eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

AutoHair: Fully Automatic Hair Modeling from A Single Image

Menglei Chai   Tianjia Shao   Hongzhi Wu   Yanlin Weng   Kun Zhou

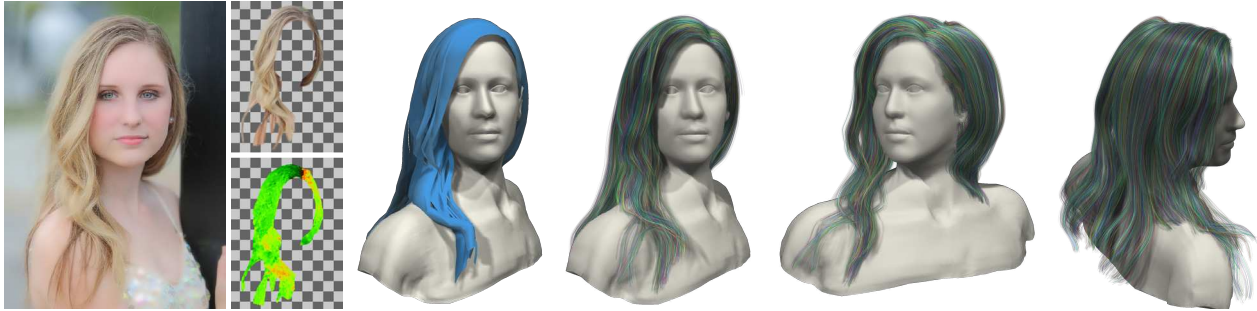State Key Lab of CAD&CG, Zhejiang University*

Figure 1: *Fully automatic hair modeling from a single image. Given a single portrait image as input, our method computes a hair segmentation and a hair growth direction map, which are used to obtain a matching hair shape from a large set of 3D model exemplars. A complete and high-quality 3D hair model is then generated based on the matching shape and the direction map. The whole process is fully automatic, with no user interaction or parameter tuning. Original image courtesy of Bob HARRIS.*

Abstract

We introduce *AutoHair*, the first fully automatic method for 3D hair modeling from a single portrait image, with no user interaction or parameter tuning. Our method efficiently generates complete and high-quality hair geometries, which are comparable to those generated by the state-of-the-art methods, where user interaction is required. The core components of our method are: a novel hierarchical deep neural network for automatic hair segmentation and hair growth direction estimation, trained over an annotated hair image database; and an efficient and automatic data-driven hair matching and modeling algorithm, based on a large set of 3D hair exemplars. We demonstrate the efficacy and robustness of our method on Internet photos, resulting in a database of around 50K 3D hair models and a corresponding hairstyle space that covers a wide variety of real-world hairstyles. We also show novel applications enabled by our method, including 3D hairstyle space navigation and hair-aware image retrieval.

Keywords: hair modeling, image segmentation, data-driven modeling, deep neural network

Concepts: •**Computing methodologies** → **Shape modeling;** *Parametric curve and surface models;*

1 Introduction

Hair is crucial for the perceived realism in computer-generated imagery of digital characters. However, considerable efforts are needed to model realistic hair, due to the intricate structure of hair

strands and the wide variety of hairstyles. Impressive reconstruction of challenging hairstyles have been demonstrated with image-based approaches (e.g., [Paris et al. 2008; Luo et al. 2013]). But these methods typically rely on complex capture setups, which are beyond the reach of non-professional users.

Recently, single-image-based methods [Chai et al. 2012; Chai et al. 2013; Chai et al. 2015; Hu et al. 2015] are proposed to model 3D hair from only one portrait image, along with some user-specified strokes. These techniques enable many interesting consumer-level applications, including portrait pop-ups and hairstyle virtual tryon [Chai et al. 2012], virtual hair cutting and image-space physical hair simulation [Chai et al. 2013], and portrait relighting and 3D-printed portrait reliefs [Chai et al. 2015]. However, one major downside in existing techniques is the requirement of user interaction, which is needed in conjunction with carefully designed priors to tackle the ill-posedness of single-view hair geometry reconstruction. For example, all existing techniques require the user to segment the hair region from the input image. In one state-of-the-art technique [Hu et al. 2015], the user needs to draw a few strokes from root to tip to reveal the hair connectivity and topology. The whole interaction process could take as long as five minutes. The requirement of user interaction hinders wider consumer-level applications of these techniques, for example, mobile applications that allow users to manipulate their portrait images in the photo album, or cloud applications that need to timely process a huge number of photos uploaded simultaneously by Internet users. For such applications, it is highly desirable to have a fully automatic hair modeling pipeline.

In this paper, we introduce *AutoHair*, the first fully automatic method for 3D hair modeling from a single portrait image, with no user interaction or parameter tuning. Given an image as input, we classify it into a few precomputed hair spatial distribution classes, and then generate an accurate hair segmentation and a hair growth direction map, using a novel deep neural network, specifically tailored for the current hair distribution class. This neural network is trained from a large annotated hair image database, and serves as one key component in our system. Next, we perform an image-based search in a large set of 3D hair exemplars generated from public repositories to obtain a few best matching candidates. These candidate models are deformed and refined based on boundary correspondences. We select the deformed candidate with the closest

matching direction map to the input image. Finally, we generate 3D hair strands, guided by both the selected 3D hair exemplar and the direction map of the image.

Our method is able to produce complete and high-quality 3D hair models, that are comparable to those generated by state-of-the-art methods, in which manual interaction is required. The computation time is modest, less than one minute for an 800×800 image. This makes our method an attractive solution for wide deployment in consumer-level applications. For example, once a user takes a portrait photograph or selects one from the photo album, our method can automatically and instantly model the hair in the image, enabling the user to immediately enjoy various operations for manipulating the portrait, such as rendering the portrait in a novel view and 3D-printing portrait reliefs.

The full automation and high performance of AutoHair make it suitable for modeling hairs for Internet photos. The constructed hair models span a *3D hairstyle space* that covers a wide variety of hairstyles observed in real world. Such a large-scale 3D hairstyle space could be valuable in hairstyle design tasks and other portrait-related applications. As a proof of concept, we collect portrait images from Flickr, and construct a hairstyle database of around 50K 3D hair models, called *3D Hairs in the Wild* (3DHW). All hair models in the database are organized into a graph structure, which represents the spanned hairstyle space. We also develop a visualization tool to allow intuitive navigation of the hairstyle space, and demonstrate novel hairstyles generated through hairstyle interpolation.

In summary, the contributions of this paper are:

- We propose, to our knowledge, the first fully automatic method for 3D hair model reconstruction from a single image. The constructed hair models are comparable to those generated by state-of-the-art methods that require user interaction.

- We introduce a novel algorithm that applies deep convolutional neural networks to hair segmentation and hair growth direction estimation. The quality of our results is superior to that of previous work.

- We develop an efficient and automatic data-driven hair matching and modeling algorithm, based on a large set of 3D hair exemplars.

- We construct a large-scale database (3DHW) of around 50K 3D hair models from Internet photos. We make public the database to facilitate further research in this field [1].

2 Related Work

Hair Modeling. Professional skills and laborious manual work are often involved in hair modeling from scratch. We refer readers to the survey of [Ward et al. 2007] for a detailed discussion. One related method to our work is the hair mesh [Yuksel et al. 2009], which generates hair strands from coarse polygonal meshes that encode hair positions and directions. In our data-driven hair modeling algorithm, we also generate hair strands from hair mesh exemplars. However, we need to convert the low-quality hair meshes into a volumetric orientation field for strand generation.

Image-based hair modeling offers a promising way to create compelling hair geometries from captured hair images. Hair acquisition techniques based on multi-view images [Paris et al. 2008; Jakob et al. 2009; Herrera et al. 2012; Echevarria et al. 2014; Luo et al. 2013; Hu et al. 2014a; Hu et al. 2014b] often require complex capture setups and long processing cycles. They are thus not suitable

---

---

for average users, and are too costly for generating a large number of 3D hair models.

Recently, single-image-based hair modeling methods achieve impressive results on single-view, uncalibrated images that are widely available on the Internet, by utilizing different kinds of priors including layer boundary and occlusion [Chai et al. 2012; Chai et al. 2013], a 3D hair model database [Hu et al. 2015], and shading cues [Chai et al. 2015]. Although these methods, especially [Hu et al. 2015], produce high-quality results, various forms of user interactions are needed. For example, binary hair masks generated with interactive segmentation tools are required as input. Chai et al. [2013] need sparse strokes to resolve the local direction ambiguity. Hu et al. [2015] require user-drawn 2D strands for database retrieval. The whole user interaction process could take as long as 5 minutes. Meanwhile, their method takes about 20 minutes to generate a result, which does not scale well for constructing a large number of models from Internet photos. In comparison, our method is fully automatic, and makes the modeling process efficient enough for handling Internet photos.

Semantic Image Segmentation. Hair segmentation is a special case of scene parsing, which is an active research topic in computer vision that tries to understand an image. The problem can also be regarded as a combination of general image segmentation and object recognition. Rabinovich et al. [2007] treat recognition as a post-processing step, after applying an off-the-shelf segmentation method that works directly on super-pixels. However, unsupervised segmentation can result in improper granularity and wrong segment boundaries that affect the effectiveness of recognition. Shotton et al. [2006] try to address this problem by performing recognition and segmentation simultaneously on a joint pixel-level model of texture patterns and spatial relationships. Krähenbühl and Koltun [2011] further improve the method with efficient inference on fully-connected conditional random fields (CRFs). Cheng et al. [2014] propose to jointly estimate both object and attribute labels, to enable interactive verbal refinement.

Recently, deep convolutional neural networks (DCNN) have been adopted in semantic segmentation [Chen et al. 2015; Zheng et al. 2015; Liu et al. 2015; Long et al. 2015]. They introduce Markov random fields (MRFs) or CRFs into the framework to improve the coarse and noisy output of traditional convolutional neural networks (CNN). Dai et al. [2015] reduce the training workload by only requiring annotated bounding boxes to achieve comparable results. We also build our hair segmentation algorithm upon CNN, specifically tailored for hair. We take hair spatial distribution prior into account to cluster different hairstyles into a few classes. Then, an independent segmentation network is trained for each class, following a global classification network, which achieves better results with more accurate boundary and less outliers.

Portrait Parsing. Hair segmentation has also been a part of portrait parsing. Warrel and Prince [2009] label face components including hair, by training a per-pixel adaboost-based unary classifier. For hair segmentation, Wang et al. [2011] combine prior knowledge such as color, position and structure to select hair seed points, and then perform graph-cut to segment the hair region. Luo et al. [2012] produce high-quality masks for facial components, using a hierarchical CNN-based parsing network. Smith et al. [2013] generate facial component masks, by aligning matching exemplar images with the input and optimizing the exemplar weights for the per-pixel labels. The method also identifies a few hair seed points and performs alpha-matting to generate the hair mask. Much of existing work focuses on facial components, but ignores hairs or handles only simple cases. Instead, we work on robust pixel-level hair segmentation, which can be applied to various hairstyles and is accurate enough for modeling.

Shape Set Organization and Exploration. Much work has been developed to organize a set of unordered shape models, and help user navigate and retrieve from the set. Ovsjanikov et al. [2011] propose a correspondence-free navigation interface by letting the user deform a base template shape to match the similar shape. Kim et al. [2012] present an interactive exploration method based on fuzzy correspondences among shape parts. Huang et al. [2013] organize shape collections into a hierarchical structure with qualitative measurements of quartets. O'Donovan et al. [2014] organize fonts according to high-level attributes and perceptual similarity. Hueting et al. [2015] present a joint framework that combines class-labeled images and 3D model collections to resolve camera pose variations. Li et al. [2015] propose a joint embedding space for both 2D images and 3D models, and use CNN to purify and map images to the space. In this paper, we propose to organize a large number of 3D hair models into a graph structure, and develop a visualization tool for intuitive navigation.

3 Overview

Given a single image as input, our pipeline automatically produces the hair segmentation along with a direction map (§4). This information is then combined with precomputed 3D hair model exemplars (§5), to generate the final strand-level hair model (§6).

Specifically, we first classify the input image as one of a few hair spatial distribution classes using a global classifier (§4.2). Next, we generate an accurate hair mask and a coarsely quantized direction map (§4.3), using a novel deep neural network, trained from a large number of annotated hair images (§4.1) and tailored for the current hair distribution class. Then, we perform an image-based search (§6.1) in a large set of 3D hair exemplars generated from public repositories (§5) to obtain a few best matching candidates. These candidate models are further deformed and refined using boundary correspondences, and the model with the direction map closest to that of the input image is selected (§6.2). Finally, we generate 3D hair strands, guided by both the selected 3D model and direction estimation of the input image (§6.3).

4 Hair Image Parsing

We describe in this section how to perform automatic hair segmentation and direction estimation on a single image. Observe that hair is highly complex and variable, making it extremely difficult to accurately segment an image automatically. No rule is universally applicable to describe the characteristics of pixels that correspond to hair in common portrait images. Heuristically, one may assume: 1) hair distributes in regions close to the face, which can be easily detected (e.g., [Wang et al. 2011; Smith et al. 2013]); but this is not valid for long hairs; 2) hair has its narrow prior color distribution [Wang et al. 2011]; however this is not robust against environment lighting and hair dyeing, not to mention the ambiguity with the background; 3) hair pixels are typically of high frequency locally; however they can also be super flat for dark hair and low-quality inputs. In summary, it is almost impossible to find a robust hair segmentation solution based on simple heuristic rules. This suggests that this problem should be solved in a more generic end-to-end way, instead of using hand-crafted features as in previous methods (see comparisons in Fig. 10).

To directly apply a generic segmentation method based on DCNN is not sufficient for our application, with problems like inaccurate boundaries and outlier regions (see Fig. 10). To address these issues, we propose a novel neural network for fully automatic high-quality hair parsing. Specifically, we preprocess a large number of hair images (§4.1), and compute various classes of different spatial distributions of hair from images (§4.2). We then train a classifier

Figure 2: Preprocessing training portrait images. *From left to right: the original image, the hair region mask with super-imposed segmentation strokes, and the direction-based segmentation and the direction map. From top to bottom, original images courtesy of rawartistsmedia and Kris Krüg.*

to determine the hair distribution class, as well as neural networks for each class, to generate a corresponding hair segmentation along with a direction map for a single image (§4.3).

4.1 Preprocessing Training Images

We collect about 100K portrait images from Flickr, and prepare training images according to the following steps:

- **Image selection.** We select 20,000 high-resolution natural photographs with clearly visible human faces and hair, excluding those with over-occluded faces/hair, insufficient illumination or uncommon stylization.

- **Mask segmentation.** We obtain a binary hair region mask M_h for each selected image using stroke-based interactive segmentation and matting tools of PaintSelection [Liu et al. 2009].

- **Direction guidance.** For each selected image, we manually segment the hair region M_h into several subregions with coherent and smoothly varying hair growth directions, by drawing strokes along the subregion boundaries. Within each subregion, a single stroke is added to annotate the strand growth direction. We diffuse the stroke direction to fill the entire subregion and combine it with the densely-calculated per-pixel nondirectional orientation map O [Luo et al. 2013] to achieve the final direction map D. Last, we discretize the direction range $[0, 2\pi)$ into 4 bins (i.e., $[0, 0.5\pi)$, $[0.5\pi, \pi)$, $[\pi, 1.5\pi)$, $[1.5\pi, 2\pi)$) and assign the label to each pixel to obtain the direction label map M_d. Pixels not in the hair region are also assigned a particular label.

At the end of the process, we get 20,000 annotated portrait images, each of which has two label maps: one binary mask map M_h for the hair mask, and one direction label map M_d for the quantized per-pixel hair growth direction in the hair region.

Note that the direction map in this paper is defined as the hair's direction of growth, from the scalp surface toward tips. Existing methods are able to produce fairly robust estimation of the nondirectional local orientation of the hair in a portrait image. However, it is difficult to determine the correct growth direction of individual strands – there is a directional ambiguity as described in [Chai et al. 2013]. The hair direction label map is intended to be combined with
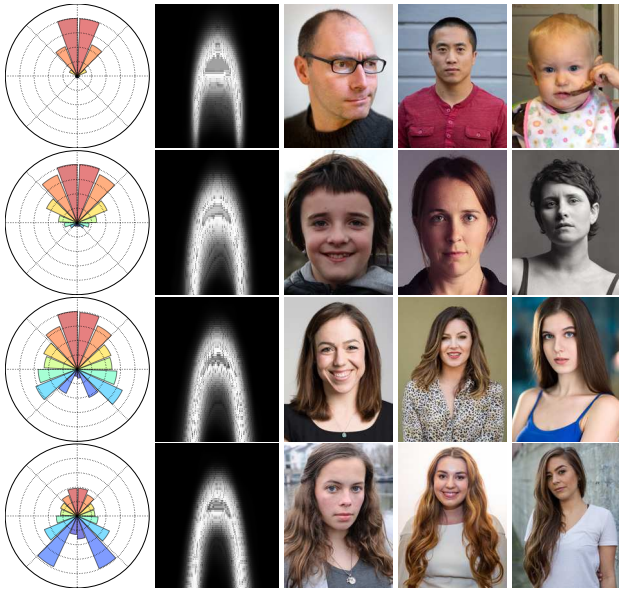
Figure 3: Visualization of hair distribution classes. *In each row, we show a visualization of the representative circular histogram and the average hair mask for a particular hair distribution class. From left to right, top to bottom, original images courtesy of Barney Moss, Kris Krüg, Rachael Ludwick, Ordiziako Jakintza Ikastola, Pat David, Jay DeFehr, rawartistsmedia, Kris Krüg, Ivan Malafeyev, Loren Kerns, rawartistsmedia, and Kris Krüg.*

the nondirectional orientation information to resolve this ambiguity. Since we only need to decide the direction sign for each pixel, a coarsely-quantized label map works well for this purpose.

4.2 Computing Hair Distribution Classes

Part of the difficulties in automatic hair segmentation comes from the myriad of hair shapes and distributions. By leveraging reference face localization, we cluster all images for a few fuzzy clusters of hair distribution. Each cluster represents a class of hairstyles with similar spatial distribution around the face, which is used for high-quality hair parsing, as will be described in the next subsection. We first detail how to compute the hair distribution classes as follows.

For each annotated image I, we first detect and localize facial landmarks using the robust face alignment method of [Cao et al. 2012], and rigidly register I to I' in a reference face coordinate system, for scale and up-direction rectification. As shown in Fig. 3, we then construct a circular distribution histogram $\mathcal{H}$ with $n_{\mathcal{H}}$ bins (16 in our implementation) in the polar coordinate system around the origin (face center) o_{face}. Each bin counts hair pixels, whose polar angles fall in the range of that bin. After normalization, $\mathcal{H}$ can be regarded as the $n_{\mathcal{H}}$-dimensional distribution feature vector for the image. Finally, based on these distribution feature vectors, we cluster the selected images into several groups (four in our experiments) with K-means clustering [Lloyd 1982]. During clustering, we calculate the distance between two histograms $\mathcal{H}_a$ and $\mathcal{H}_b$ using the L_1-based Earth Mover's Distance (EMD) [Ling and Okada 2007]:

$$d_{\mathcal{H}}(\mathcal{H}_a, \mathcal{H}_b) = \min_{\alpha(i,j)} \sum_{i=1}^{n_{\mathcal{H}}} \sum_{j=1}^{n_{\mathcal{H}}} \alpha(i,j) d_b(i,j),$$

$$\text{s.t.} \quad \sum_i \alpha(i,j) = \mathcal{H}_i, \quad \sum_j \alpha(i,j) = \mathcal{H}_j, \quad \alpha(i,j) \geq 0, \quad (1)$$

where $d_b(i,j)$ is the circular distance between bin i and j: $\min(|i-j|, n_{\mathcal{H}} - |i-j|)$. The center of cluster g is computed as a
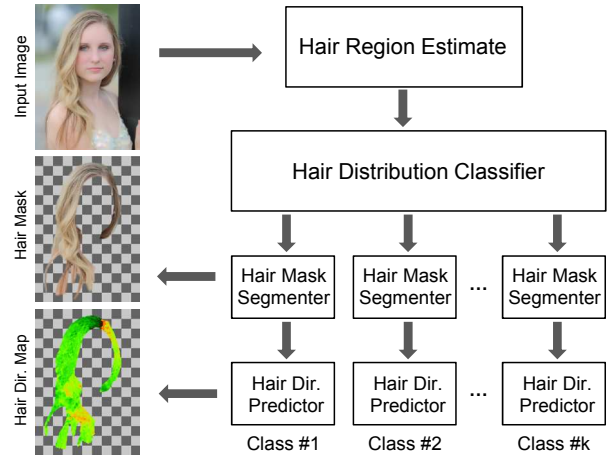

Figure 4: Our hair image parsing pipeline. *The pipeline starts with a hair region estimate on the top level, then a hair distribution classifier, followed by a hair mask segmenter and a direction segmenter for each hair distribution class. Photo credit: Bob HARRIS.*

representative histogram $\mathcal{H}_g$, which has a minimal sum of distances with all other elements in the cluster.

4.3 Computing the Hierarchical Network

With the preprocessed training images and hair distribution classes computed in §4.2, we design our hair image parsing pipeline in a top-down hierarchy as follows (Fig. 4):

- **Hair region estimate.** A face detector [Cao et al. 2012] is employed to generate an estimate of the hair region candidates (i.e., hair bounding boxes) that will be used by the hair classifier in the next layer.

- **Hair classifier.** With the hair region candidates, an R-CNN (regions with CNN features) network [Girshick et al. 2014] is trained to select one hair distribution class with the maximal score and pass the image to its class-specific segmenter.

- **Hair mask segmenter.** The per-class hair mask segmenter is trained with DCNN that outputs a downsampled probability map. We upsample the map and apply fully-connected CRFs [Krähenbühl and Koltun 2011] to refine the often over-smoothed, initial segmentation result M.

- **Hair direction predictor.** In addition to the mask segmenter, a direction predictor generates a direction label map. We then upsample the direction label map, and combine it with the pixel orientations inside the mask as the final direction map D as in §4.1.

In the beginning of the pipeline, we run the face alignment algorithm of [Cao et al. 2012] to detect a set of facial feature points, which are used to register the image in the reference face coordinate system as in §4.2. Next, for each hair distribution class, we adapt its 20 typical hair region bounding boxes for the face in the image through rotation and scaling, resulting in a set of hair region candidates for the image. The typical hair region bounding boxes are generated by pre-clustering the hair bounding boxes of the training images inside each distribution class. The region candidates are then cropped and fed to the hair classifier for processing independently. We note that this hair region estimation is executed only once for the testing stage, not in the training stage, where the hair region of each training image is known.

The hair classifier exactly follows the structure of the R-CNN network described in [Girshick et al. 2014], and is fine-tuned on la-
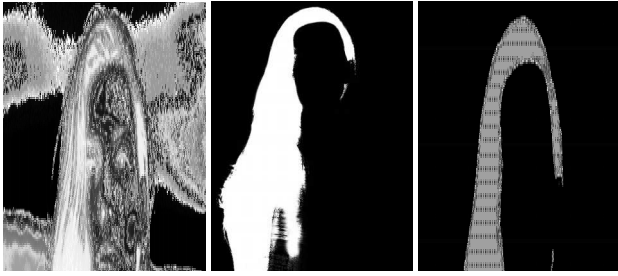
Figure 5: Segmentation with different distribution classes. *From left to right: the input image, the segmentation result with the (correct) long-hair class, and the result with the (wrong) short-hair class. Original image courtesy of Mycatkins.*

beled hair regions in our dataset. The CNN network structures of mask segmenters and direction predictors are designed based on the publicly-available network VGG16 [Simonyan and Zisserman 2014], which is a pre-trained image classification network on ImageNet with 1000 labels, similar to [Chen et al. 2015] and [Liu et al. 2015]. We make some changes to VGG16 to output pixel unary labels (2 for mask segmenters and 5 for direction predictors). First, the last two 2×2 max-pooling layers are removed to increase the network layer resolution, and the receptive field of their directly following convolution layers are enlarged from 3×3 and 7×7 to 5×5 and 25×25 respectively with padded zeros. Second, all fully-connected layers are changed to convolution layers to make the single-label classification network compatible with our dense per-pixel unary label segmenter. Third, in the training stage, the loss layer calculates the sum of cross entropy over the entire image between the output and the ground-truth labels in the training image, with its resolution downsampled 8× due to the three max-pooling layers in VGG16. Finally, in the testing stage, the output label map is bilinearly upsampled to the original image size and refined with a fully connected CRF.

Our implementation is largely built upon the open source convolutional architecture, Caffe [Jia 2013]. All images are resized to 512×512 for both the training and testing stages. During runtime testing, given an input image I, the face detector first aligns the image to the face coordinate and generates hair region candidates around the face. The hair classifier then iteratively tests each candidate to select a distribution class with the highest score. The pair of hair mask segmenter and direction predictor corresponding to the distribution class are eventually applied to I. The mask segmenter outputs an equal-sized hair mask M_I with an alpha channel, and the direction predictor outputs a direction label map which is combined with the nondirectional orientation map to produce the final direction map D_I.

Our algorithm is not only robust but also efficient, due to the following reasons: 1) region candidate generation often occupies a large portion of R-CNN execution time [Ren et al. 2015]; our hair region candidates avoid an exhaustive search [Uijlings et al. 2013] and generate robust results; 2) even if the hair classifier falsely assigns wrong labels to some challenging images, the subsequent segmenter still shares certain level of generality to produce reasonable results (see Fig. 5); 3) by restricting training data to similar hair distributions, the hair segmenter is more specific and achieves higher-quality results.

5 Hair Model Exemplars

In this section, we describe how to generate and organize 3D hair model exemplars, which will be later used in the strand-level hair

reconstruction, in conjunction with the hair segmentation and the direction map estimated from the input image.

Hu et al. [2015] demonstrate that, with the help of a rich set of hair geometry exemplars, excellent hair modeling results can be obtained, which match the image cues and retain the realism of hair geometry. However, in this method, user interactions are needed to reveal the global hair structure, which cannot be robustly estimated from local image information. A subsequent time-consuming optimization is also required to adapt the exemplars to the user-specified global structure. In comparison, we propose a novel exemplar construction solution that exhaustively generates plausible combinations by remixing initial models. The exemplars are organized for compact storage and efficient run-time match that requires no user interaction at all. Essentially, we avoid user interactions by bringing forward the model remixing step from runtime in [Hu et al. 2015] to the precomputation stage.

5.1 Preprocessing

Following [Hu et al. 2015], we collect an initial set of about 300 3D models of different hair styles $\{H\}$, downloaded from public repositories online [Electronic Arts 2014], which have already been aligned to the same reference head model. All these models are composed of a large number of independent thin polygon-strips $\{S^H\}$. Each strip represents a coherent hair wisp, with relative strand growth directions encoded in parametric texture coordinates. Similar to [Hu et al. 2015], we take a volumetric view of the entire hair and treat these polygon-strips as sparse 3D orientation cues in the volume. Instead of directly converting each model to a densely diffused 3D orientation volume and growing strands explicitly afterwards, we keep this raw polygonal representation for processing in later stages, since they are more compact and easier to manipulate.

Before proceeding to the next steps, we further clean these models to improve the mesh quality:

- **Dangling strips.** For strips not connected to the scalp, we find strips that are both connected to the scalp and spatially nearest to the end points of the dangling ones, and smoothly connect them to form a longer strip that extends to the scalp.

- **Over-wide strips.** For coarse strips with a width larger than a threshold w_{max}^S (one-tenth of the head radius in our implementation), we uniformly subdivide and separate the strip along the growth direction into two separate ones. This process is repeated until we reach a valid width.

5.2 Populating the Exemplars

We decompose each 3D hair model in the initial set into strip clusters, and recombine these clusters to form a large number of new exemplars.

Model decomposition. We use a simplified internal representation for each hair strip S_i: an equally-spaced polyline $\{\boldsymbol{p}_1^i, \boldsymbol{p}_2^i, ..., \boldsymbol{p}_{n_i}^i\}$ that goes through the strip center, plus an average radius r_i. For each hair model, we cluster its strips into a few strip groups with minimal mutual distances, defined as:

$$d_S(S_a, S_b) = \frac{\sum_{i=1}^{\min(n_a, n_b)} \max(\|\boldsymbol{p}_i^a - \boldsymbol{p}_i^b\| - r_a - r_b, 0)}{\min(n_a, n_b)}. \quad (2)$$

In the end, we decompose each hair model into around ten strip clusters.

Model recombination. We create novel hair exemplars between pairs of existing ones to enrich our exemplar set. Note that all the hair models downloaded from public repositories are aligned to the
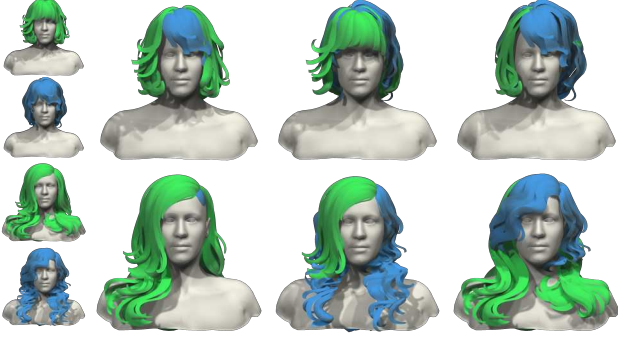
Figure 6: Populating hair model exemplars by remixing. *In each row, we show two original hair model exemplars (left), and three novel shapes generated by recombining the strip clusters of the original models (right).*

same reference head model. Between each hair pair within these models, one model is chosen to be the target H_t, and the other the source H_s. The recombination process is performed by exhaustively testing every strip cluster subset H'_s of H_s. For each H'_s, we add it to H_t with its position unchanged. Given the new combination $H_t + H'_s$, we remove strip clusters H'_t from H_t that are conflict with the 3D orientation field introduced by H'_s to produce a combined model $H_t - H'_t + H'_s$. To avoid generating unnatural hair shapes in combining drastically different models, we further enforce that the volume boundary shape of H'_s should not be largely different from that of H_t (30% of the head radius away from the boundary of H_t in our implementation). Eventually, we enrich our 3D hair exemplar set to contain more than 40K models.

5.3 Organization

After generating remixed hair models, we organize them in the ascending order of the frontal view area of the hair mask. This permits early rejection in the hair image matching detailed in the next section.

For similar efficiency consideration, we further generate two types of thumbnails for each hair exemplar H:

- **Mask thumbnail M_H^*.** This is a binary image of the projected hair. To avoid the influence of possible stray strips, we apply Gaussian filter to smooth the result.

- **Direction thumbnail D_H^*.** We render the direction thumbnail by encoding projected hair direction with colors, using the XYZ components of the direction vector as the RGB channels.

To handle non-frontal views, we uniformly sample 6 angles for both pitch and yaw in $[-\pi/4, \pi/4]$, resulting a 2D array of 6×6 thumbnails for each hair exemplar. All thumbnails are downsampled to the resolution of 100×100 for efficient evaluations.

6 Data-Driven Hair Modeling

In this section, we describe how to produce high-quality hair model H_I that matches a single input image I, with the hair mask M_I, the direction map D_I (§4), and a large number of 3D hair exemplars $\{H\}$ (§5), computed in previous stages. Our method first performs an image-based search over 3D hair exemplars to find a few matching candidates (§6.1). These exemplar models are then deformed and refined using boundary correspondences, and the model with the direction map closest to that of the input image is selected (§6.2). Final strand-level hair geometry is generated with

the guidance from both the selected 3D model and the direction estimation of the input image (§6.3).

6.1 Image-Based Candidate Search

The first step in our hair modeling algorithm is to retrieve a few good-matching candidates from the large set of exemplars. The sheer size of our exemplar set prohibits straightforward exhaustive element-wise comparisons between the input image and the precomputed thumbnails. Thanks to the variation in human hair shape and structure, we apply early rejection on candidates that are drastically different from the input image to improve the search efficiency. Specifically, we perform a two-step search based on the frontal view area of the hair mask and the precomputed thumbnails.

- **Area test.** We align the input image with the same reference coordinate system used in constructing exemplars, according to facial landmarks. We then compare the hair mask area of the image $|M_I|$ and that of the exemplar $|M_H|$. The candidates with $|M_H|$ in the range of $(0.8|M_I|, 1.25|M_I|)$ are retained. This step prunes a large number of candidates from the more expensive thumbnail-based comparisons described below.

- **Thumbnail matching.** For each candidate that passes the area test, we compare its thumbnails of the hair region mask M_H^* and the direction map D_H^* with those of the input image, (M_I^*, D_I^*). The exemplars with matching boundary shapes and internal structures (i.e., the pixel direction field inside the hair region) are selected. For an input image with a non-frontal view, (M_H^*, D_H^*) with the closest pitch and yaw in the precomputed array of thumbnails are chosen for comparison. We calculate the distance between masks M_H^* and M_I^* based on the distance field W_I^* of the boundary of M_I^*, similar to [Balan et al. 2007]:

$$d_M(M_H^*, M_I^*) = \frac{\sum_{i \in M_H^* \oplus M_I^*} |W_I^*(i)|}{|M_I^*|}, \qquad (3)$$

where $M_H^* \oplus M_I^*$ represents the symmetric difference between two masks. The distance between direction maps D_H^* and D_I^* is the sum of the pixel direction (angle) differences $d_d \in [0, \pi)$:

$$d_D(D_H^*, D_I^*) = \frac{\sum_{i \in M_H^* \cap M_I^*} d_d(D_H^*(i), D_I^*(i))}{n_{M_H^* \cap M_I^*}}. \qquad (4)$$

We keep exemplars $\{H\}$ with $d_M(M_H^*, M_I^*) < 4$ and $d_D(D_H^*, D_I^*) < 0.5$ as our final matching candidates.

In our experiments, typically 5-40 candidates are selected for common portrait images. If more than 50 candidates are found, we keep 50 with the smallest d_D for efficiency.

6.2 Hair Shape Deformation

We deform each selected candidate from the image-base search to better align with the hair region in the input image. First, we compute boundary correspondences between the hair region of the input image and that of the rendered image of a candidate. Next, the correspondences are interpolated to the entire hair region, using a globally smooth warping function. Based on the correspondences, we compute an optimal deformation for the candidate.

Boundary matching. For each candidate H, we first transform it to the same pose as the face in I, and render its mask and direction map (M_H, D_H) as in §5.3. The only difference is that now we render maps at the same resolution as the input image, instead of that of thumbnails. Then, we uniformly sample 200/2000 2D points $\{\boldsymbol{P}^H\}$ / $\{\boldsymbol{P}^I\}$ on the mask boundaries of M_H / M_I. For each boundary point $\boldsymbol{P}_i^H$ / $\boldsymbol{P}_j^I$, we denote its position as $\boldsymbol{p}_i^H$ / $\boldsymbol{p}_j^I$ and outward normal as $\boldsymbol{n}_i^H$ / $\boldsymbol{n}_j^I$. Note that all the above vectors are 2D.
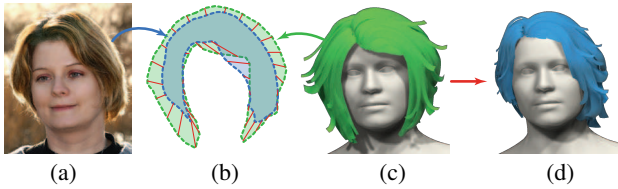
| (a) | (b) | (c) | (d) |

Figure 7: Hair shape deformation. *We first compute the boundary correspondences (b) between the hair region of an input image (a) and that of the rendered image of a candidate model (c). The correspondences are then interpolated to the entire hair region, based on which we compute an optimal deformation of the candidate (d). Original image courtesy of Bob HARRIS.*

We compute point-to-point correspondences $\mathcal{M}(\{\boldsymbol{P}^H\} \rightarrow \{\boldsymbol{P}^I\})$ to match the boundaries. For each point $\boldsymbol{P}_i^H$ on the hair mask boundary of a candidate model, $\boldsymbol{P}_{\mathcal{M}(i)}^I$ is its optimal corresponding point on the mask boundary of the input image, computed by minimizing the following matching cost function:

$$\arg\min_{\mathcal{M}} \sum_{\boldsymbol{P}_i^H} \left(E_p(\boldsymbol{P}_i^H) + E_e(\boldsymbol{P}_i^H, \boldsymbol{P}_{i+1}^H) \right). \qquad (5)$$

Here E_p and E_e are energy terms that measure the matching quality of points and edges, respectively. E_p penalizes the distance between each pair of corresponding positions and normals. The weight λ_n is set as 10 in our implementation. E_e encourages the mapping $\mathcal{M}$ to preserve the length of the original boundary:

$$E_p(\boldsymbol{P}_i^H) = \|\boldsymbol{p}_i^H - \boldsymbol{p}_{\mathcal{M}(i)}^I\|^2 + \lambda_n(1 - \boldsymbol{n}_i^H \cdot \boldsymbol{n}_{\mathcal{M}(i)}^I)^2, \qquad (6)$$

$$E_e(\boldsymbol{P}_i^H, \boldsymbol{P}_{i+1}^H) = (\|\boldsymbol{p}_i^H - \boldsymbol{p}_{i+1}^H\| - \|\boldsymbol{p}_{\mathcal{M}(i)}^I - \boldsymbol{p}_{\mathcal{M}(i+1)}^I\|)^2. \qquad (7)$$

We minimize the matching cost function in the framework of Hidden Markov Model (HMM), using the classic Viterbi algorithm [Forney Jr 1973]. We treat $\{\boldsymbol{P}^H\}$ and $\{\boldsymbol{P}^I\}$ as sets of states and observations, respectively. The solution, which is the mapping $\mathcal{M}(\{\boldsymbol{P}^H\} \rightarrow \{\boldsymbol{P}^I\})$, is an optimal path in the Trellis table, expanded by states and observations. In the framework of HMM, we convert the point and edge matching energies to the emission and transition probabilities, and set the initial state as the single point with largest height value (at the top of head).

Warping function. We further interpolate the boundary correspondences to every pixel in the mask map M_H via a global smooth warping function $\mathcal{W}(M_H \rightarrow M_I)$, using the Thin-Plate-Spline (TPS) method [Bookstein 1989]:

$$\arg\min_{\mathcal{W}} \sum_{\boldsymbol{P}_i^H} \|\mathcal{W}(\boldsymbol{p}_i^H) - \boldsymbol{p}_{\mathcal{M}(i)}^I\|^2 + \lambda E_W(\mathcal{W}). \qquad (8)$$

Here $\mathcal{W}(\boldsymbol{p}_i^H)$ is the corresponding position in I for point $\boldsymbol{p}_i^H$. The term $E_W(\mathcal{W})$ measures the Frobenius norm of the second-order partial derivatives of the correspondence matrix $\mathcal{W}$, representing the smoothness of the warping function. The weight λ is set as 1000 in our implementation.

Deformation optimization. Finally, we deform each vertex $\boldsymbol{v}$ in the candidate hair model H to $\boldsymbol{v}'$ by minimizing the following energy:

$$\arg\min_{\boldsymbol{v}'} \sum_{\boldsymbol{v}_i \in \boldsymbol{V}_H} \left(\|\boldsymbol{v}_i' - \mathcal{W}(\boldsymbol{v}_i)\|^2 + \lambda_s \left\| \Delta\boldsymbol{v}_i' - \frac{\delta_i}{|\Delta\boldsymbol{v}_i'|} \Delta\boldsymbol{v}_i' \right\|^2 \right). \qquad (9)$$

Here $\boldsymbol{V}_H$ is the vertex set of H. $\mathcal{W}(\boldsymbol{v}_i)$ is the corresponding position of $\boldsymbol{v}_i$, with the XY coordinates warped by $\mathcal{W}$, while keeping the Z coordinate unchanged. Δ is the discrete mesh Laplacian operator based on the cotangent formula [Desbrun et al. 1999], δ_i is the

magnitude of the Laplacian coordinates of vertex $\boldsymbol{v}_i$ in the original model H. The weight λ_s is set to 1 in our implementation. The first term measures the sum of squared distances between the position of a vertex to its deformation target via $\mathcal{W}$. The second term is a Laplacian regularization term, which aims to preserve local geometric features in the original shape. The energy can be minimized using the inexact Gauss-Newton method as described in [Huang et al. 2006]. Now we have refined hair models $\{H'\}$ that better match the hair region in the input image. Fig. 7 shows a deformation example.

6.3 Final Hair Generation

Given the deformed candidates $\{H'\}$, we perform a final direction distance test (see §6.1) on full-resolution maps. We select the exemplar model H^* with the closest matching direction map to that estimated from the input image.

Following previous solutions [Paris et al. 2008; Chai et al. 2013; Hu et al. 2015], we convert H^* to a 3D direction volume within the bounding box of the entire mesh, and perform direction diffusion inside the volume by treating the direction vectors given by H^* and head surface normals near the scalp region as constraints. We then grow 10,000 strands from seeds uniformly sampled on the scalp, with the guidance from the volumetric direction field. Finally, these strands are deformed, according to projected image direction map D_I, as in [Hu et al. 2015].

7 Results and Applications

We have implemented the described method on a PC with a quad-core Intel i7 CPU, 32GB of memory and an NVIDIA GeForce GTX 970 graphics card. We demonstrate the effectiveness of our method on many Internet images with various hairstyles. Fig. 1, 8, and 16 show some examples.

Given a single-view image, we first locate a set of facial feature points using a face alignment algorithm [Cao et al. 2012], and fit a 3D morphable head model [Blanz and Vetter 1999] for the subject in the image to match the feature points, in the same way as in [Chai et al. 2012]. Then our hair modeling method generates a complete strand-level 3D hair model that is both visually plausible and matches the input image. For a typical image with a size of 800×800, the entire processing steps can be finished within one minute, including less than 3 seconds for hair parsing, about 20 seconds for 3D hair database matching and deformation, and less than 30 seconds for final hair strands generation.

We outsourced the preprocessing of the hair image database to a commercial company. It takes on average one minute to process an image. The 3D hair exemplar population takes less than ten hours, and the training of our neural network takes around eight hours.

7.1 Comparisons

Hair modeling. We compare our method with state-of-the-art single-view hair modeling methods that require user interaction [Chai et al. 2013; Hu et al. 2015] (see Fig. 8). Note that we replace their head and bust models with ours for consistency. As shown in the figure, our method produces results with a quality comparable to that of [Hu et al. 2015]. The results closely match the input image in the original view, and are visually plausible when rotated. Our method is fully automatic and takes less than one minute to process an image. In comparison, the method of [Hu et al. 2015] requires manual segmentation and strands guidance, and takes around 20 minutes for each image.
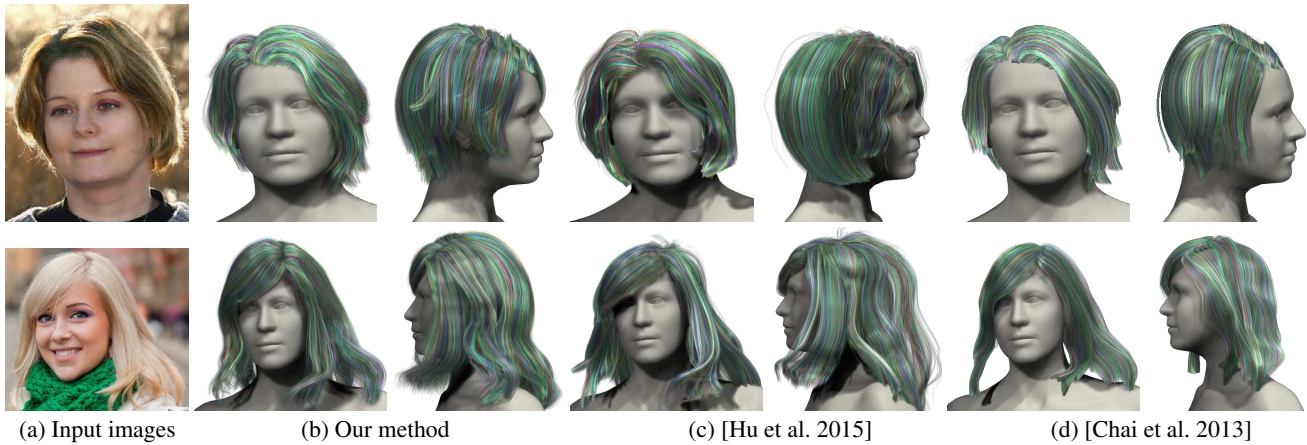
(a) Input images   (b) Our method   (c) [Hu et al. 2015]   (d) [Chai et al. 2013]

Figure 8: Comparisons with state-of-the-art hair modeling techniques. *From left to right: input images, our method, [Hu et al. 2015] and [Chai et al. 2013], respectively. From top to bottom, original images courtesy of Bob HARRIS and Chris Zerbes.*
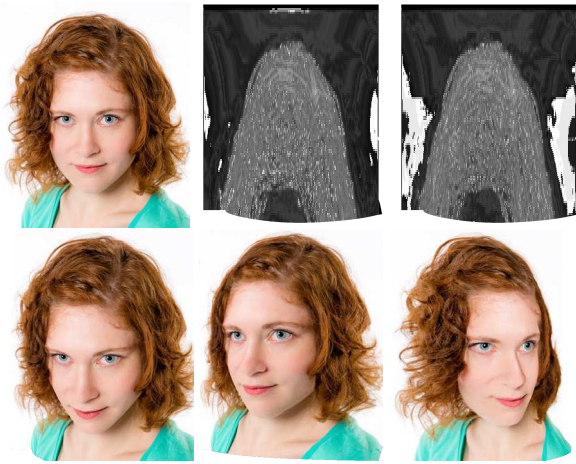

Figure 9: Combining our pipeline with height-map-oriented hair modeling. *Top (left to right): the input image, our result, and the result generated by [Chai et al. 2015]. Bottom: rendering the portrait in three views. Original image courtesy of Bob HARRIS.*

Note that our modeling pipeline can also be used in conjunction with hair modeling techniques that produce a height map result, e.g., [Chai et al. 2015]. This is achieved by passing the estimated hair segmentation and direction map from our pipeline as input to those techniques. As shown in Fig. 9, our result is comparable in quality with that of [Chai et al. 2015].

Hair parsing. To evaluate the accuracy of our automatic hair parsing algorithm, we randomly select 5,000 images from our labeled image database for training and use the remaining 15,000 ones for validation. We compare our algorithm against both the state-of-the-art automatic hair parsing methods [Wang et al. 2011; Smith et al. 2013] and a general deep-learning-based semantic image segmentation method [Chen et al. 2015], all using the same training dataset. As shown in Fig. 10, our method generates better results than the other three methods. Using the commonly-adopted metric of IOU (Intersection Over Union) to measure the segmentation accuracy with respect to the ground truth segmentation on the validation dataset, we report that our method can achieve as high as 90% accuracy, while the other three methods [Chen et al. 2015; Smith et al. 2013; Wang et al. 2011] achieve 78%, 62% and 48% accuracy respectively.

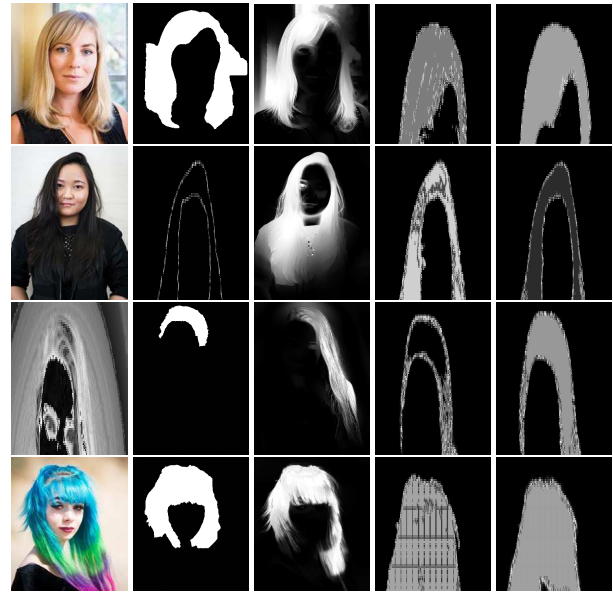Fig. 11 demonstrates that using more training images can improve

Figure 10: Comparisons with other automatic hair segmentation methods. *From left to right: input images, segmentation results using [Wang et al. 2011], [Smith et al. 2013], [Chen et al. 2015] and our algorithm, respectively. From top to bottom, original images courtesy of rawartistsmedia, rawartistsmedia, Ralph, and Pranavian.*

the segmentation accuracy for both our algorithm and [Chen et al. 2015]. But our algorithm cannot achieve significant accuracy improvement with more than 5K training images, which is also true for the general deep-learning-based method [Chen et al. 2015]. Also note that since the network structure of our hair mask segmenter is similar to that of [Chen et al. 2015], the accuracy gain of our method largely comes from the extra layer of our hierarchical hair parsing pipeline, i.e., the hair classifier.

7.2 Applications

Portrait manipulation. Our method can be adopted to automate previous work on 3D-aware portrait manipulation [Chai et al. 2012; Chai et al. 2015], where manual annotations are required. We show examples of novel-view portrait rendering (Fig. 9) and printable high-relief model generation (Fig. 12). For applications that require a rough 3D body shape, we train a body segmentation network, sim-
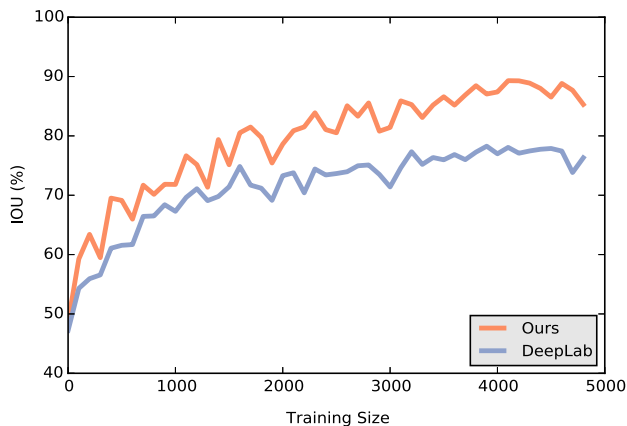
Figure 11: Hair segmentation accuracy curves. *Using more images used for training can improve the segmentation accuracy. But with more than 5K training images, both our algorithm and the DCNN of [Chen et al. 2015] cannot achieve significant accuracy improvement further.*

Figure 12: 3D-printed portrait reliefs. *Three examples generated by our method. From left to right, original images courtesy of Bob HARRIS, Bob HARRIS, and Kris Krüg.*

Figure 13: Physically-based hair animation. *The hair model constructed by our method can be used for physically-based simulation.*

ilar to the hair mask segmenter in §4.3, to automatically segment the body region. We then adopt the estimation method in [Chai et al. 2015] to generate a rough body shape.

Physically-based animation. The strand-based hair models generated by our method are physically correct – all strands are fixed on the scalp and grow naturally in space, making them suitable for physically-based simulation. Fig. 13 shows an animation generated by a mass-spring dynamic model [Selle et al. 2008] (see the accompanying video).

3D hairstyle space navigation. Our method makes it possible to build a large number of 3D hair models from Internet photos. As a proof of concept, we collect portrait photos from Flickr, and construct a hairstyle database of about 50K 3D hair models. Due to its sheer size, our database covers a wide range of real-world hairstyles, spanning a *3D hairstyle space*. Navigating in such a space could be very useful in hairstyle design and other portrait-related applications.
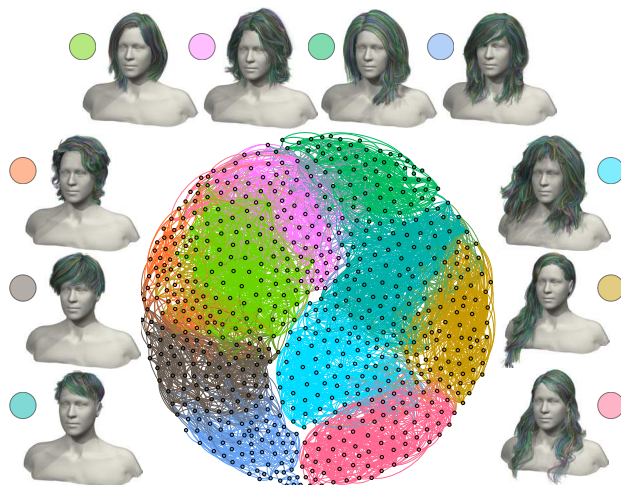

Figure 14: 3D hairstyle space. *A visualization of the graph structure of our 3DHW database, with one random representative hair model of each cluster shown in a corresponding cluster color. For brevity, we only display a sub-graph with 1000 hair models here.*

Figure 15: Hair-aware image retrieval. *From left to right: the input image, four retrieved images with the most similar hair geometry style (top) and hair boundary shapes (bottom). From left to right, top to bottom, original images courtesy of Kris Krüg, Conte Oliver, Brian Tomlinson, anisahullah, Paolo Perini, Dallas Floer Photography, rawartistsmedia, rawartistsmedia, Paulo Guereta, and rawartistsmedia.*

Specifically, we first construct our 3D hairstyle space by representing all hair models as nodes in a graph, and connecting node to its 20 nearest neighbors with respect to a hair-to-hair distance. The distance is defined as an Earth Mover's Distance between two models H_a and H_b as:

$$d_H(H_a, H_b) = \min_{\alpha(s_i, s_j)} \sum_{s_i \in H_a} \sum_{s_j \in H_b} \alpha(s_i, s_j) d_s(s_i, s_j),$$

$$\text{s.t.} \sum_{s_j \in H_b} \alpha(s_i, s_j) = 1, \sum_{s_i \in H_a} \alpha(s_i, s_j) = 1. \quad (10)$$

We uniformly sample each strand into 10 vertices $v_1^s, v_2^s, ..., v_{10}^s$, and define the strand distance as $d_s(s_i, s_j) = \sum_{k=1}^{10} \|v_k^i - v_k^j\|$. We then partition the graph into 10 clusters using [Karypis and Kumar 1998], to minimize the total cut edge weight and balance the size of each cluster. Finally, we map and visualize the graph in 2D with the visualization tool Gephi [Gephi 2016] (see Fig. 14). The user can navigate through a huge number of hair models, by simply moving the cursor across the graph in 2D. In addition, the user can select one model and visualize its direct neighbors in the graph. We can also generate a smooth animation of changing hairstyles, by interpolating pairwise 3D hair models along a given path on the

Figure 16: Hair modeling results. *From left to right: the input image, automatic mask segmentation and direction estimation results, the matching hair shape after deformation, and the final hair model in three views. Original images courtesy of Bob HARRIS (the first four rows) and Kris Krüg (the bottom row).*

graph, using the method of [Weng et al. 2013]. Please refer to the accompanying video for a live navigation demo.

Hair-aware image retrieval. Our method also enables efficient hair-aware image retrieval that searches for portraits with most similar hair to a given input. Specifically, we use our method to process

all portrait images to calculate their hair masks and direction maps. Given an input image, we automatically calculate its hair mask and direction map, and perform image-based hair matching, based on the mask and direction distance (d_M, d_D) described in §6.1 to find the best matching portrait images. The matching distance is defined as $d = d_M + \lambda_D d_D$, where λ_D is a relative weight. As shown in

Fig. 15, the user can use different λ_D to search for images that are similar in boundary shape or hair geometry style.

8 Conclusion

We have presented AutoHair, the first fully automatic method for 3D hair modeling from a single portrait image. Our method can generate high-quality hair models with modest computation time, making it suitable for processing Internet photos. We have constructed a large-scale database (3DHW) of around 50K 3D hair models. We also demonstrated novel applications enabled by our method, including hair-aware image retrieval and 3D hairstyle space navigation.

Figure 17: Failure cases. *Our method generates less satisfactory results for buzz-cut hairstyles (left) and tilted head poses (right). From left to right, original images courtesy of Kris Krüg and Francesca Giachetta.*

Our method is subject to a few limitations, which might be addressed in future work. First, we rely on the face alignment method of [Cao et al. 2012] to register input images to a common reference frame, which may fail in extreme cases like side views of the face. We hope that this problem will be alleviated as more robust face alignment techniques are developed. Moreover, we are not able to recover the ground-truth hair strands at the back view of the image, as no such information is available. Note that this is a common issue with existing single-view hair modeling techniques. Additionally, similar to [Hu et al. 2015], our method could produce less satisfactory results when the 3D hair exemplar database does not contain models matching the input image (see Fig. 17). Also, our method does not handle constrained hairstyles such as braids and buns, unlike the method of [Hu et al. 2014b]. It would be interesting to extend our method to model these more complicated cases.

Acknowledgment

References

BALAN, A. O., SIGAL, L., BLACK, M. J., DAVIS, J. E., AND HAUSSECKER, H. W. 2007. Detailed human shape and pose from images. In *CVPR*, 1–8.

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

BOOKSTEIN, F. L. 1989. Principal warps: thin-plate splines and the decomposition of deformations. *Trans. PAMI 11*, 6, 567–585.

CAO, X., WEI, Y., WEN, F., AND SUN, J. 2012. Face alignment by explicit shape regression. In *CVPR*, 2887–2894.

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

CHAI, M., WANG, L., WENG, Y., JIN, X., AND ZHOU, K. 2013. Dynamic hair manipulation in images and videos. *ACM Trans. Graph. 32*, 4, 75:1–75:8.

CHAI, M., LUO, L., SUNKAVALLI, K., CARR, N., HADAP, S., AND ZHOU, K. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph. 34*, 6, 204:1–204:10.

CHEN, L.-C., PAPANDREOU, G., KOKKINOS, I., MURPHY, K., AND YUILLE, A. L. 2015. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*.

CHENG, M.-M., ZHENG, S., LIN, W.-Y., VINEET, V., STURGESS, P., CROOK, N., MITRA, N. J., AND TORR, P. 2014. Imagespirit: Verbal guided image parsing. *ACM Trans. Graph. 34*, 1, 3:1–3:11.

DAI, J., HE, K., AND SUN, J. 2015. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 1635–1643.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH '99*, 317–324.

ECHEVARRIA, J. I., BRADLEY, D., GUTIERREZ, D., AND BEELER, T. 2014. Capturing and stylizing hair for 3D fabrication. *ACM Trans. Graph. 33*, 4, 125:1–125:11.

ELECTRONIC ARTS, 2014. The Sims Resource. http://www.thesimsresource.com/.

FORNEY JR, G. D. 1973. The Viterbi algorithm. *Proceedings of the IEEE 61*, 3, 268–278.

GEPHI, 2016. The Open Graph Viz Platform. https://gephi.org.

GIRSHICK, R., DONAHUE, J., DARRELL, T., AND MALIK, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 580–587.

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, 146:1–146:9.

HU, L., MA, C., LUO, L., AND LI, H. 2014. Robust hair capture using simulated examples. *ACM Trans. Graph. 33*, 4, 126:1–126:10.

HU, L., MA, C., LUO, L., WEI, L.-Y., AND LI, H. 2014. Capturing braided hairstyles. *ACM Trans. Graph. 33*, 6, 225:1–225:9.

HU, L., MA, C., LUO, L., AND LI, H. 2015. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph. 34*, 4, 125:1–125:9.

HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L.-Y., TENG, S.-H., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Subspace gradient domain mesh deformation. *ACM Trans. Graph. 25*, 3, 1126–1134.

HUANG, S.-S., SHAMIR, A., SHEN, C.-H., ZHANG, H., SHEFFER, A., HU, S.-M., AND COHEN-OR, D. 2013. Qualitative organization of collections of shapes via quartet analysis. *ACM Trans. Graph. 32*, 4, 71:1–71:10.

HUETING, M., OVSJANIKOV, M., AND MITRA, N. J. 2015. CrossLink: Joint understanding of image and 3D model collections through shape and camera pose variations. *ACM Trans. Graph. 34*, 6, 233:1–233:13.

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

JIA, Y., 2013. Caffe: An open source convolutional architecture for fast feature embedding.

KARYPIS, G., AND KUMAR, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing 20*, 1, 359–392.

KIM, V. G., LI, W., MITRA, N. J., DIVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. Graph. 31*, 4, 54:1–54:11.

KRÄHENBÜHL, P., AND KOLTUN, V. 2011. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 109–117.

LI, Y., SU, H., QI, C. R., FISH, N., COHEN-OR, D., AND GUIBAS, L. J. 2015. Joint embeddings of shapes and images via CNN image purification. *ACM Trans. Graph. 34*, 6, 234:1–234:12.

LING, H., AND OKADA, K. 2007. An efficient earth mover's distance algorithm for robust histogram comparison. *Trans. PAMI 29*, 5, 840–853.

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

LIU, Z., LI, X., LUO, P., LOY, C. C., AND TANG, X. 2015. Semantic image segmentation via deep parsing network. In *ICCV*, 1377–1385.

LLOYD, S. P. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2, 129–137.

LONG, J., SHELHAMER, E., AND DARRELL, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

LUO, P., WANG, X., AND TANG, X. 2012. Hierarchical face parsing via deep learning. In *CVPR*, 2480–2487.

LUO, L., LI, H., AND RUSINKIEWICZ, S. 2013. Structure-aware hair capture. *ACM Trans. Graph. 32*, 4, 76:1–76:12.

O'DONOVAN, P., LIBEKS, J., AGARWALA, A., AND HERTZMANN, A. 2014. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph. 33*, 4, 92:1–92:9.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph. 30*, 4, 33:1–33:10.

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.

RABINOVICH, A., VEDALDI, A., GALLEGUILLOS, C., WIEWIORA, E., AND BELONGIE, S. 2007. Objects in context. In *ICCV*, 1–8.

REN, S., HE, K., GIRSHICK, R., AND SUN, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

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

SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 1–15.

SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SMITH, B. M., ZHANG, L., BRANDT, J., LIN, Z., AND YANG, J. 2013. Exemplar-based face parsing. In *CVPR*, 3484–3491.

UIJLINGS, J. R., VAN DE SANDE, K. E., GEVERS, T., AND SMEULDERS, A. W. 2013. Selective search for object recognition. *IJCV 104*, 2, 154–171.

WANG, D., CHAI, X., ZHANG, H., CHANG, H., ZENG, W., AND SHAN, S. 2011. A novel coarse-to-fine hair segmentation method. In *IEEE Automatic Face Gesture Recognition and Workshops*, 233–238.

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. *TVCG 13*, 2, 213–234.

WARRELL, J., AND PRINCE, S. J. D. 2009. Labelfaces: Parsing facial features by multiclass labeling with an epitome prior. In *ICIP*, 2481–2484.

WENG, Y., WANG, L., LI, X., CHAI, M., AND ZHOU, K. 2013. Hair interpolation for portrait morphing. In *Computer Graphics Forum*, vol. 32, 79–84.

YUKSEL, C., SCHAEFER, S., AND KEYSER, J. 2009. Hair meshes. *ACM Trans. Graph. 28*, 5, 166:1–166:7.

ZHENG, S., JAYASUMANA, S., ROMERA-PAREDES, B., VINEET, V., SU, Z., DU, D., HUANG, C., AND TORR, P. H. S. 2015. Conditional random fields as recurrent neural networks. In *ICCV*, 1529–1537.