of principal components. During fitting, the Active Shape Model [Milborrow and Nicolls 2008] is used to localize about 76 facial feature points on the original image, including features along the face boundary and important facial landmarks such as eye corners, mouth boundary and the nose tip. Then the coefficient vector β is solved by minimizing an energy function.

3.2 Generating 2D strands

Many existing image-based hair modeling methods (e.g., [Paris et al. 2004; Wei et al. 2005; Paris et al. 2008]) generate individual hair strands by first computing a dense 3D orientation field and then tracing strands within the field. Although this can guarantee the smoothness of traced hair strands, the underlying assumption that any tiny volume inside the hair region can be characterized by a single orientation is not necessarily true because in a real hairstyle strand segments with different orientations are frequently in contact with each other. This problem is much worse for a 2D hair image where it is unavoidable for projected hair strands to overlap.

Typically a per-pixel orientation is estimated by applying a series of oriented filters and finding the orientation that gives the maximum response. Due to the imperfection of the original image, the resulting orientation map is usually noisy or over-smoothed. We instead estimate a sparse but accurate orientation map that allows us to robustly handle hair intersections and occlusions.

3.2.1 Orientation estimation

Similar to previous methods, we filter the input image I with a bank of oriented filters $\{K_\theta\}$, where a filter kernel K_θ is designed to detect an orientation at angle θ. Let $F(x, y, \theta) = (K_\theta * I)(x, y)$ be the response of K_θ at pixel (x, y), an estimated local orientation $\tilde{\theta}(x, y)$ at each pixel is then given by $\tilde{\theta}(x, y) = \arg\max_\theta(F(x, y, \theta))$.

Our filter bank is composed of 32 even-symmetric Gabor kernels [Jain and Farrokhnia 1991] with their orientations evenly spaced between 0° and 180°:

$$K_\theta(u, v) = \exp\left(-\frac{1}{2}\left[\frac{\tilde{u}^2}{\sigma_u^2} + \frac{\tilde{v}^2}{\sigma_v^2}\right]\right)\cos\left(\frac{2\pi\tilde{u}}{\lambda}\right), \qquad (2)$$

where $\tilde{u} = u\cos\theta + v\sin\theta$ and $\tilde{v} = -u\sin\theta + v\cos\theta$. Such a cosine Gabor kernel proves to be a reliable orientation estimator at image-space *ridges* [Jakob et al. 2009].

In addition to the orientation $\tilde{\theta}$, we also calculate a *confidence* $w(x, y)$ at each pixel, which encodes the likelihood of pixel (x, y) belonging to a hair strand in the image:

$$w(x, y) = \sum_\theta \left(d(\theta, \tilde{\theta}) \cdot (F(\theta) - F(\tilde{\theta}))^2\right)^{0.5}, \qquad (3)$$

where $d(\theta_1, \theta_2)$ measures the minimum angle between two orientations as in [Paris et al. 2004]. By thresholding the confidence, we discard unreliable orientation estimations and obtain a sparse but robust orientation map for prominent hair features in the input image (Figure 2).

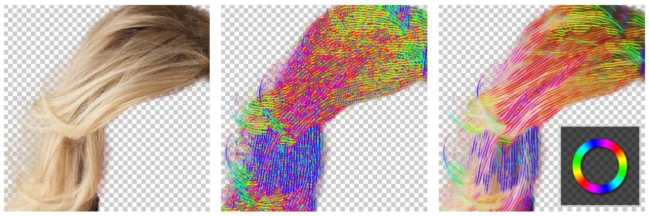

Figure 2: *Left: A cropped hair region. Middle: Orientation estimated at every pixel within the hair region. Right: High-confidence-only orientation overlaid on the hair region. The orientation angle is represented by color as shown in the inset.*

Ideally, the parameters of the Gabor kernel should be proportional to the size of visible strand features in the image, but for the results shown in the paper we simply use $\sigma_u = 1.8$, $\sigma_v = 2.4$, and $\lambda = 4$.

Iterative refinement: Due to the imperfections in input images, such as the presence of camera noise and image compression artifacts, sometimes an unreliably estimated orientation may have a relatively high confidence value. It is hard to distinguish such false positives from true strand features by simple thresholding.

We introduce an iterative orientation refinement process based on an intuitive observation: a high-confidence pixel that belongs to a hair strand is more likely to have a high-confidence neighborhood along the estimated orientation than a pixel with false-positive high confidence. Therefore, after an initial orientation map $\tilde{\theta}$ is estimated, we use its corresponding confidence map w as input to the next iteration and estimate a new orientation map $\tilde{\theta}'$ and the corresponding confidence map w' from w, using the same oriented-filtering method described above. As shown in Figure 3, this simple process can effectively filter out those high-confidence estimations caused by image artifacts, resulting in a cleaner and more reliable orientation map from which we can now extract prominent hair strands. In practice, 1 to 2 iterations suffice for the inputs we have tested.

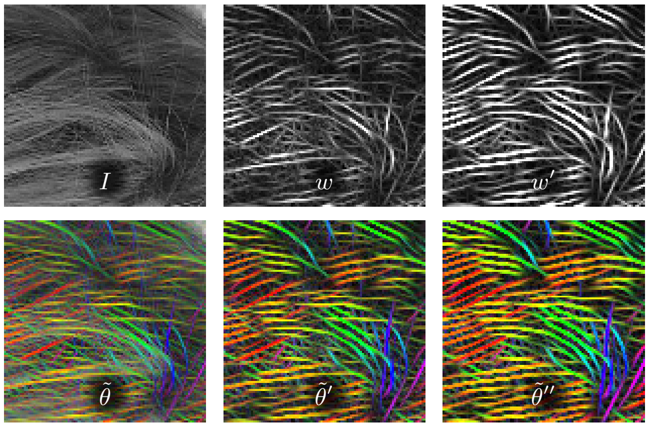

Figure 3: *Iterative orientation refinement. Given the input image intensity I, we calculate an orientation map $\tilde{\theta}$ and a confidence map w. The confidence map w is then used as input to calculate a refined orientation map $\tilde{\theta}'$ and a corresponding w'.*

3.2.2 Strand tracing

Next we convert the sparse orientation map into a set of geometric curves that correspond to individual hair strands. Since our input is a single image under uncontrolled illumination, recovering accurate 3D curves (e.g., from shading [Paris et al. 2004]) is impractical. Instead, at this stage we trace all curves on the 2D image plane and wherever two hair strands intersect in the image, we also try to resolve the correct topology and store the *local layering* information to the corresponding curve vertices.

We first perform non-maximum suppression similar to [Jakob et al. 2009], but on the confidence map instead of the original image. A pixel p is considered to be a *seed pixel* if and only if $w(p) > w_{\text{high}}$ and

$$\frac{w(p) - \max\{w(p_{\text{L}}), w(p_{\text{R}})\}}{w(p)} > \epsilon, \qquad (4)$$

where $w(p_{\text{L}})$ and $w(p_{\text{R}})$ are bilinearly sampled along the line passing p and normal to the estimated local orientation $\tilde{\theta}(p)$. They are on opposite sides of p. The results in this paper are generated with $w_{\text{high}} = 0.3, w_{\text{low}} = 0.05, \epsilon = 0.2$.

Given a seed pixel p, we extend it in both opposite directions along the estimated orientation $\tilde{\theta}(p)$ simultaneously. Similar to existing methods [Paris et al. 2004; Wei et al. 2005; Paris et al. 2008], one step of tracing proceeds by selecting one of the two possible directions along the orientation $\tilde{\theta}(p)$ at the current location, $\vec{v}(p)$, that minimizes the bending angle, and taking a step forward along that direction: $p_{i+1} = p_i + \delta\vec{v}(p_i)$, where p_i is the location at the i-th tracing step.

Inspired by the use of hysteresis thresholding in edge detection [Canny 1983], we maintain a *certainty status* and a *health point* of the current hair tracing thread: tracing one strand in one direction starts with a positive health point (5 in our experiments) and ends when the health point drops to zero. When tracing status is “certain”, the tracing direction is determined by the local orientation map and the health point refills to the initial value after every step; When tracing status is “uncertain”, the tracing direction is estimated from the previous traced vertices so that the curvature is approximately maintained, and the health point decreases by 1 after every step.

At each step i, we update the tracing status according to the following rules:

1. Set the status to “certain” if $i = 0$.
2. Change the status to “uncertain” if $w(p_i) < w_{\text{low}}$.
3. Change the status to “uncertain” if $w(p_i) \geq w_{\text{low}}$ and $\arccos(\vec{v}(p_i) \cdot \vec{v}(p_{i-1})) > \theta_{\text{max}}$.
4. Change the status to “certain” if $w(p_i) \geq w_{\text{low}}$ and $\arccos(\vec{v}(p_i) \cdot \vec{v}(p_{i-1})) \leq \theta_{\text{max}}$.
5. Keep the status unchanged otherwise.

When rule #3 is applicable, we assume the currently traced vertex and all the succeeding “uncertain” vertices are occluded by another strand, thus mark them with a special flag indicating their being *occluded*.

During the tracing process, we also remove the pixels along the curve from the list of seed pixels so that each hair strand will be traced only once. We set the step size δ to 75% of the pixel width and $\theta_{\text{max}} = \pi/6$ for all the results in this paper. Finally, the “uncertain” vertices at both ends of a strand are trimmed after tracing.

Re-centering correction: Because of error accumulation, a traced curve can easily drift away from the true center of the curve in the image. We perform a re-centering correction for each traced vertex on the curve: For a vertex with 2D location p, we sample the confidence at p as well as its two nearby locations p_{L} and p_{R} that lie on a line normal to the current tracing direction (Figure 4 left). A tent function $\wedge(t)$ is fit with $\wedge(0) = w(p)$, $\wedge(-1) = w(p_{\text{L}})$, and $\wedge(1) = w(p_{\text{R}})$, and we translate p by $\arg\max \wedge(t)$ along the line normal to its tracing direction (Figure 4 right).

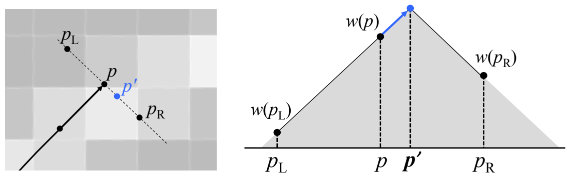

Figure 4: *Re-centering for traced strand vertices.*

Compared with the previous method of [Paris et al. 2004] that traces the streamlines of a filtered dense orientation map, our tracing algorithm usually generates a cleaner and more robust result (Figure 5).

Strand color: In addition to geometry, we also sample the color and alpha value for each strand vertex from the segmented hair region in the input image.

3.3 Generating 3D strands

Given the head model (§ 3.1) and the sparse 2D hair strands (§ 3.2) as input, we generate a 3D hair model using a two-step approach: first, the depth of each strand vertex is estimated through an optimization that takes into account depth constraints, local layering constraints and regularization terms. We then use the resulting sparse 3D strands to define a bounding volume around the head, and generate more strands inside it to fill the part of the hair volume that is occluded in the original image.

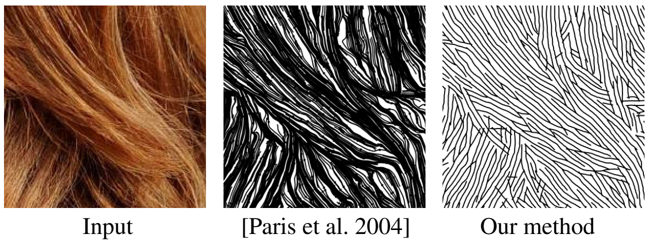

Figure 5: *2D curves traced from a filtered dense orientation map [Paris et al. 2004] (middle) and from a sparse orientation map using our method (right).*

3.3.1 Depth estimation

We use the following simple and effective heuristic rules: hair strands near the silhouette have a depth of 0, which is also the depth of the center of the head; hair strands over the forehead have a depth slightly smaller than that of the head mesh; all other hair strands have a depth somewhere in between and should observe the local occlusion relationships estimated earlier.

We incorporate the following types of constraints in our strand depth optimization:

1. *Depth constraints* to enforce known depths on strands near the forehead or silhouette. Such constraints can also be manually specified via a stroke-based tool.
2. *Strand constraints* to maintain smoothness and prevent sharp angles along strands.
3. *Neighbor constraints* to maintain depth coherence among nearby strands with similar orientations.

Let p_i be the position of the i-th vertex of a strand, $z(p)$ be the depth (i.e., z coordinate) of p, we formally define the following energy terms according to the above constraints as:

$$\begin{aligned} E_Z(p_i) &= (z(p_i) - \bar{z}(p_i))^2, \\ E_S(p_i) &= \left(z(p_i) - \frac{z(p_{i-1}) + z(p_{i+1})}{2} \right)^2, \\ E_N(p_i) &= \left(z(p_i) - \frac{\sum_{q \in \mathcal{N}(p_i)} z(q)}{|\mathcal{N}(p_i)|} \right)^2, \end{aligned}$$

where $\bar{z}(p)$ is the depth value constraint at p, $\mathcal{N}(p)$ contains the neighbor vertices of p that are not on the same strand of p but have a similar orientation and the same occlusion flag (§ 3.2.2) as p.

Finally, we solve for the depth of all strand vertices by minimizing the overall energy:

$$E = w_Z E_Z + w_S E_S + w_N E_N \tag{5}$$

using the biconjugate gradient method. For the results in this paper we empirically set the weights to $w_Z = 5$, $w_S = 2$ and $w_N = 1$.

3.3.2 Synthesizing additional strands

Real hair occupies a volumetric region around the head. The hair strands we have generated so far only include those visible from the input image. They roughly cover the outmost layer of the frontal half of the entire hair volume. Due to the complex structures and widely different styles of real hair, any assumption about what the invisible part of the hair looks like may not always hold. Therefore, we aim to define a hair volume that 1) does not add unnecessary hair strands when rendered from the original viewpoint of the input image; and 2) transitions smoothly from the frontal (visible) half to the rear (invisible) half.

Hair volume definition: To start with, we define three depth maps D_{front}, D_{mid}, and D_{back}. The first depth map is generated by rasterizing the computed sparse 3D strands. The depths at pixels covered by the sparse strands are diffused to the entire hair region using a procedure similar to that in [Paris et al. 2008].

To calculate the second depth map, we first obtain an extended hair region that is the *union* of the original hair region and the projection of the head model on the image plane. The depths near the outer silhouette of the original hair region are set as the boundary condition. They are diffused to the entire extended hair region.

The third depth map is calculated by fixing the depth values at the region boundary, and pushing the internal depth values smoothly backward so that the depth map reaches outside the head and covers the rear half of the head.

The three layers partition the entire 3D region around the head into two closed halves, i.e., a 3D point $p(x, y, z)$ is inside the volume if $D_{\text{front}}(x, y) < z \leq D_{\text{mid}}(x, y)$ or $D_{\text{mid}}(x, y) < z < D_{\text{back}}(x, y)$. We again rely on diffusion to propagate the 3D orientations as well as color and alpha values from the three layers to the internal voxels. 3D orientations are first converted to structure tensors before diffusion as in [Paris et al. 2008]. The color of non-hair regions in D_{mid} and D_{back} is synthesized from surrounding hair regions automatically using PatchMatch [Barnes et al. 2009]. For our specific purpose, we do not need to fill all the internal voxels, but only those sufficient to form a thick and seamless outer layer. This saves a significant amount of time and storage.

3D strand tracing in the volume: Unlike previous work on hair capture, we do not force 3D hair strands to grow from the scalp as this may result in unnecessary discrepancies between the input and any synthetic image rendered from the original view. Instead, we trace strands from random locations sampled inside the hair volume. During tracing, we also keep a record of *strand density* within each voxel. Tracing terminates at a voxel whose strand density exceeds a given threshold.

4 Portrait Manipulation

In this section we show several applications of our hair model in portrait manipulation.

4.1 Portrait pop-ups

Recovering a 3D head avatar from a single image is of great interest in interactive applications. However, due to the complexity of hair geometry, most existing work focuses only on the facial region and optionally add some ad-hoc hair to the fitted head geometry. By using our strand-based hair model, we are able to generate more vivid head avatars which we call *portrait pop-ups*.

A portrait pop-up is composed of a background layer, a body layer, a head model, and our hair model. Generally, the body and background layers will be occluded within the overlapped regions, and when manipulating the portrait such as rotating the view or changing the hair, some overlapped regions may become visible. To overcome this problem, we fill the occluded holes in these two layers with the PatchMatch image completion method [Barnes et al. 2009]. For the body layer, the user may need to draw the body boundary on the hole region to define the body shape and to constrain the completion target region. Furthermore, for layers with complex structural information, the user may also draw additional guide strokes to get better results (Figure 6). For the 3D head model, we construct a 2D parameterization and use it to compute a texture map. For each triangle of the model, if its projection on the image is bounded by the head region, the texture colors inside the triangle can be obtained from the image and stored in the texture map. Then the image completion method is used to generate colors for texels that do not obtain colors from the image.

We also allow the user to add depth variations to the body layer to match the real geometry of the subject's body. In this case, the user draws some sparse strokes with predefined depth value on the body layer, the depth at any other locations is calculated by simply solving a Laplace equation with Dirichlet boundary conditions.

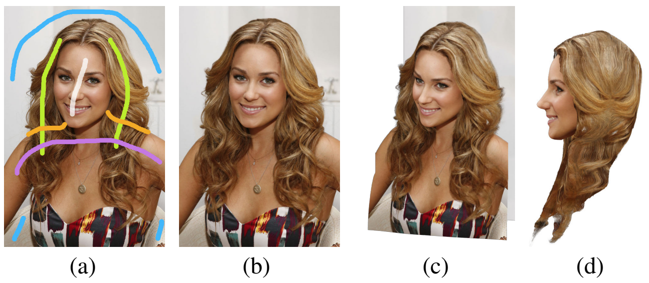

Figure 6: *(a) The input image overlaid with the strokes the user has drawn for layer segmentation and completion. (b) The generated portrait pop-up rendered in the original view. (c)(d) The portrait rendered in two novel views. Original image courtesy of Mark von Holden.*

4.2 Hairstyle replacement

Replacing an individual's hair in a photograph allows one to virtually try on different hairstyles. There exist many commercial software products designed exactly for this purpose (e.g., HairMaster, DailyMakeover.com, taaz.com, etc.). As far as we know none of these systems utilizes a strand-based 3D hair model as we do.

Given a source portrait I_{src} and a target portrait I_{dst}, our goal is to replace the hair of the subject in I_{dst} by the hair in I_{src}. We start by extracting the 3D hair model and head model from I_{src} (§ 3) and creating a pop-up model from I_{dst} (§ 4.1). Making use of the one-to-one vertex correspondence between the source and target head models, we can calculate a transformation M_t from the head model in I_{src} to the head model in I_{dst} to compensate for the changes in the head's shape and position. Currently M_t consists of two parts, one is a translation and rotation matrix computed during head fitting and the other is a scaling matrix that aligns the bounding boxes of the two head models. M_t is then applied to the hair model generated from I_{src} to transform it to the correct position in I_{dst}.

To increase realism, we can also calculate a *mutual* ambient occlusion term for the transferred hair geometry and the target subject. "Mutual" here means that, when calculating the AO term on the hair model we only consider the target subject (excluding the hair model itself) as the occluder and vice versa. Self occlusion is ignored to avoid undesired darkening on the subject. Such mutual AO effects often provide useful visual clues about the 3D relationship between the replaced hair and the subject (Figure 7).

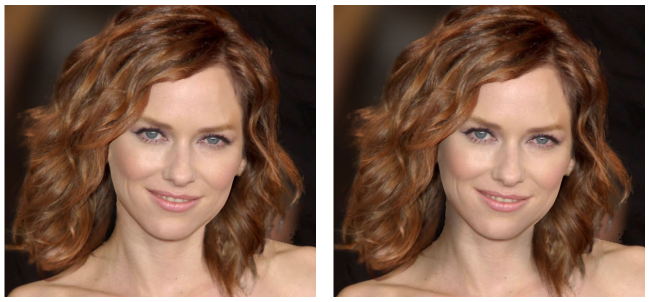

Figure 7: *Hair replacement without (left) and with (right) ambient occlusion. Original image courtesy of Retna Ltd.*

It should be noted that *hairstyle* replacement has dramatically different effects from *face* replacement. While replacing the facial features tends to *de-identify* the subject and results in a "familiar-looking stranger" [Bitouk et al. 2008], replacing the hairstyle will usually maintain the subject's identity well to a human observer. We believe both techniques have their own unique applications.

4.3 Hairstyle editing

By utilizing the strand-based model generated from the image, we have also implemented tools that allow the user to manipulate certain aspects of the hair in real time, such as adjusting the hair's color, smoothness, shininess, or adding geometric noises [Yu 2001; Bonneel et al. 2009; Xu et al. 2011] (Figure 8). When editing specular highlights, we initialize a single point light located at the camera (assuming the photograph was shot with a camera-mounted flashlight) and allow the user to modify its properties in real time.

In our current prototype system, the user can adjust the global hair color or add per-strand random color variations, both in HLS color space. The smoothness is controlled by simply filtering the color along each strand with an adjustable-sized kernel. We further adopt the real-time implementation of Marschner et al.'s model [2003] for adding specular highlights to the hair [Nguyen and Donnelly 2004].

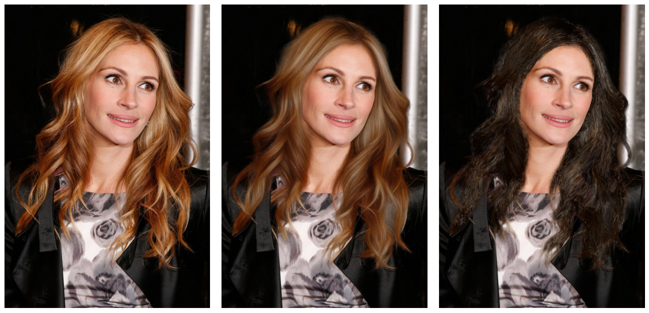

Figure 8: *Hair editing. Left: original image. Middle: new hair color with increased smoothness. Right: new hair color with increased shininess and added geometric noise. Original image courtesy of Getty Images.*

5 Results and Discussion

We have tested our method on different input images including digital photographs, paintings, and pencil drawings as shown in Figure 10. The hair models generated in this paper contain up to 50000 strands each. A single strand is stored as a polyline (Figure 1(b)) with vertex colors sampled from the original image. During rendering, these strands are expanded into screen-aligned quadstrips in real time by a geometry shader on an NVidia GTX460 graphics card. We enabled *alpha-to-coverage* with 8× MSAA which striked a good balance between visual quality and performance.

Comparisons: We have also compared our 3D strand-based hair replacement with two alternative approaches. *The first one* simply maps the segmented hair region to a flat plane that can be scaled and rotated (as in some existing hair try-on systems like HairMaster, HairStyled.com, etc.). *The second one* first calulates a depth map for the hair region by applying the algorithm described in § 3.3.1 to pixels, then use this depth to displace a planar mesh. A flat plane cannot convey the depth variation of the hair and fails to handle occlusions when the viewpoint is changed even slightly. While a displaced planar mesh can reflect the global depth variation, it cannot represent the local depth order of 3D strands. Moreover, the hair may look distorted and blurred when rendered in a new view. In comparison, our hair model consists of a volumetric arrangement of 3D strands, and thus better handles these problems (Figure 9).

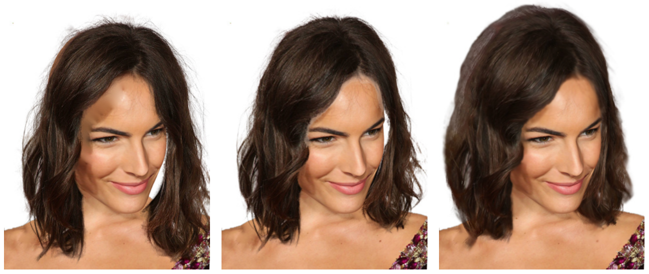

Figure 9: *Hair replacement using 2D warping (left), depth-based warping (middle), and our strand-based model (right). Original image courtesy of Getty Images.*

Amount of interaction: We have invited several users to try our prototype system. It usually takes a user less than three minutes to interactively mark up different semantic regions (i.e., hair, body, background etc.) in the image. For some cases (such as in Figure 1) when the automatic depth estimation cannot yield a satisfactory result, the user may need to specify a few more strokes for depth constraints. The system then takes about 30 seconds to generate the 3D hair model and the portrait pop-up for an image with a hair region of approximately 500 × 500 pixels, measured on an Intel Xeon E5620 CPU with 8GB memory.

Limitations and future work: Our method has several limitations. The hair model we generate is not a full 3D model of the entire hair volume. The hair that is originally invisible in the input image is hallucinated and simply incorrect. When rendered from a new view which deviates too much from the original (e.g., greater than 45°), the result may look less realistic (see e.g., Figure 6(d)). Hair strands in some portraits do not have distinguishable features. In such cases our method may not be able to recover reliable sparse strands. Nevertheless, our method will still generate a hair model that looks the same as the input in the original view, but with less geometric details when rendered from a novel view. It would be hard to directly extend our method to the reconstruction of a hair animation from a video sequence, partly due to the fact that the strands are not traced from the scalp. During hair replacement, the source and the target images may have dramatically different illuminations and thus the transferred hair may look unrealistic. Finding a way to robustly estimate the light sources in the input images and perform hair relighting presents many challenges and is worthy of further investigation.

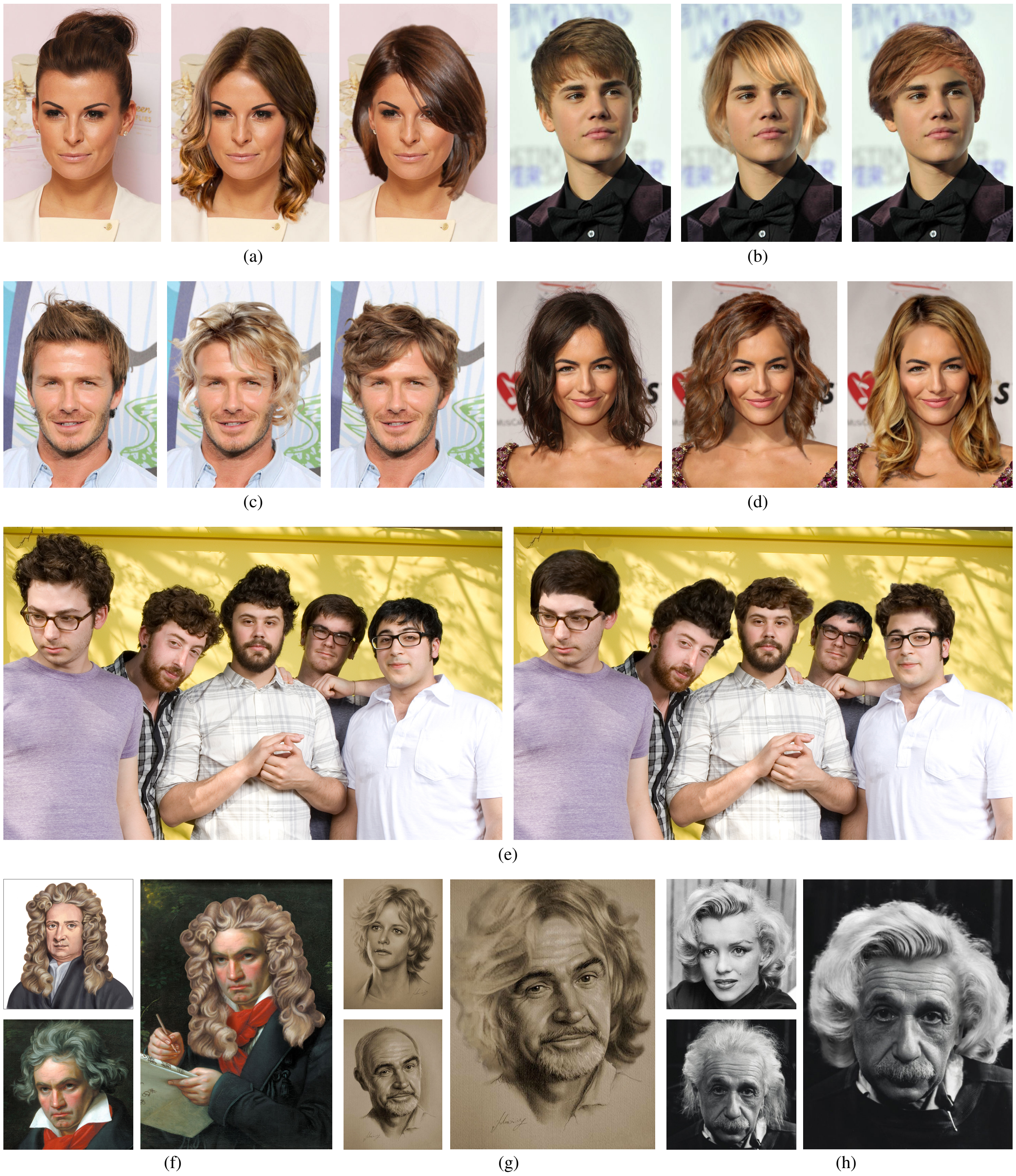

Figure 10: *(a)-(d): Hairstyle replacement results. In each group the original image is on the left. In (e), the hairstyles of the five subjects are swapped. The hair transfer results in (f), (g), and (h) show how our method can be applied to paintings, drawings, as well as historical photographs. Original images courtesy of Getty Images (a–d), Chrissy Piper (e), and Krzysztof Lukasiewicz (g).*

6 Conclusion

We have presented a single-view hair modeling technique with modest user interaction. Our goal is a plausible high-resolution 3D hair model adequate for portrait manipulation. This is made possible by an effective high-precision 2D strand tracing algorithm, which explicitly models uncertainty and local layering, and a hair depth optimization algorithm, which simultaneously considers depth constraints, layering constraints as well as regularization terms. Our single-view hair modeling enables a number of interesting applications that were previously challenging.

Acknowledgments

This work is partially supported by NSFC (No. 61003145 and No. 60825201) and the 973 program of China (No. 2009CB320801). We would like to thank Xin Tong for helpful discussions, the SIGGRAPH reviewers for their constructive comments, and the image owners for letting us use their work.

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, 24:1–24:11.

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*, 39:1–39:8.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of 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.

CANNY, J. 1983. *Finding Edges and Lines in Images*. Master's thesis, MIT. http://hdl.handle.net/1721.1/6939.

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

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

JAIN, A. K., AND FARROKHNIA, F. 1991. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition 24*, 12, 1167–1186.

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

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

JOSHI, N., MATUSIK, W., ADELSON, E. H., AND KRIEGMAN, D. J. 2010. Personal photo enhancement using example images. *ACM Trans. Graph. 29*, 3, 12:1–12:15.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2008. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 2, 228–242.

LEYVAND, T., COHEN-OR, D., DROR, G., AND LISCHINSKI, D. 2008. Data-driven enhancement of facial attractiveness. *ACM Trans. Graph. 27*, 3, 38:1–38:9.

LI, Y., SUN, J., TANG, C., AND SHUM, H. 2004. Lazy snapping. *ACM Trans. Graph. 23*, 3, 303–308.

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.

MILBORROW, S., AND NICOLLS, F. 2008. Locating facial features with an extended active shape model. In *Proceedings of ECCV '08*, Springer, 504–513.

NGUYEN, H., AND DONNELLY, W. 2004. Hair animation and rendering in the Nalu demo. In *GPU Gems 2*. Addison-Wesley Professional.

ÖZTIRELI, A. C., UYUMAZ, U., POPA, T., SHEFFER, A., AND GROSS, M. 2011. 3D modeling with a symmetric sketch. In *Proceedings of SBIM*, 23–30.

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, 30:1–30:9.

PIGHIN, F., AND LEWIS, J. P. 2006. Performance-driven facial animation. In *ACM SIGGRAPH 2006 Courses*.

RIVERS, A., IGARASHI, T., AND DURAND, F. 2010. 2.5D cartoon models. *ACM Trans. Graph. 29*, 4, 59:1–59:7.

SHLIZERMAN, I. K., SHECHTMAN, E., GARG, R., AND SEITZ, S. M. 2010. Being John Malkovich. In *Proceedings of ECCV '10*, 341–353.

SHLIZERMAN, I. K., SHECHTMAN, E., GARG, R., AND SEITZ, S. M. 2011. Exploring photobios. *ACM Trans. Graph. 30*, 4, 61:1–61:9.

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.

XU, K., MA, L.-Q., REN, B., WANG, R., AND HU, S.-M. 2011. Interactive hair rendering and appearance editing under environment lighting. *ACM Trans. Graph. 30*, 6, 173:1–173:10.

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, 60:1–60:10.

YU, Y. 2001. Modeling realistic virtual hairstyles. In *Proceedings of Pacific Graphics '01*, 295–304.

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, 126:1–126:10.