Neural Hair Rendering

Menglei Chai(✉), Jian Ren, and Sergey Tulyakov

Snap Inc., Santa Monica, USA
cmlatsim@gmail.com

Abstract. In this paper, we propose a generic neural-based hair rendering pipeline that can synthesize photo-realistic images from virtual 3D hair models. Unlike existing supervised translation methods that require model-level similarity to preserve consistent structure representation for both real images and fake renderings, our method adopts an unsupervised solution to work on arbitrary hair models. The key component of our method is a shared latent space to encode appearance-invariant structure information of both domains, which generates realistic renderings conditioned by extra appearance inputs. This is achieved by domain-specific pre-disentangled structure representation, partially shared domain encoder layers and a structure discriminator. We also propose a simple yet effective temporal conditioning method to enforce consistency for video sequence generation. We demonstrate the superiority of our method by testing it on a large number of portraits and comparing it with alternative baselines and state-of-the-art unsupervised image translation methods.

Keywords: Neural rendering · Unsupervised image translation

1 Introduction

Hair is a critical component of human subjects. Rendering virtual 3D hair models into realistic images has been long studied in computer graphics, due to the extremely complicated geometry and material of human hair. Traditional graphical rendering pipelines try to simulate every aspect of natural hair appearance, including surface shading, light scattering, semi-transparent occlusions, and soft shadowing. This is usually achieved by leveraging physics-based shading models of hair fibers, global illumination rendering algorithms, and artistically designed material parameters. Given the extreme complexity of the geometry and associated lighting effects, such a direct approximation of physical hair appearance requires a highly detailed 3D model, carefully tuned material parameters, and a huge amount of rendering computation. However, for interactive application scenarios that require efficient feedback and user-friendly interactions, such as games and photo editing softwares, it is often too expensive and unaffordable.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58523-5_22) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12363, pp. 371–388, 2020.
https://doi.org/10.1007/978-3-030-58523-5_22

With the recent advances in generative adversarial networks, it becomes natural to formulate hair rendering as a special case of the conditional image generation problem, with the hair structure controlled by the 3D model, while realistic appearance synthesized by neural networks. In the context of image-to-image translation, one of the major challenges is how to bridge both the source and target domains for proper translation. Most existing hair generation methods fall into the supervised category, which demands enough training image pairs to provide direct supervision. For example, sketch-based hair generation methods [28,34,49] construct training pairs by synthesizing user sketches from real images. While several such methods are introduced, rendering 3D hair models with the help of neural networks do not receive similar treatment. The existing work on this topic [66] requires real and fake domains considerably overlap, such that the common structure is present in both domains. This is achieved at the cost of a complicated strand-level high-quality model, which allows for extracting edge and orientation maps that serve as the common representations of hair structures between real photos and fake models. However, preparing such a high-quality hair model is itself expensive and non-trivial even for a professional artist, which significantly restricts the application scope of this method.

In this paper, we propose a generic neural-network-based hair rendering pipeline that provides efficient and realistic rendering of a generic low-quality 3D hair model borrowing the material features extracted from an arbitrary reference hair image. Instead of using a complicated strand-level model to match real-world hairs like [66], we allow users to use any type of hair model requiring only the isotropic structure of hair strands be properly represented. Particularly, we adopt sparse polygon strip meshes which are much more widely used in interactive applications [65]. Given the dramatic difference between such a coarse geometry and real hair, we are not able to design common structure representations at the model level. Therefore, supervised image translation methods will be infeasible due to the lack of paired data.

To bridge the domains of real hair images and low-quality virtual hair models in an unsupervised manner, we propose to construct a shared latent space between both real and fake domains, which encodes a common structural representation from distinct inputs of both domains and renders the realistic hair image from this latent space with the appearance conditioned by an extra reference. This is achieved by 1) different domain structure encodings used as the network inputs, to pre-disentangle geometric structure and chromatic appearance for both real hair images and 3D models; 2) a UNIT [39]-like architecture adopted to enable common latent space by partially sharing encoder weights between two auto-encoder branches that are trained with in-domain supervision; 3) a structure discriminator introduced to further match the distribution of the encoded structure features; 4) supervised reconstruction enforced on both branches to guarantee all necessary structure information is kept in the shared feature space. In addition, to enable temporally-smooth animation rendering, we introduce a simple yet effective temporal condition method with single image training data only, utilizing the exact hair model motion fields. We demonstrate the effectiveness of the pipeline and each key component by extensively testing on a large amount of diverse human portraits and various hair models. We also compare our method with general unsupervised image translation methods, and show that due to the limited sampling ability on the synthetic hair domain, all existing methods fail to produce convincing results.

2 Related Work

Image-to-image translation aims at converting images from one domain to another while keeping the structure of the source image unchanged. The literature contains various methods performing this task in different settings. Paired image-to-image translation methods [27,64] operate when pairs of images in both domains are available. For example, semantic labels to scenes [8,48,64], edges to objects [54], and image super-resolution [29,33]. However, paired data are not always available in many tasks. Unsupervised image-to-image translation tackles a setting in which paired data is not available, while sampling from two domains is possible [12,26,39,40,55,58,73]. Clearly, unpaired image-to-image translation is an ill-posed problem for there are numerous ways an image can be transformed to a different domain. Hence, recently proposed methods introduce constraints to limit the number of possible transformations. Some studies enforce certain domain properties [1,55], while other concurrent works apply cycle-consistency to transform images between different domains [31,69,73]. Our work differs from existing studies that we focus on a specific challenging problem, which is the realistic hair generation, where we want to translate manually designed hair models from the domain of rendered images to the domain of real hair. For the purpose of controllable hair generation, we leverage rendered hair structure and arbitrary hair appearance to synthesize diverse realistic hairstyles. The further difference in our work compared to the image-to-image translation papers is unbalanced data. The domain of images containing real hair is far more diverse than that of rendered hair, making it even more challenging for classical image-to-image translation works to address the problem.

Neural style transfer is related to image-to-image translation in a way that image style is changed while content is maintained [9,16,20,25,36–38,63]. Style in this case is represented by unique style of an artist [16,63] or is copied from an example image provided by the user. Our work follows the research idea from example-guide style transfer that hairstyle is obtained from reference real image. However, instead of changing the style of a whole image, our aim is to keep the appearance of the human face and background unchanged, while having full control over the hair region. Therefore, instead of following exiting works that inject style features into image generation networks directly [25,48], we propose a new architecture that combines only hair appearance features and latent features that encodes image content and adapted hair structure for image generation. This way we can achieve the goal that only the style of the hair region is manipulated according to the provided exemplar image.

Domain Adaptation addresses the domain-shift problem that widely exists between the source and target domains [53]. Various feature-based methods have been proposed to tackle the problem [13,17,18,32,62]. Recent works on adversarial learning for the embedded feature alignment between source and target domains achieve better results than previous studies [14,15,22,41,60,61]. Efforts using domain adaptation for both classification and pixel-level prediction tasks have gained significantly progress [1,10,60]. In this work, we follow the challenging setting of unsupervised domain adaptation that there is no corresponding annotation between source and target domains. We aim at learning an embedding space that only contains hair structure information for both rendered and real domains. Considering the domain gap, instead of using original images as input, we use rendered and real structure map as inputs to the encoders, which contain both domain-specific layers and shared layers, to obtain latent features. The adaptation is achieved by adversarial training and image reconstruction.

Hair Modeling, Rendering, and Generation share a similar goal with our paper, which is synthesizing photo-realistic hair images. With 3D hair models manually created [65,70], captured [19,23,42,47,71], or reconstructed from images [3–6,24,72], traditional graphical hair rendering methods focus on improving rendering quality and performance by either more accurately modeling the special hair material and lighting behaviours [11,43,44,68], or approximating certain aspects of rendering pipeline to reduce the computation complexity [45,50,52,67,74]. However, the extremely huge computation cost for realistic hair rendering usually prohibits them to be directly applied in real-time applications. Utilizing the latest advances in GANs, recent works [28,34,46,49,59] achieved impressive progress on conditioned hair image generation as supervised image-to-image translation. A GAN-based hair rendering method [66] proposes to perform conditioned 3D hair rendering by starting with a common structure representation and progressively enforce various conditions. However, it requires the hair model to be able to generate consistent representation (strand orientation map) with real images, which is challenging for low-quality mesh-based models, and cannot achieve temporally smooth results.

3 Approach

Let $\boldsymbol{h}$ be the target 3D hair model, with camera parameters $\boldsymbol{c}$ and hair material parameters $\boldsymbol{m}$, we formulate the *traditional graphic rendering pipeline* as $\mathrm{R}_t(\boldsymbol{h}, \boldsymbol{m}, \boldsymbol{c})$. Likewise, our *neural network-based rendering pipeline* is defined as $\mathrm{R}_n(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{c})$, with a low-quality hair model $\boldsymbol{h}$ and material features extracted from an arbitrary reference hair image $\boldsymbol{r}$.

3.1 Overview of Network Architecture

The overall system pipeline is shown in Fig. 1, which consists of two parallel branches for both domains of real photo (i.e., *real*) and synthetic renderings (i.e., *fake*), respectively.

On the encoding side, the *structure adaptation subnetwork*, which includes a real encoder E_r and a fake encoder E_f, achieves cross-domain structure embedding $\boldsymbol{e}$. Similar to UNIT [39], we share the weights of the last few ResNet layers

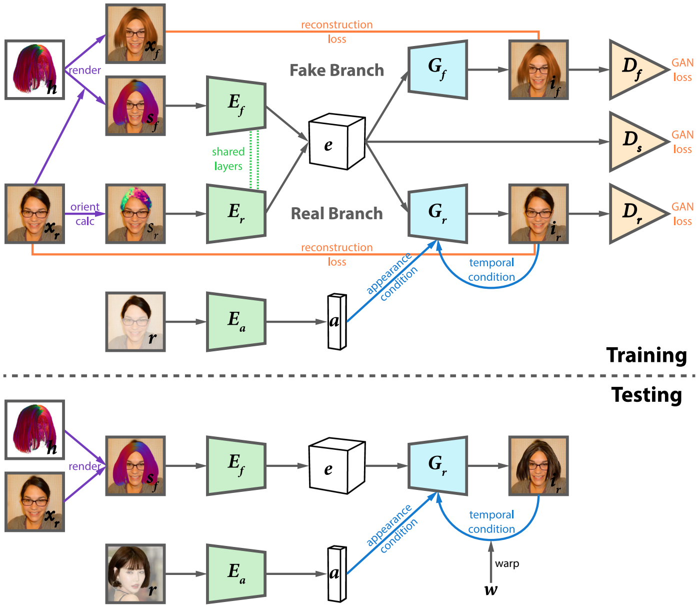

Fig. 1. The overall pipeline of our neural hair rendering framework. We use two branches to encode hair structure features, one for the real domain and the other for the fake domain. A domain discriminator is applied to the outputs from both encoders, to achieve domain invariant features. We also use two decoders to reconstruct images for two domains. The decoder in the real domain is different from the one in the fake domain, for it is conditioned on a reference image. Additionally, to generate consistent videos, we apply a temporal condition on the real branch. During inference, we use the encoder in the fake branch to get hair structure features from a 3D hair model and use the generator in the real branch to synthesized an appearance conditioned image.

in E_r and E_f to extract consistent structural representation from two domains. In addition, a structure discriminator D_s is introduced to match the high-level feature distributions between two domains to enforce the shared latent space further to be domain invariant.

On the decoding side, the *appearance rendering subnetwork*, consisting of G_r and G_f for the real and fake domain respectively, is attached after the shared latent space $\boldsymbol{e}$ to reconstruct the images in the corresponding domain. Each decoder owns its exclusive domain discriminator D_r and D_f to ensure the reconstruction matches the domain distribution, besides the reconstruction losses. The hair appearance is conditioned in an asymmetric way that G_r accepts the extra condition of material features extracted from a reference image $\boldsymbol{r}$ by using material encoder E_m, while the unconditional decoder G_f is asked to memorize the appearance, which is made on purpose for training data generation (Sect. 4.1).

At the *training stage*, all these networks are jointly trained using two sets of image pairs $(\boldsymbol{s}, \boldsymbol{x})$ for both real and fake domains, where $\boldsymbol{s}$ represents a domain-specific structure representation of the corresponding hair image $\boldsymbol{x}$ in this domain. Both real and fake branches try to reconstruct the image $G(E(\boldsymbol{x}))$ from its paired structure image $\boldsymbol{s}$ independently through their own encoder-decoder networks, while the shared structural features are enforced to match each other consistently by the structure discriminator D_s. We set the appearance reference $\boldsymbol{r} = \boldsymbol{x}$ in the real branch to fully reconstruct $\boldsymbol{x}$ in a paired manner.

At the *inference stage*, only the fake branch encoder E_f and the real branch decoder G_r are activated. G_r generates the final realistic rendering using structural features encoded by E_f on the hair model. The final rendering equation R_n can be formulated as:

$$\mathrm{R}_n(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{c}) = G_r(E_f(\mathrm{S}_f(\boldsymbol{h}, \boldsymbol{c})), E_m(\boldsymbol{r})), \tag{1}$$

where the function $\mathrm{S}_f(\boldsymbol{h}, \boldsymbol{c})$ renders the structure encoded image $\boldsymbol{s}_f$ of the model $\boldsymbol{h}$ in camera setting $\boldsymbol{c}$.

3.2 Structure Adaptation

The goal of the structure adaptation subnetwork, formed by the encoding parts of both branches, is to encode cross-domain structural features to support final rendering. Since the inputs to both encoders are manually disentangled structure representation (Sect. 4.1), the encoded features $E(\boldsymbol{s})$ only contain structural information of the target hair. Moreover, as the appearance information is either conditioned by extra decoder input in a way that non-spatial-varying structural information is leaked (the real branch) or simple enough to be memorized by the decoder (the fake branch) (Sect. 3.3), the encoded features should also include all the structural information necessary to reconstruct $\boldsymbol{x}$.

E_r and E_f share a similar network structure: five downsampling convolution layers followed by six ResBlks. The last two ResBlks are weight-sharing to enforce the shared latent space. D_s follows PatchGAN [27] to distinguish between the latent feature maps from both domains:

$$\mathcal{L}_{D_s} = \mathbb{E}_{\boldsymbol{s}_r}[\log(D_s(E_r(\boldsymbol{s}_r)))] + \mathbb{E}_{\boldsymbol{s}_f}[\log(1 - D_s(E_f(\boldsymbol{s}_f)))]. \tag{2}$$

3.3 Appearance Rendering

The hair appearance rendering subnetwork decodes the shared cross-domain hair features into the real domain images. The decoders G_r and G_f have different network structures and do not share weights since the neural hair rendering is a unidirectional translation that aims to map the rendered 3D model in the fake domain to real images in the real domain. Therefore, G_f is required to make sure the latent features $\boldsymbol{e}$ encode all necessary information from the input 3D model, instead of learning to render various appearance. On the other hand, G_r is designed in a way to accept arbitrary inputs for realistic image generation.

Specifically, the unconditional decoder G_f starts with two ResBlks, and then five consecutive upsampling transposed convolutional layers followed by one final convolutional layer. G_r adopts a similar structure as G_f, with each transposed convolutional layer replaced with a SPADE [48] ResBlk to use appearance feature maps $\boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}$ at different scales to condition the generation. Assuming the binary hair mask of the reference and the target images are $\boldsymbol{m}_{\boldsymbol{r}}$ and $\boldsymbol{m}_{\boldsymbol{s}}$, the appearance encoder E_m extracts the appearance feature vector on $\boldsymbol{r} \times \boldsymbol{m}_{\boldsymbol{r}}$ with five downsampling convolutional layers and an average pooling. This feature vector $E_m(\boldsymbol{r})$ is then used to construct the feature map $\boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}$ by duplicating it spatially in the target hair mask $\boldsymbol{m}_{\boldsymbol{s}}$ as follows:

$$\boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}(p) = \begin{cases} E_m(\boldsymbol{r}), & \text{if } \boldsymbol{m}_{\boldsymbol{s}_r}(p) = 1, \\ 0, & \text{if } \boldsymbol{m}_{\boldsymbol{s}_r}(p) = 0. \end{cases} \tag{3}$$

To make sure the reconstructed real image $G_r(E_r(\boldsymbol{s}_r), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r})$ and the reconstructed fake image $G_f(E_f(\boldsymbol{s}_f))$ belong to their respective distributions, we apply two domain specific discriminator D_r and D_f for the real and fake domain respectively. The adversarial losses write as:

$$\mathcal{L}_{D_r} = \mathbb{E}_{\boldsymbol{x}_r}[\log(D_r(\boldsymbol{x}_r))] + \mathbb{E}_{\boldsymbol{s}_r,\boldsymbol{r}}[\log(1 - D_r(G_r(E_r(\boldsymbol{s}_r), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r})))], \tag{4}$$

$$\mathcal{L}_{D_f} = \mathbb{E}_{\boldsymbol{x}_f}[\log(D_f(\boldsymbol{x}_f))] + \mathbb{E}_{\boldsymbol{s}_f}[\log(1 - D_f(G_f(E_f(\boldsymbol{s}_f))))]. \tag{5}$$

We also adopt perceptual losses to measure high-level feature distance utilizing the paired data:

$$\begin{aligned} \mathcal{L}_p = \sum_{l=0}^{L} & \|\boldsymbol{\Psi}_l(G_r(E_r(\boldsymbol{s}_r), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r})) - \boldsymbol{\Psi}_l(\boldsymbol{x}_r)\|_1 \\ & + \|\boldsymbol{\Psi}_l(G_f(E_f(\boldsymbol{s}_f))) - \boldsymbol{\Psi}_l(\boldsymbol{x}_f)\|_1, \end{aligned} \tag{6}$$

where $\boldsymbol{\Psi}_l(\boldsymbol{i})$ computes the activation feature map of input image $\boldsymbol{i}$ at the lth selected layer of VGG-19 [56] pre-trained on ImageNet [51].

Finally, we have the overall training objective as:

$$\min_{E,G} \max_{D} \; (\lambda_s \mathcal{L}_{D_s} + \lambda_g(\mathcal{L}_{D_r} + \mathcal{L}_{D_f}) + \lambda_p \mathcal{L}_p). \tag{7}$$

3.4 Temporal Conditioning

The aforementioned rendering network is able to generate plausible single-frame results. However, despite the hair structure is controlled by smoothly-varying inputs of $\boldsymbol{s}_f$ with the appearance conditioned by a fixed feature map $\boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}$, the spatially-varying appearance details are still generated in a somewhat arbitrary manner which tends to flicker in time (Fig. 5). Fortunately, with the availability of the 3D model, we can calculate the exact hair motion flow $\boldsymbol{w}^t$ for each pair of frames $t-1$ and t, which can be used to warp image $\boldsymbol{i}$ from $t-1$ to t as $\mathrm{W}(\boldsymbol{i}, \boldsymbol{w}^t)$. We utilize this dense correspondences to enforce temporal smoothness.

Let $\boldsymbol{I} = \{\boldsymbol{i}^0, \boldsymbol{i}^1, \ldots, \boldsymbol{i}^T\}$ be the generated result sequence, we achieve this temporal conditioning by simply using the warped result of the previous frame

$\mathrm{W}(\boldsymbol{i}^{t-1}, \boldsymbol{w}^t)$ as an additional condition, stacked with the appearance feature map $\boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}$, to the real branch decoder G_r when generating the current frame $\boldsymbol{i}^t$.

We achieve temporally consistent by changing the real branch decoder only with temporally finetuning. During temporal training, we fix all other networks and use the same objective as Eq. 7, but randomly (50% of chance) concatenate $\boldsymbol{x}_r$ into the condition inputs to the SPADE ResBlks of G_r^t. The generation pipeline of the real branch now becomes $G_r^t(E_r(\boldsymbol{s}_r), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_r}, \boldsymbol{x}_r)$, so that the network learns to preserve the consistency if the previous frame is inputted as the temporal condition, or generate randomly from scratch if the condition is zero.

Finally, we have the rendering equation for sequential generation:

$$\boldsymbol{i}^t = \mathrm{R}_n(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{c}^t) = \begin{cases} G_r(E_f(\boldsymbol{s}_f^t), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_f^t}), & \text{if } t = 0, \\ G_r^t(E_f(\boldsymbol{s}_f^t), \boldsymbol{a}_{\boldsymbol{r},\boldsymbol{s}_f^t}, \mathrm{W}(\boldsymbol{i}^{t-1}, \boldsymbol{w}^t)). & \text{if } t > 0, \end{cases} \tag{8}$$
$$\boldsymbol{s}_f^t = \mathrm{S}_f(\boldsymbol{h}, \boldsymbol{c}^t).$$

4 Experiments

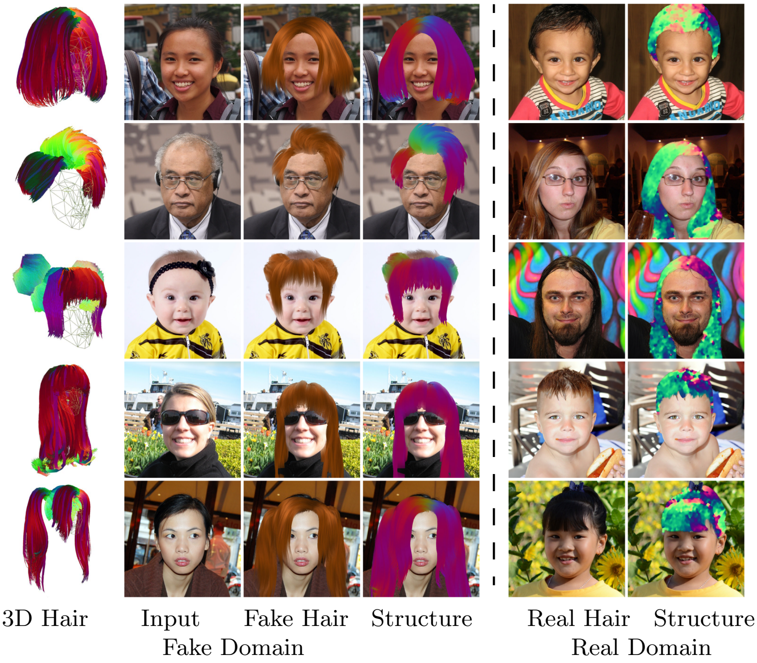

Fig. 2. Training data preparation. For the fake domain (left), we use hair model and input image to generate fake rendering and model structure map. For the real domain (b), we generate image structure map for each image.

4.1 Data Preparation

To train the proposed framework, we generate a dataset that includes image pairs $(\boldsymbol{s}, \boldsymbol{x})$ for both real and fake domains. In each domain, $\boldsymbol{s} \rightarrow \boldsymbol{x}$ indicates the mapping from structure to image, where $\boldsymbol{s}$ encodes only the structure information, and $\boldsymbol{x}$ is the corresponded image that conforms to the structure condition.

Real Domain. We adopt the widely used FFHQ [30] portrait dataset to generate the training pairs for the real branch, given it contains diverse hairstyles on shapes and appearances. To prepare real data pairs, we use original portrait photos from FFHQ as $\boldsymbol{x}_r$, and generate $\boldsymbol{s}_r$ to encode only structure information from hair. However, obtaining $\boldsymbol{s}_r$ is a non-trivial process since hair image also contains material information, besides structural knowledge. To fully disentangle structure and material, and construct a universal structural representation $\boldsymbol{s}$ of all real hair, we apply a dense pixel-level orientation map in the hair region, which is formulated as $\boldsymbol{s}_r = \mathrm{S}_r(\boldsymbol{x}_r)$, calculated with oriented filter kernels [47]. Thus, we can obtain $\boldsymbol{s}_r$ that only consists of local hair strand flow structures. Example generated pairs are presented in Fig. 2b.

For the purpose of training and validation, we randomly select $65,000$ images from FFHQ as training, and use the remaining $5,000$ images for testing. For each image $\boldsymbol{x}_r$, we perform hair segmentation using off-the-shelf model [4], and calculate $\boldsymbol{s}_r$ for the hair region.

Fake Domain. There are multiple ways to model and render virtual hair models. From coarse to fine, typical virtual hair models range from a single rigid shape, coarse polygon strips representing detached hair wisps, to a large number of thin hair fibers that mimic real-world hair behaviors. Due to various granularity of the geometry, the structural representation is hardly shared with each other or real hair images. In our experiments, all the hair models we used are polygon strips based considering this type of hair model is widely adopted in real-time scenarios for it is efficient to render and flexible to be animated. To generate $\boldsymbol{s}_f$ for a given hair model $\boldsymbol{h}$ and specified camera parameters $\boldsymbol{c}$, we use smoothly varying color gradient as texture to render $\boldsymbol{h}$ into a color image that embeds the structure information of the hair geometry, such that $\boldsymbol{s}_f = \mathrm{S}_f(\boldsymbol{h}, \boldsymbol{c})$. As for $\boldsymbol{x}_f$, we use traditional graphic rendering pipeline to render $\boldsymbol{h}$ with a uniform appearance color and simple diffuse shading, so that the final synthetic renderings have a consistent appearance that can be easily disentangled without any extra condition, and keep all necessary structural information to verify the effectiveness of the encoding step. Example pairs are shown in Fig. 2a.

For the 3D hair used for fake data pairs, we create five models (leftmost column in Fig. 2). The first four models are used for training, and the last one is used to evaluate the generalization capability of the network, for the network has never seen it. All these models consist of 10 to 50 polygon strips, which is sparse enough for real-time applications. We use the same training set from the real domain to form training pairs. Each image is overlaid by one of the four 3D hair models according to the head position and pose. Then the image with the fake hair model is used to generate $\boldsymbol{x}_f$ through rendering the hair model with

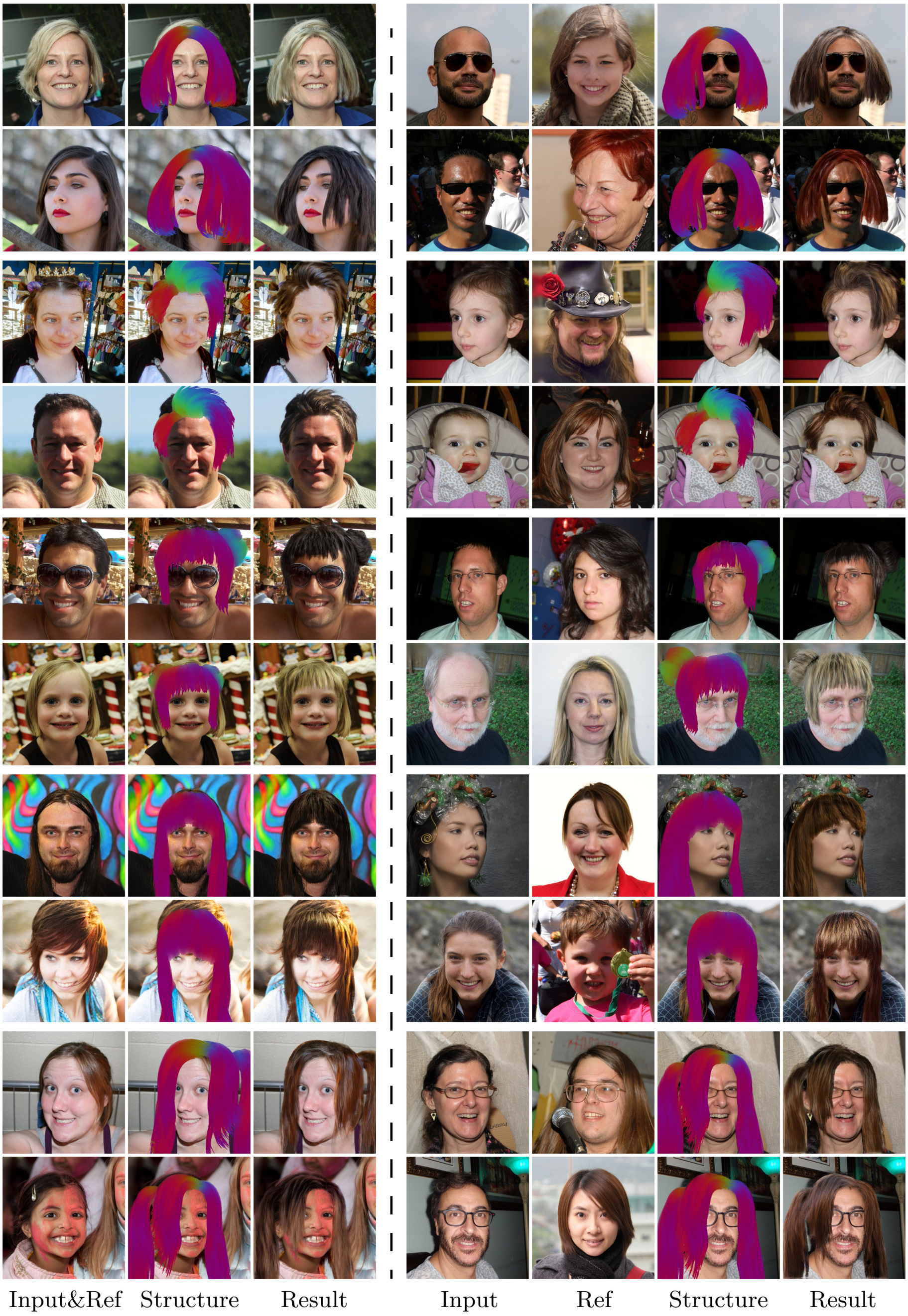

Fig. 3. Results for the hair models used in this study (2 rows per model). We visualize examples where the input and the reference image are the same (left), and the input and the reference are different images (right). In the former case the method copies appearance from another image.

simple diffuse shading, and $\boldsymbol{s}_f$ by exporting color textures that encode surface tangent of the mesh. We strictly use the same shading parameters, including lighting and color, to enforce a uniform appearance of hair that can be easily disentangled by the networks.

4.2 Implementation Details

We apply a two-stage learning strategy. During the first stage, all networks are trained jointly following Eq. 7 for the single-image renderer R_n. After that, we temporally fine-tune the decoder G_r of the real branch, to achieve temporally-smooth renderer R_n^t, by introducing the additional temporal condition as detailed in Sect. 3.4. To make the networks of both stages consistent, we keep the same condition input dimensions, including appearance and temporal, but set the temporal condition to zero during the first stage. During the second stage, we set it to zero with 50% of chance. The network architecture discussed in Sect. 3 is implemented using PyTorch. We adopt Adam solver with a learning rate set to 0.0001 for the first stage, and 0.00001 for the fine-tuning stage. The training resolution of all images is 512 × 512, with the mini-batch size set to 4. For the loss functions, weights λ_p, λ_s, and λ_g are set to 10, 1, and 1, respectively. All experiments are conducted on a workstation with 4 Nvidia Tesla P100 GPUs. During test time, rendering a single frame takes less than 1 second, with structure encoding less than 200 ms and final generation less than 400 ms.

4.3 Qualitative Results

We present visual hair rendering results from two settings in Fig. 3. The left three columns in Fig. 3 show that the reference image $\boldsymbol{r}$ is the same as $\boldsymbol{x}_r$. By applying a hair model, we can modify human hair shape but keep the original hair appearance and orientation. The right four columns show that the reference image is different from $\boldsymbol{x}_r$, therefore, both structure and appearance of hair from $\boldsymbol{x}_r$ can be changed at the same time to render the hair with a new style. We also demonstrate our video results in Fig. 5 (please click the image to watch video results online), where we adopt 3D face tracking [2] to guide the rigid position of the hair model, and physics-based hair simulation method [7] to generate secondary hair motion. These flexible applications demonstrate that our method can be easily applied to modify hair and generate novel high-quality hair images.

4.4 Comparison Results

To the best of our knowledge, there is no previous work that tackles the problem of neural hair rendering; thus, a direct comparison is not feasible. However, in light of our methods aim to bridge two different domains without ground-truth image pairs, which is related to unsupervised image translation, we compare our network with state-of-the-art unpaired image translation studies. It is important

Table 1. Quantitative comparison results. We compare our method against commonly adopted image-to-image translation frameworks, reporting Fréchet Inception Distance (FID, lower the better), Intersection over Union (IoU, higher the better) and pixel accuracy (Accuracy, higher the better). Additionally we report ablation studies by first removing the structural discriminator (w/o SD) followed by removing both the structural discriminator and the shared latent space (w/o SL and SD).

Method	FID ↓	IoU(%) ↑	Accuracy(%) ↑
Graphic Renderer	98.62	55.77	86.17
CycleGAN [73]	107.11	46.46	84.06
UNIT [39]	116.79	30.89	84.27
DRIT [35]	174.39	30.69	65.80
w/o SL and SD	94.25	80.10	93.89
w/o SD	77.09	86.60	96.35
Ours	**57.08**	**86.74**	**96.45**

to stress that although our hair rendering translation falls into the range of image translation problems, there exist fundamental differences compared to the generic unpaired image translation formulations for the following two reasons.

First and foremost, compared with translation between two domains, such as painting styles, or seasons/times of the day, which have roughly the same amount of images for two domains and enough representative training images can be sampled to provide nearly-uniform domain coverage, our real/fake domains have dramatically different sizes–it is easy to collect a huge amount of real human portrait photos with diverse hairstyles to form the real domain. Unfortunately, for the fake domain, it is impossible to reach the same variety since it would require manually designing *every* possible hair shape and appearance to describe the distribution of the whole domain of rendered fake hair. Therefore, we focus on a realistic assumption that only a limited set of such models are available for training and testing, such that we use four 3D models for training and one for testing, which is far from being able to produce variety in the fake domain.

Second, as a deterministic process, hair rendering should be conditioned strictly on both geometric shape and chromatic appearance, which can be hardly achieved with unconditioned image translation frameworks.

With those differences bearing in mind, we show the comparison between our method and three unpaired image translation studies, including CycleGAN [73], DRIT [35], and UNIT [39]. For the training of these methods, we use the same sets of images, $\boldsymbol{x}_r$ and $\boldsymbol{x}_f$, for both real and fake domains, and the default hyperparameters reported by the original papers. Additionally, we compare with the images generated by the traditional graphic rendering pipeline. We denote the method as **Graphic Renderer**. Finally, we report two ablation studies to evaluate the soundness of the network and the importance of each step: 1) we first remove the structural discriminator (termed as **w/o SD**); 2) we then additionally remove the shared latent space (termed as **w/o SL and SD**).

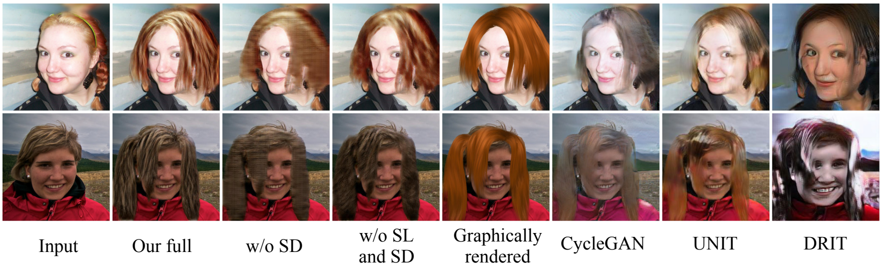

Fig. 4. Visual comparisons. We show selected visual comparisons against commonly adopted image-to-image translation methods as well as visualize ablation results. Our method synthesizes more realistic hair images compared to other approaches.

Quantitative Comparison. For quantitative evaluation, we adopt FID (Fréchet Inception Distance) [21] to measure the distribution distance between two domains. Moreover, inspired by the evaluation protocol from existing work [8,64], we apply a pre-trained hair segmentation model [57] on the generated images to get the hair mask, and compare it with the ground truth. Intuitively, the segmentation model should predict the hair mask that similar to the ground-truth for the realistic synthesized images. To measure the segmentation accuracy, we use both Intersection-over-Union (IoU) and pixel accuracy (Accuracy).

The quantitative results are reported in Table 1. Our method significantly outperforms the state-of-the-art unpaired image translation works and graphic rendering approach by a large margin for all three evaluation metrics. The low FID score proves our method can generate high-fidelity hair images that contain similar hair appearance distribution as images from the real domain. The high IoU and Accuracy demonstrate the ability of the network to minimize the structure gap between real and fake domains so that the synthesized images can follow the manually designed structure. Furthermore, the ablation analysis in Table 1 shows both shared encoder layers and the structural discriminator are essential parts of the network, for the shared encoder layers help the network to find a common latent space that embeds hair structural knowledge, while the structural discriminator forces the hair structure features to be domain invariant.

Qualitative Comparison. The qualitative comparison of different methods is shown in Fig. 4. It can be easily seen that our generated images have much higher quality than the synthesized images created by other state-of-the-art unpaired image translation methods, for they have clearer hair mask, follow hair appearance from reference images, maintain the structure from hair models, and look like natural hair. Compared with the ablation methods (Fig. 4c and d), our full method (Fig. 4b) can follow the appearance from reference images (Fig. 4a) by generating hair with similar orientation.

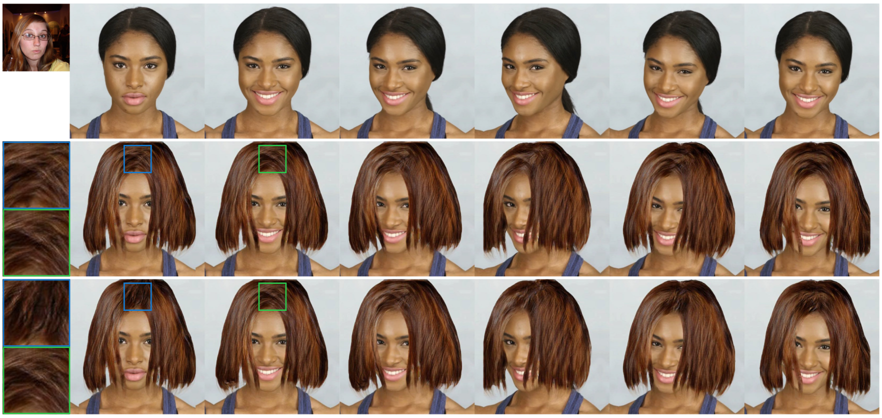

Fig. 5. Video results and comparisons. Top row: the first image is the appearance reference image and others are continuous input frames; middle row: generated hair images with temporal conditioning; bottom row: generated hair images without temporal conditioning. We show two zoom-in hair regions for each result. By applying temporal conditioning, our model synthesizes hair images with consistent appearance, while not using temporal conditioning leads to hair appearance flickering as indicated by blue and green boxes. ***Click the image to play the video results and comparisons.*** (Color figure online)

We also show the importance of temporal conditioning (Sect. 3.4) in Fig. 5. The temporal conditioning helps us generate consistent and smooth video results, for hair appearance and orientation are similar between continuous frames. Without temporal conditioning, the hair texture could be different between frames, as indicated by blue and green boxes, which may result in flickering for the synthesized video. Please refer to the supplementary video for more examples.

5 Conclusions

We propose a neural-based rendering pipeline for general virtual 3D hair models. The key idea of our method is that instead of enforcing model-level representation consistency to enable supervised paired training, we relax the strict requirements on the model and adopt an unsupervised image translation framework. To bridge the gap between real and fake domains, we construct a shared latent space to encode a common structure feature space for both domains, even if their inputs are dramatically different. In this way, we can encode a virtual hair model into such a structure feature, and switch it into the real generator to produce realistic rendering. The conditional real generator not only allows flexible appearance conditioning but can also be used to introduce temporal conditioning to generate smooth sequential results.

Our method has several limitations. First, the current method does not change the input. A smaller fake hair won't be able to fully occlude the original one in the input image. It is possible to do face inpainting to remove the excessive hair regions to fix this issue. Second, when the lighting/material of the appearance reference is dramatically different from the input, the result may look unnatural. Better reference selection would help to make the results better. Third, the current method simply blends the generated hair onto the input, which causes blending artifacts in some results especially when the background is complicated. A simple solution is to train a supervised boundary refinement network to achieve better blending quality.

References

1. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR, pp. 95–104 (2017)
2. Cao, C., Chai, M., Woodford, O.J., Luo, L.: Stabilized real-time face tracking via a learned dynamic rigidity prior. ACM Trans. Graph. **37**(6), 233:1–233:11 (2018)
3. Chai, M., Luo, L., Sunkavalli, K., Carr, N., Hadap, S., Zhou, K.: High-quality hair modeling from a single portrait photo. ACM Trans. Graph. **34**(6), 204:1–204:10 (2015)
4. Chai, M., Shao, T., Wu, H., Weng, Y., Zhou, K.: AutoHair: fully automatic hair modeling from a single image. ACM Trans. Graph. **35**(4), 116:1–116:12 (2016)
5. Chai, M., Wang, L., Weng, Y., Jin, X., Zhou, K.: Dynamic hair manipulation in images and videos. ACM Trans. Graph. **32**(4), 75:1–75:8 (2013)
6. Chai, M., Wang, L., Weng, Y., Yu, Y., Guo, B., Zhou, K.: Single-view hair modeling for portrait manipulation. ACM Trans. Graph. **31**(4), 116:1–116:8 (2012)
7. Chai, M., Zheng, C., Zhou, K.: A reduced model for interactive hairs. ACM Trans. Graph. **33**(4), 124:1–124:11 (2014)
8. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV, pp. 1520–1529 (2017)
9. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. CoRR abs/1612.04337 (2016)
10. Chen, Y., Chen, W., Chen, Y., Tsai, B., Wang, Y.F., Sun, M.: No more discrimination: cross city adaptation of road scene segmenters. In: ICCV, pp. 2011–2020 (2017)
11. d'Eon, E., François, G., Hill, M., Letteri, J., Aubry, J.: An energy-conserving hair reflectance model. Comput. Graph. Forum **30**(4), 1181–1187 (2011)
12. Dundar, A., Liu, M., Wang, T., Zedlewski, J., Kautz, J.: Domain stylization: a strong, simple baseline for synthetic to real image domain adaptation. CoRR abs/1807.09384 (2018)
13. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV, pp. 2960–2967 (2013)
14. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: ICML, vol. 37, pp. 1180–1189 (2015)
15. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 59:1–59:35 (2016)
16. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR, pp. 2414–2423 (2016)
17. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR, pp. 2066–2073 (2012)

18. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: an unsupervised approach. In: ICCV, pp. 999–1006 (2011)
19. Herrera, T.L., Zinke, A., Weber, A.: Lighting hair from the inside: a thermal approach to hair reconstruction. ACM Trans. Graph. **31**(6), 146:1–146:9 (2012)
20. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.: Image analogies. In: SIGGRAPH, pp. 327–340 (2001)
21. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS, pp. 6626–6637 (2017)
22. Hoffman, J., et al.: CyCADA: cycle-consistent adversarial domain adaptation. ICML, vol. 80, pp. 1994–2003 (2018)
23. Hu, L., Ma, C., Luo, L., Li, H.: Robust hair capture using simulated examples. ACM Trans. Graph. **33**(4), 126:1–126:10 (2014)
24. Hu, L., Ma, C., Luo, L., Li, H.: Single-view hair modeling using a hairstyle database. ACM Trans. Graph. **34**(4), 125:1–125:9 (2015)
25. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV, pp. 1510–1519 (2017)
26. Huang, X., Liu, M., Belongie, S.J., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV, vol. 11207, pp. 179–196 (2018)
27. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR, pp. 5967–5976 (2017)
28. Jo, Y., Park, J.: SC-FEGAN: face editing generative adversarial network with user's sketch and color. In: ICCV, pp. 1745–1753 (2019)
29. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
30. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR, pp. 4401–4410 (2019)
31. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: ICML, vol. 70, pp. 1857–1865 (2017)
32. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: CVPR, pp. 1785–1792 (2011)
33. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR, pp. 105–114 (2017)
34. Lee, C., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. CoRR abs/1907.11922 (2019)
35. Lee, H., Tseng, H., Huang, J., Singh, M., Yang, M.: Diverse image-to-image translation via disentangled representations. ECCV, vol. 11205, pp. 36–52 (2018)
36. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 702–716. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_43
37. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. In: IJCAI, pp. 2230–2236 (2017)
38. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.: Diversified texture synthesis with feed-forward networks. In: CVPR, pp. 266–274 (2017)
39. Liu, M., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NeurIPS, pp. 700–708 (2017)

40. Liu, M., et al.: Few-shot unsupervised image-to-image translation. In: ICCV, pp. 10550–10559 (2019)
41. Liu, M., Tuzel, O.: Coupled generative adversarial networks. In: NIPS, pp. 469–477 (2016)
42. Luo, L., Li, H., Rusinkiewicz, S.: Structure-aware hair capture. ACM Trans. Graph. **32**(4), 76:1–76:12 (2013)
43. Marschner, S.R., Jensen, H.W., Cammarano, M., Worley, S., Hanrahan, P.: Light scattering from human hair fibers. ACM Trans. Graph. **22**(3), 780–791 (2003)
44. Moon, J.T., Marschner, S.R.: Simulating multiple scattering in hair using a photon mapping approach. ACM Trans. Graph. **25**(3), 1067–1074 (2006)
45. Moon, J.T., Walter, B., Marschner, S.: Efficient multiple scattering in hair using spherical harmonics. ACM Trans. Graph. **27**(3), 31 (2008)
46. Olszewski, K., et al.: Intuitive, interactive beard and hair synthesis with generative models. In: CVPR, pp. 7446–7456 (2020)
47. Paris, S., et al.: Hair photobooth: geometric and photometric acquisition of real hairstyles. ACM Trans. Graph. **27**(3), 30 (2008)
48. Park, T., Liu, M., Wang, T., Zhu, J.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR, pp. 2337–2346 (2019)
49. Qiu, H., Wang, C., Zhu, H., Zhu, X., Gu, J., Han, X.: Two-phase hair image synthesis by self-enhancing generative model. Comput. Graph. Forum **38**(7), 403–412 (2019)
50. Ren, Z., Zhou, K., Li, T., Hua, W., Guo, B.: Interactive hair rendering under environment lighting. ACM Trans. Graph. **29**(4), 55:1–55:8 (2010)
51. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
52. Sadeghi, I., Pritchett, H., Jensen, H.W., Tamstorf, R.: An artist friendly hair shading system. ACM Trans. Graph. **29**(4), 56:1–56:10 (2010)
53. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
54. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: controlling deep image synthesis with sketch and color. In: CVPR, pp. 6836–6845 (2017)
55. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: CVPR, pp. 2242–2251 (2017)
56. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
57. Svanera, M., Muhammad, U.R., Leonardi, R., Benini, S.: Figaro, hair detection and segmentation in the wild. In: ICIP, pp. 933–937 (2016)
58. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: ICLR (2017)
59. Tan, Z., et al.: MichiGAN: multi-input-conditioned hair image generation for portrait editing. ACM Trans. Graph. **39**(4), 95:1–95:13 (2020)
60. Tsai, Y., Hung, W., Schulter, S., Sohn, K., Yang, M., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: CVPR, pp. 7472–7481 (2018)
61. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR, pp. 2962–2971 (2017)
62. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR abs/1412.3474 (2014)

63. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: feed-forward synthesis of textures and stylized images. In: ICML, vol. 48, pp. 1349–1357 (2016)
64. Wang, T., Liu, M., Zhu, J., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR, pp. 8798–8807 (2018)
65. Ward, K., Bertails, F., Kim, T., Marschner, S.R., Cani, M., Lin, M.C.: A survey on hair modeling: styling, simulation, and rendering. IEEE Trans. Vis. Comput. Graph. **13**(2), 213–234 (2007)
66. Wei, L., Hu, L., Kim, V.G., Yumer, E., Li, H.: Real-time hair rendering using sequential adversarial networks. In: ECCV, vol. 11208, pp. 105–122 (2018)
67. Xu, K., Ma, L., Ren, B., Wang, R., Hu, S.: Interactive hair rendering and appearance editing under environment lighting. ACM Trans. Graph. **30**(6), 173 (2011)
68. Yan, L., Tseng, C., Jensen, H.W., Ramamoorthi, R.: Physically-accurate fur reflectance: modeling, measurement and rendering. ACM Trans. Graph. **34**(6), 185:1–185:13 (2015)
69. Yi, Z., Zhang, H.R., Tan, P., Gong, M.: DualGAN: unsupervised dual learning for image-to-image translation. In: ICCV, pp. 2868–2876 (2017)
70. Yuksel, C., Schaefer, S., Keyser, J.: Hair meshes. ACM Trans. Graph. **28**(5), 166 (2009)
71. Zhang, M., Chai, M., Wu, H., Yang, H., Zhou, K.: A data-driven approach to four-view image-based hair modeling. ACM Trans. Graph. **36**(4), 156:1–156:11 (2017)
72. Zhou, Y., et al.: HairNet: single-view hair reconstruction using convolutional neural networks. In: ECCV, vol. 11215, pp. 249–265 (2018)
73. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV, pp. 2242–2251 (2017)
74. Zinke, A., Yuksel, C., Weber, A., Keyser, J.: Dual scattering approximation for fast multiple scattering in hair. ACM Trans. Graph. **27**(3), 32 (2008)