Intuitive, Interactive Beard and Hair Synthesis with Generative Models

Kyle Olszewski[13], Duygu Ceylan[2], Jun Xing[35], Jose Echevarria[2], Zhili Chen[27], Weikai Chen[36], and Hao Li[134]
[1]University of Southern California, [2]Adobe Inc., [3]USC ICT, [4]Pinscreen Inc.,
[5]miHoYo, [6]Tencent America, [7]ByteDance Research
{olszewski.kyle,duygu.ceylan,junxnui}@gmail.com echevarr@adobe.com
{iamchenzhili,chenwk891}@gmail.com hao@hao-li.com

Abstract

We present an interactive approach to synthesizing realistic variations in facial hair in images, ranging from subtle edits to existing hair to the addition of complex and challenging hair in images of clean-shaven subjects. To circumvent the tedious and computationally expensive tasks of modeling, rendering and compositing the 3D geometry of the target hairstyle using the traditional graphics pipeline, we employ a neural network pipeline that synthesizes realistic and detailed images of facial hair directly in the target image in under one second. The synthesis is controlled by simple and sparse guide strokes from the user defining the general structural and color properties of the target hairstyle. We qualitatively and quantitatively evaluate our chosen method compared to several alternative approaches. We show compelling interactive editing results with a prototype user interface that allows novice users to progressively refine the generated image to match their desired hairstyle, and demonstrate that our approach also allows for flexible and high-fidelity scalp hair synthesis.

1. Introduction

The ability to create and edit realistic facial hair in images has several important, wide-ranging applications. For example, law enforcement agencies could provide multiple images portraying how missing or wanted individuals would look if they tried to disguise their identity by growing a beard or mustache, or how such features would change over time as the subject aged. Someone considering growing or changing their current facial hair may want to pre-visualize their appearance with a variety of potential styles without making long-lasting changes to their physical appearance. Editing facial hair in pre-existing images would also allow users to enhance their appearance, for example in images used for their social media profile pictures. Insights into how to perform high-quality and controllable facial hair synthesis would also prove useful in improving face-swapping technology such as Deepfakes for subjects with complex facial hair.

Figure 1: Given a target subject image, a masked region in which to perform synthesis, and a set of strokes of varying colors provided by the user, our approach interactively synthesizes hair with the appropriate structure and appearance.

One approach would be to infer the 3D geometry and appearance of any facial hair present in the input image, then manipulate or replace it as as desired before rendering and compositing into the original image. However, single view 3D facial reconstruction is in itself an ill-posed and under constrained problem, and most state-of-the-art approaches struggle in the presence of large facial hair, and rely on parametric facial models which cannot accurately represent such structures. Furthermore, even state-of-the-art 3D hair rendering methods would struggle to provide sufficiently realistic results quickly enough to allow for interactive feedback for users exploring numerous subtle stylistic variations.

One could instead adopt a more direct and naive approach, such as copying regions of facial hair from exemplar images of a desired style into the target image. However, it would be extremely time-consuming and tedious to either find appropriate exemplars matching the position, perspective, color, and lighting conditions in the target image, or to modify these properties in the selected exemplar regions so as to assemble them into a coherent style matching both the target image and the desired hairstyle.

In this paper we propose a learning-based interactive approach to image-based hair editing and synthesis. We exploit the power of generative adversarial networks (GANs), which have shown impressive results for various image editing tasks. However, a crucial choice in our task is the input to the network guiding the synthesis process used during training and inference. This input must be sufficiently detailed to allow for synthesizing an image that corresponds to the user's desires. Furthermore, it must also be tractable to obtain training data and extract input closely corresponding to that provided by users, so as to allow for training a generative model to perform this task. Finally, to allow novice artists to use such a system, authoring this input should be intuitive, while retaining interactive performance to allow for iterative refinement based on realtime feedback.

A set of sketch-like "guide strokes" describing the local shape and color of the hair to be synthesized is a natural way to represent such input that corresponds to how humans draw images. Using straightforward techniques such as edge detection or image gradients would be an intuitive approach to automatically extract such input from training images. However, while these could roughly approximate the types of strokes that a user might provide when drawing hair, we seek to find a representation that lends itself to intuitively editing the synthesis results without explicitly erasing and replacing each individual stroke.

Consider a vector field defining the dominant local orientation across the region in which hair editing and synthesis is to be performed. This is a natural representation for complex structures such as hair, which generally consists of strands or wisps of hair with local coherence, which could easily be converted to a set of guide strokes by integrating the vector field starting from randomly sampled positions in the input image. However, this representation provides additional benefits that enable more intuitive user interaction. By extracting this vector field from the original facial hair in the region to be edited, or by creating one using a small number of coarse brush strokes, we could generate a dense set of guide strokes from this vector field that could serve as input to the network for image synthesis. Editing this vector field would allow for adjusting the overall structure of the selected hairstyle (*e.g.*, making a straight hairstyle more curly or tangled, or vice versa) with relatively little user input, while still synthesizing a large number of guide strokes corresponding to the user's input. As these strokes are used as the final input to the image synthesis networks, subtle local changes to the shape and color of the final image can be accomplished by simply editing, adding or removing individual strokes.

We carefully chose our network architectures and training techniques to allow for high-fidelity image synthesis, tractable training with appropriate input data, and interactive performance. Specifically, we propose a two-stage pipeline. While the first stage focuses on synthesizing realistic facial hair, the second stage aims to refine this initial result and generate plausible compositions of the generated hair within the input image.

The success of such a learning-based method depends on the availability of a large-scale training set that covers a wide range of facial hairstyles. To our knowledge, no such dataset exists, so we fill this void by creating a new synthetic dataset that provides variation along many axes such as the style, color, and viewpoint in a controlled manner. We also collect a smaller dataset of real facial hair images we use to allow our method to better generalize to real images. We demonstrate how our networks can be trained using these datasets to achieve realistic results despite the relatively small amount of real images used during training.

We introduce a user interface with tools that allow for intuitive creation and manipulation of the vector fields used to generate the input to our synthesis framework. We conduct comparisons to alternative approaches, as well as extensive ablations demonstrating the utility of each component of our approach. Finally, we perform a perceptual study to evaluate the realism of images authored using our approach, and a user study to evaluate the utility of our proposed user interface. These results demonstrate that our approach is indeed a powerful and intuitive approach to quickly author realistic illustrations of complex hairstyles.

2. Related Work

Texture Synthesis As a complete review of example-based texture synthesis methods is out of the scope of this paper, we refer the reader to the surveys of [65, 2] for comprehensive overviews of modern texture synthesis techniques. In terms of methodology, example-based texture synthesis approaches can be mainly categorized into pixel-based methods [66, 19], stitching-based methods [18, 40, 42], optimization-based approaches [39, 26, 68, 35] and appearance-space texture synthesis [44]. Close to our work, Lukáč *et al*. [48] present a method that allows users to paint using the visual style of an arbitrary example texture. In [47], an intuitive editing tool is developed to support example-based painting that globally follows user-specified shapes while generating interior content that preserves the textural details of the source image. This tool is not specifically designed for hair synthesis, however, and thus lacks local controls that users desire, as shown by our user study.

Recently, many researchers have attempted to leverage neural networks for texture synthesis [23, 45, 52]. However, it remains nontrivial for such techniques to accomplish simple editing operations, *e.g*. changing the local color or structure of the output, which are necessary in our scenario.

Style Transfer The recent surge of style transfer research suggests an alternate approach to replicating stylized features from an example image to a target domain [22, 33, 46, 58]. However, such techniques make it possible to handle varying styles from only one exemplar image. When considering multiple frames of images, a number of works have been proposed to extend the original technique to handle video [57, 60] and facial animations [21]. Despite the great success of such neural-based style transfer techniques, one key limitation lies in their inability to capture fine-scale texture details. Fišer *et al*. [20] present a non-parametric model that is able to reproduce such details. However, the guidance channels employed in their approach is specially tailored for stylized 3D rendering, limiting its application.

Hair Modeling Hair is a crucial component for photorealistic avatars and CG characters. In professional production, human hair is modeled and rendered with sophisticated devices and tools [11, 38, 67, 71]. We refer to [64] for an extensive survey of hair modeling techniques. In recent years, several multi-view [49, 28] and single-view [9, 8, 29, 7] hair modeling methods have been proposed. An automatic pipeline for creating a full head avatar from a single portrait image has also been proposed [30]. Despite the large body of work in hair modeling, however, techniques applicable to facial hair reconstruction remain largely unexplored. In [3], a coupled 3D reconstruction method is proposed to recover both the geometry of sparse facial hair and its underlying skin surface. More recently, Hairbrush [69] demonstrates an immersive data-driven modeling system for 3D strip-based hair and beard models.

Image Editing Interactive image editing has been extensively explored in computer graphics community over the past decades. Here, we only discuss prior works that are highly related to ours. In the seminal work of Bertalmio *et al*. [4], a novel technique is introduced to digitally inpaint missing regions using isophote lines. Pérez *et al*. [54] later propose a landmark algorithm that supports general interpolation machinery by solving Poisson equations. Patch-based approaches [18, 5, 12, 1, 13] provide a popular alternative solution by using image patches adjacent to missing context or in a dedicated source image to replace the missing regions. Recently, several techniques [31, 59, 74, 53] based on deep learning have been proposed to translate the content of a given input image to a target domain.

Closer to our work, a number of works investigate editing techniques that directly operate on semantic image attributes. Nguyen *et al*. [51] propose to edit and synthesize beards by modeling faces as a composition of multiple layers. Mohammed *et al*. [50] perform facial image editing by leveraging a parametric model learned from a large facial image database. Kemelmacher-Shlizerman [37] presents a system that enables editing the visual appearance of a target portrait photo by replicating the visual appearance from a reference image. Inspired by recent advances in deep neural networks, Brock *et al*. [6] propose a neural algorithm to make large semantic changes to natural images. This technique has inspired follow-up works which leverage deep generative networks for eye inpainting [17], semantic feature interpolation [63] and face completion [70]. The advent of generative adversarial networks (GANs) [25] has inspired a large body of high-quality image synthesis and editing approaches [10, 72, 73, 16, 61] using the power of GANs to synthesize complex and realistic images. The latest advances in sketch [55, 32] or contour [15] based facial image editing enables users to manipulate facial features via intuitive sketching interfaces or copy-pasting from exemplar images while synthesizing results plausibly corresponding to the provided input. While our system also uses guide strokes for hair editing and synthesis, we find that intuitively synthesizing realistic and varied facial hair details requires more precise control and a training dataset with sufficient examples of such facial hairstyles. Our interactive system allows for editing both the color and orientation of the hair, as well as providing additional tools to author varying styles such as sparse or dense hair. Despite the significant research in the domain of image editing, few prior works investigate high quality and intuitive synthesis of facial hair. Though Brock *et al*. [6] allows for adding or editing the overall appearance of the subject's facial hair, their results lack details and can only operate on low-resolution images. To the best of our knowledge, we present the first interactive framework that is specially tailored for synthesizing high-fidelity facial hair with large variations.

3. Overview

In Sec. 4 we describe our network pipeline (Fig. 2), the architectures of our networks, and the training process. Sec. 5 describes the datasets we use, including the large synthetic dataset we generate for the initial stage of our training process, our dataset of real facial hair images we use for the final stage of training for refinement, and our method for annotating these images with the guide strokes used as input during training (Fig. 3). Sec. 6 describes the user interface tools we provide to allow for intuitive and efficient authoring of input data describing the desired hairstyle (Fig. 4). Finally, Sec. 7 provides sample results (Fig. 5), comparisons with alternative approaches (Figs. 6 and 7), an ablation analysis of our architecture and training process (Table 1), and descriptions of the perceptual and user study we use to evaluate the quality of our results and the utility of our interface.

4. Network Architecture and Training

Given an image with a segmented region defining the area in which synthesis is to be performed, and a set of guide strokes, we use a two-stage inference process that populates the selected region of the input image with desired hairstyle

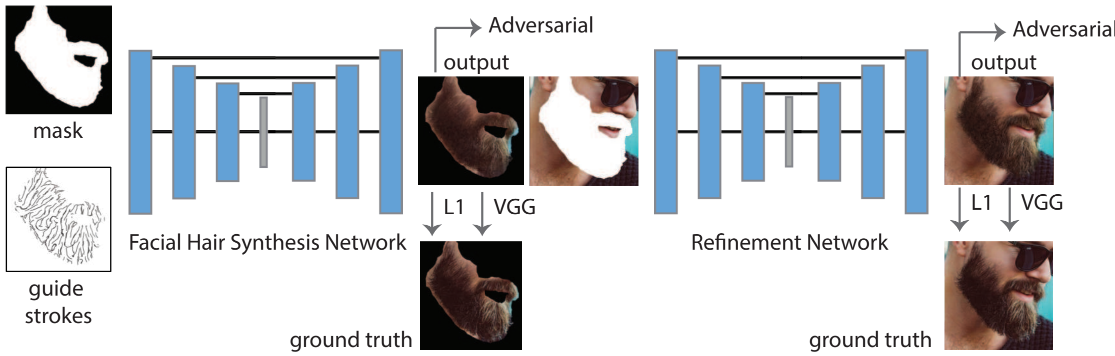

Figure 2: We propose a two-stage network architecture to synthesize realistic facial hair. Given an input image with a user-provided region of interest and sparse guide strokes defining the local color and structure of the desired hairstyle, the first stage synthesizes the hair in this region. The second stage refines and composites the synthesized hair into the input image.

as shown in Fig. 2. The first stage synthesizes an initial approximation of the content of the segmented region, while the second stage refines this initial result and adjusts it to allow for appropriate compositing into the final image.

Initial Facial Hair Synthesis. The input to the first network consists of a 1-channel segmentation map of the target region, and a 4-channel (RGBA) image of the provided guide strokes within this region. The output is a synthesized approximation of the hair in the segmented region.

The generator network is an encoder-decoder architecture extending upon the image-to-image translation network of [31]. We extend the decoder architecture with a final 3x3 convolution layer, with a step size of 1 and 1-pixel padding, to refine the final output and reduce noise. To exploit the rough spatial correspondence between the guide strokes drawn on the segmented region of the target image and the expected output, we utilize skip connections [56] to capture low-level details in the synthesized image.

We train this network using the L_1 loss between the ground-truth hair region and the synthesized output. We compute this loss only in the segmented region encouraging the network to focus its capacity on synthesizing the facial hair with no additional compositing constraints. We also employ an adversarial loss [25] by using a discriminator based on the architecture of [31]. We use a conditional discriminator, which accepts both the input image channels and the corresponding synthesized or real image. This discriminator is trained in conjunction with the generator to determine whether a given hair image is real or synthesized, and whether it plausibly corresponds to the specified input. Finally, we use a perceptual loss metric [34, 24], represented using a set of higher-level feature maps extracted from a pre-existing image classification network (*i.e.*, VGG-19 [62]). This is effective in encouraging the network to synthesize results with content that corresponds well with plausible images of real hair. The final loss $L(I_s, I_{gt})$ between the synthesized (I_s) and ground truth facial hair images (I_{gt}) is thus:

$$L_f(I_s,I_{gt}) = \omega_1 L_1(I_s,I_{gt}) + \omega_{adv} L_{adv}(I_s,I_{gt}) + \omega_{per} L_{per}(I_s,I_{gt}), \quad (1)$$

where L_1, L_{adv}, and L_{per} denote the L_1, adversarial, and perceptual losses respectively. The relative weighting of these losses is determined by ω_1, ω_{adv}, and ω_{per}. We set these weights ($\omega_1 = 50$, $\omega_{adv} = 1$, $\omega_{per} = 0.1$), such that the average gradient of each loss is at the same scale. We first train this network until convergence on the test set using our large synthetic dataset (see Sec. 5). It is then trained in conjunction with the refinement/compositing network on the smaller real image dataset to allow for better generalization to unconstrained real-world images.

Refinement and Compositing. Once the initial facial hair region is synthesized, we perform refinement and compositing into the input image. This is achieved by a second encoder-decoder network. The input to this network is the output of the initial synthesis stage, the corresponding segmentation map, and the segmented target image (the target image with the region to be synthesized covered by the segmentation mask). The output is the image with the synthesized facial hair refined and composited into it.

The architecture of the second generator and discriminator networks are identical to the first network, with only the input channel sizes adjusted accordingly. While we use the adversarial and perceptual losses in the same manner as the previous stage, we define the L_1 loss on the entire synthesized image. However, we increase the weight of this loss by a factor of 0.5 in the segmented region containing the facial hair. The boundary between the synthesized facial hair region and the rest of the image is particularly important for plausible compositions. Using erosion/dilation operations on the segmented region (with a kernel size of 10 for each operation), we compute a mask covering this boundary. We further increase the weight of the loss for these boundary region pixels by a factor of 0.5. More details on the training process can be found in the supplementary material.

5. Dataset

To train a network to synthesize realistic facial hair, we need a sufficient number of training images to represent the wide variety of existing facial hairstyles (*e.g.*, varying shape, length, density, and material and color properties), captured under varying conditions (*e.g.*, different viewpoints). We also need a method to represent the distinguishing features of these hairstyles in a simple and abstract manner that can be easily replicated by a novice user.

Figure 3: We train our network with both real (row 1, columns 1-2) and synthetic (row 1, columns 3-4) data. For each input image we have a segmentation mask denoting the facial hair region, and a set of guide strokes (row 2) that define the hair's local structure and appearance.

Data collection To capture variations across different facial hairstyles in a controlled manner, we generate a large-scale synthetic dataset using the Whiskers plugin [43] provided for the Daz 3D modeling framework [14]. This plugin provides 50 different facial hairstyles (*e.g.* full beards, moustaches, goatees), with parameters controlling the color and length of the selected hairstyle. The scripting interface provided by this modeling framework allows for programmatically adjusting the aforementioned parameters and rendering the corresponding images. By rendering the alpha mask for the depicted facial hair, we automatically extract the corresponding segmentation map. For each facial hairstyle, we synthesize it at 4 different lengths and 8 different colors. We render each hairstyle from 19 viewpoints sampled by rotating the 3D facial hair model around its central vertical axis in the range $[-90°, 90°]$ at $10°$ intervals, where $0°$ corresponds to a completely frontal view and $90°$ corresponds to a profile view (see Fig. 3, columns 3-4 for examples of these styles and viewpoints). We use the Iray [36] physically-based renderer to generate 30400 facial hair images with corresponding segmentation maps.

To ensure our trained model generalizes to real images, we collect and manually segment the facial hair region in a small dataset of such images (approximately 1300 images) from online image repositories containing a variety of styles, *e.g.* short, stubble, long, dense, curly, and straight, and large variations in illumination, pose and skin color.

Dataset Annotation Given input images with masks denoting the target region to be synthesized, we require guide strokes providing an abstract representation of the desired facial hair properties (*e.g.*, the local color and shape of the hair). We simulate guide strokes by integrating a vector field computed based on the approach of [41], which computes the dominant local orientation from the per-pixel structure tensor, then produces abstract representations of images by smoothing them using line integral convolution in the direction of minimum change. Integrating at points randomly sampled in the vector field extracted from the segmented hair region in the image produces guide lines that resemble the types of strokes specified by the users. These lines generally follow prominent wisps or strands of facial hair in the image (see Fig. 3).

6. Interactive Editing

We provide an interactive user interface with tools to perform intuitive facial hair editing and synthesis in an arbitrary input image. The user first specifies the hair region via the mask brush in the input image, then draws guide strokes within the mask abstractly describing the overall desired hairstyle. Our system provides real-time synthesized resulted after each edit to allow for iterative refinement with instant feedback. Please refer to the supplementary video for example sessions and the supplementary document for more details on our user interface. Our use of guide strokes extracted from vector fields during training enables the use of various intuitive and lightweight tools to facilitate the authoring process. In addition, the generative power of our network allows for synthesizing a rough initial approximation of the desired hairstyle with minimal user input.

Guide stroke initialization. We provide an optional initialization stage where an approximation of the desired hairstyle is generated given only the input image, segmentation mask, and a corresponding color for the masked region. This is done by adapting our training procedure to train a separate set of networks with the same architectures described in Sec. 4 using this data without the aforementioned guide strokes. Given a segmented region and the mean RGB color in this region, the network learns a form of conditional inpainting, synthesizing appropriate facial hair based on the region's size, shape, color, and the context provided by the unmasked region of the image. For example, using small masked regions with colors close to the surrounding skin tone produces sparse, short facial hair, while large regions with a color radically different from the skin tone produces longer, denser hairstyles. The resulting facial hair is realistic enough to extract an initial set of guide strokes from the generated image as is done with real images (see Sec. 5). Fig. 4 (top row) demonstrates this process.

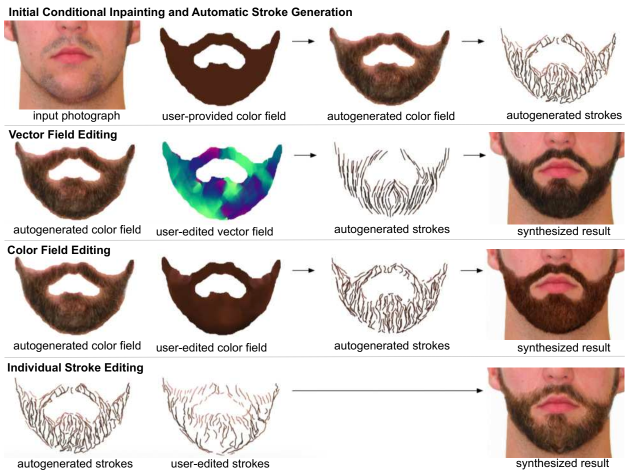

Figure 4: Editing examples. Row 1: Synthesizing an initial estimate given a user-specified mask and color. Extracting the vector field from the result allows for creating an initial set of strokes that can be then used to perform local edits. Row 2: Editing the extracted vector field to change the facial hair structure while retaining the overall color. Row 3: Changing the color field while using the vector field from the initial synthesis result allows for the creation of strokes with different colors but similar shapes to the initially generated results. Row 4: Editing the strokes extracted from the initial synthesis result allows for subtle updates, *e.g.* making the beard sparser around the upper cheeks.

Guide stroke editing These initial strokes provide a reasonable initialization the user's editing. The vector field used to compute these strokes and the initial synthesis result, which acts as the underlying color field used to compute the guide stroke color, can also be edited. As they are changed, we automatically repopulate the edited region with strokes corresponding to the specified changes. We provide brush tools to make such modifications to the color or vector fields, as seen in Fig. 4. The user can adjust the brush radius to alter the size of the region affected region, as well as the intensity used when blending with previous brush strokes. The users can also add, delete, or edit the color of guide strokes to achieve the desired alterations.

7. Results

We show various examples generated by our system in Figs. 1 and 5. It can be used to synthesize hair from scratch (Fig. 5, rows 1-2) or to edit existing facial hair (Fig. 5, rows 3-4). As shown, our system can generate facial hair of varying overall color (red vs. brown), length (trimmed vs. long), density (sparse vs. dense), and style (curly vs. straight). Row 2 depicts a complex example of a white, sparse beard on an elderly subject, created using light strokes with varying transparency. By varying these strokes and the masked region, we can generate a relatively long, mostly opaque style (column 4) or a shorter, stubbly and more translucent style (column 7). Please consult the supplementary video for live recordings of several editing sessions and timing statistics for the creation of these example results.

Perceptual study To evaluate the perceived realism of the editing results generated by our system, we conducted a perceptual study in which 11 subjects viewed 10 images of faces with only real facial hair and 10 images with facial hair manually created using our method, seen in a random order. Users observed each image for up to 10 seconds and decided whether the facial hair was real or fake/synthesized. Real images were deemed real 80% of the time, while edited images were deemed real 56% of the time. In general, facial hair synthesized with more variation in color, texture, and density were perceived as real, demonstrating the benefits of the local control tools in our interface. Overall, our system's results were perceived as generally plausible by all of the subjects, demonstrating the effectiveness of our method.

Comparison with naive copy-paste A straightforward solution to facial hair editing is to simply copy similar facial hairstyles from a reference image. While this may work for reference images depicting simple styles captured under nearly identical poses and lighting conditions to those in the the target photograph, slight disparities in these conditions result in jarring incoherence between the copied region and the underlying image. In contrast, our method allows for flexible and plausible synthesis of various styles, and enables the easy alteration of details, such as the shape and color of the style depicted in the reference photograph to allow for more variety in the final result. See Fig. 6 for some examples of copy-pasting vs. our method. Note that when copy-pasting, the total time to cut, paste, and transform (rotate and scale) the copied region to match the underlying image was in the range of 2-3 minutes, which is comparable to the amount of time spent when using our method.

Comparison with texture synthesis We compare our method to Brushables [47], which has an intuitive interface for orientation maps to synthesize images that match the target shape and orientation while maintaining the textural details of a reference image. We can use Brushables to synthesize facial hair by providing it with samples of a real facial hair image and orientation map, as shown in Fig. 7. For comparison, with our system we draw strokes in the same masked region on the face image. While Brushables synthesizes hair regions matching the provided orientation map, our results produce a more realistic hair distribution and appear appropriately volumetric in nature. Our method also handles skin tones noticeably better near hair boundaries and sparse, stubbly regions. Our method takes 1-2 seconds to process each input operation, while the optimization in Brushables takes 30-50 seconds for the same image size.

Figure 5: Example results. We show several example applications, including creating and editing facial hair on subjects with no facial hair, as well as making modifications to the overall style and color of facial hair on bearded individuals.

Figure 6: Comparison to naive copy-pasting images from reference photographs. Aside from producing more plausible results, our approach enables editing the hair's color (row 1, column 5) and shape (row 2, column 5).

Figure 7: Results of our comparison with Brushables.

	L1↓	VGG↓	MSE↓	PSNR↑	SSIM↑	FID↓
Isola *et al.* [31]	0.0304	168.5030	332.69	23.78	0.66	121.18
Single Network	0.0298	181.75	**274.88**	24.63	0.67	75.32
Ours, w/o GAN	0.0295	225.75	334.51	**24.78**	**0.70**	116.42
Ours, w/o VGG	0.0323	168.3303	370.19	23.20	0.63	67.82
Ours, w/o synth.	0.0327	234.5	413.09	23.55	0.62	91.99
Ours, only synth.	0.0547	235.6273	1747.00	16.11	0.60	278.17
Ours	**0.0275**	**119.00**	291.83	24.31	0.68	**53.15**

Table 1: Quantitative ablation analysis.

Ablation analysis As described in Sec. 4, we use a two-stage network pipeline trained with perceptual and adversarial loss, trained with both synthetic and real images. We show the importance of each component with an ablation study. With each component, our networks produce much higher quality results than the baseline network of [31].

A quantitative comparison is shown in Table 1, in which we summarize the loss values computed over our test data set using several metrics, using 100 challenging ground truth validation images not used when computing the training or testing loss. While using some naive metrics variations on our approach perform comparably well to our final approach, we note that ours outperforms all of the others

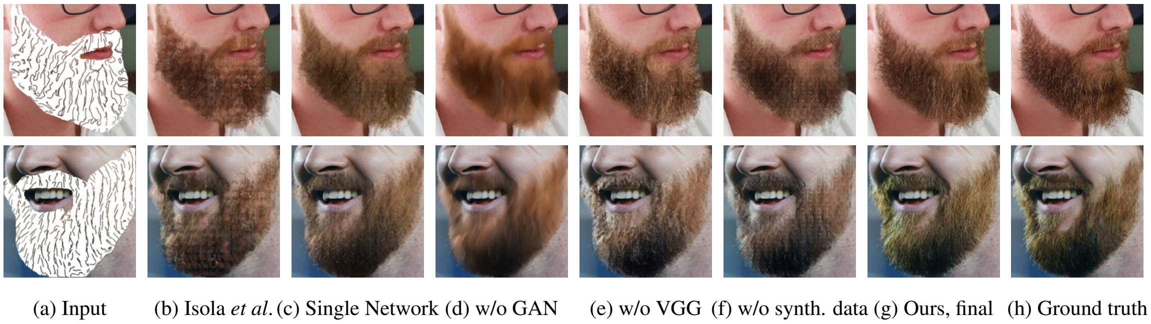

Figure 8: Qualitative comparisons for ablation study.

Figure 9: Scalp hair synthesis and compositing examples.

in terms of the Fréchet Inception Distance (FID) [27], as well as the MSE loss on the VGG features computed for the synthesized and ground truth images. This indicates that our images are perceptually closer to the actual ground truth images. Selected qualitative examples of the results of this ablation analysis can be seen in Fig. 8. More can be found in the supplementary document.

User study. We conducted a preliminary user study to evaluate the usability of our system. The study included 8 users, of which one user was a professional technical artist. The participants were given a reference portrait image and asked to create similar hair on a different clean-shaven subject via our interface. Overall, participants were able to achieve reasonable results. From the feedback, the participants found our system novel and useful. When asked what features they found most useful, some users commented that they liked the ability to create a rough approximation of the target hairstyle given only a mask and average color. Others strongly appreciated the color and vector field brushes, as these allowed them to separately change the color and structure of the initial estimate, and to change large regions of the image without drawing each individual stroke with the appropriate shape and color. Please refer to the supplementary material for the detailed results of the user study and example results created by the participants.

Application to non-facial hair. While we primarily focus on the unique challenges of synthesizing and editing facial hair in this work, our method can easily be extended to scalp hair with suitable training data. To this end, we refine our networks trained on facial hair with an additional training stage using 5320 real images with corresponding scalp hair segmentations, much in the same manner as we refine our initial network trained on synthetic data. This dataset was sufficient to obtain reasonable scalp synthesis and editing results. See Fig. 9 for scalp hair generation results. Interestingly, this still allows for the synthesis of plausible facial hair along with scalp hair within the same target image using the same trained model, given appropriately masks and guide strokes. Please consult the supplementary material for examples and further details.

Figure 10: Limitations: our method does not produce satisfactory results in some extremely challenging cases.

8. Limitations and Future Work

While we demonstrate impressive results, our approach has several limitations. As with other data-driven algorithms, our approach is limited by the amount of variation found in the training dataset. Close-up images of high-resolution complex structures fail to capture all the complexity of the hair structure, limiting the plausibility of the synthesized images. As our training datasets mostly consist of images of natural hair colors, using input with very unusual hair colors also causes noticeable artifacts. See Fig. 10 for examples of these limitations.

We demonstrate that our approach, though designed to address challenges specific to facial hair, synthesizes compelling results when applied to scalp hair given appropriate training data. It would be interesting to explore how well this approach extends to other related domains such as animal fur, or even radically different domains such as editing and synthesizing images or videos containing fluids or other materials for which vector fields might serve as an appropriate abstract representation of the desired image content.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009. 3

[2] Connelly Barnes and Fang-Lue Zhang. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, 3(1):3–20, 2017. 2

[3] Thabo Beeler, Bernd Bickel, Gioacchino Noris, Paul Beardsley, Steve Marschner, Robert W Sumner, and Markus Gross. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Transactions on Graphics (ToG)*, 31(4):117, 2012. 3

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 3

[5] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003. 3

[6] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 3

[7] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Transactions on Graphics (TOG)*, 35(4):116, 2016. 3

[8] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. Dynamic hair manipulation in images and videos. *ACM Trans. Graph.*, 32(4):75:1–75:8, July 2013. 3

[9] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, 31(4):116, 2012. 3

[10] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[11] Byoungwon Choe and Hyeong-Seok Ko. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):160–170, 2005. 3

[12] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 3

[13] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82–1, 2012. 3

[14] Daz Productions, 2017. `https://www.daz3d.com/`. 5

[15] Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T Freeman. Sparse, smart contours to represent and edit images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3511–3520, 2018. 3

[16] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 3

[17] Brian Dolhansky and Cristian Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. *arXiv preprint arXiv:1712.03999*, 2017. 3

[18] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 341–346. ACM, 2001. 2, 3

[19] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE ICCV*, pages 1033–, 1999. 2

[20] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sỳkora. Stylit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)*, 35(4):92, 2016. 3

[21] Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. Example-based synthesis of stylized facial animations. *ACM Trans. Graph.*, 36(4):155:1–155:11, July 2017. 3

[22] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 3

[23] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 262–270, Cambridge, MA, USA, 2015. MIT Press. 2

[24] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016. 4

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3, 4

[26] Jianwei Han, Kun Zhou, Li-Yi Wei, Minmin Gong, Hujun Bao, Xinming Zhang, and Baining Guo. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer*, 22(9-11):918–925, 2006. 2

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. 8

[28] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics (TOG)*, 33(4):126, 2014. 3

[29] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph.*, 34(4):125:1–125:9, July 2015. 3

[30] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single

image for real-time rendering. *ACM Transactions on Graphics (TOG)*, 36(6):195, 2017. 3

[31] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 3, 4, 7

[32] Youngjoo Jo and Jongyoul Park. SC-FEGAN: face editing generative adversarial network with user's sketch and color. *CoRR*, abs/1902.06838, 2019. 3

[33] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. 3

[34] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. 4

[35] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. In *Computer Graphics Forum*, volume 34, pages 349–359. Wiley Online Library, 2015. 2

[36] Alexander Keller, Carsten Wächter, Matthias Raab, Daniel Seibert, Dietger van Antwerpen, Johann Korndörfer, and Lutz Kettner. The iray light transport simulation and rendering system. *CoRR*, abs/1705.01263, 2017. 5

[37] Ira Kemelmacher-Shlizerman. Transfiguring portraits. *ACM Transactions on Graphics (TOG)*, 35(4):94, 2016. 3

[38] Tae-Yong Kim and Ulrich Neumann. Interactive multiresolution hair modeling and editing. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 620–629. ACM, 2002. 3

[39] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795–802, July 2005. 2

[40] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *Proc. SIGGRAPH*, SIGGRAPH '03, pages 277–286. ACM, 2003. 2

[41] Jan Eric Kyprianidis and Henry Kang. Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum*, 30(2):593–âĂŞ602, 2011. Proceedings Eurographics 2011. 5

[42] Anass Lasram and Sylvain Lefebvre. Parallel patch-based texture synthesis. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics*, pages 115–124. Eurographics Association, 2012. 2

[43] Laticis Imagery, 2017. https://www.daz3d.com/whiskers-for-genesis-3-male-s. 5

[44] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. *ACM Trans. Graph.*, 25(3):541–548, 2006. 2

[45] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016. 2

[46] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Trans. Graph.*, 36(4):120:1–120:15, July 2017. 3

[47] Michal Lukáč, Jakub Fiser, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Brushables: Example-based edge-aware directional texture painting. *Comput. Graph. Forum*, 34(7):257–267, 2015. 2, 6

[48] Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sýkora. Painting by feature: Texture boundaries for example-based image creation. *ACM Trans. Graph.*, 32(4):116:1–116:8, July 2013. 2

[49] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1490–1497. IEEE, 2012. 3

[50] Umar Mohammed, Simon JD Prince, and Jan Kautz. Visiolization: generating novel facial images. *ACM Transactions on Graphics (TOG)*, 28(3):57, 2009. 3

[51] Minh Hoai Nguyen, Jean-Francois Lalonde, Alexei A Efros, and Fernando De la Torre. Image-based shaving. In *Computer graphics forum*, volume 27, pages 627–635. Wiley Online Library, 2008. 3

[52] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. Realistic dynamic facial textures from a single image using gans. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5429–5438, 2017. 2

[53] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[54] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003. 3

[55] Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Trans. Graph.*, 37(4):99:1–99:13, July 2018. 3

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 4

[57] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. Technical report, arXiv:1604.08610, 2016. 3

[58] Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. Photorealistic facial texture inference using deep neural networks. *arXiv preprint arXiv:1612.00523*, 2016. 3

[59] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[60] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 35(4):129, 2016. 3

[61] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2107–2116, 2017. 3

[62] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[63] Paul Upchurch, Jacob Gardner, Kavita Bala, Robert Pless, Noah Snavely, and Kilian Weinberger. Deep feature interpolation for image content changes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 6090–6099, 2016. 3

[64] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R Marschner, Marie-Paule Cani, and Ming C Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE transactions on visualization and computer graphics*, 13(2):213–234, 2007. 3

[65] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117. Eurographics Association, 2009. 2

[66] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 479–488, 2000. 2

[67] Yanlin Weng, Lvdi Wang, Xiao Li, Menglei Chai, and Kun Zhou. Hair interpolation for portrait morphing. In *Computer Graphics Forum*, volume 32, pages 79–84. Wiley Online Library, 2013. 3

[68] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, March 2007. 2

[69] Jun Xing, Koki Nagano, Weikai Chen, Haotian Xu, Li-yi Wei, Yajie Zhao, Jingwan Lu, Byungmoon Kim, and Hao Li. Hairbrush for immersive data-driven hair modeling. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, 2019. 3

[70] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017. 3

[71] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Transactions on Graphics (TOG)*, 28(5):166, 2009. 3

[72] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. 3

[73] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 3

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 3