Real-Time Hair Rendering using Sequential Adversarial Networks

Lingyu Wei[1,2][0000−0001−7278−4228], Liwen Hu[1,2],
Vladimir Kim[3], Ersin Yumer[4], and Hao Li[1,2]

[1] Pinscreen Inc., Los Angeles CA 90025, USA
[2] University of Southern California, Los Angeles CA 90089, USA
[3] Adobe Research, San Jose CA 95110, USA
[4] Argo AI, Pittsburgh PA 15222, USA

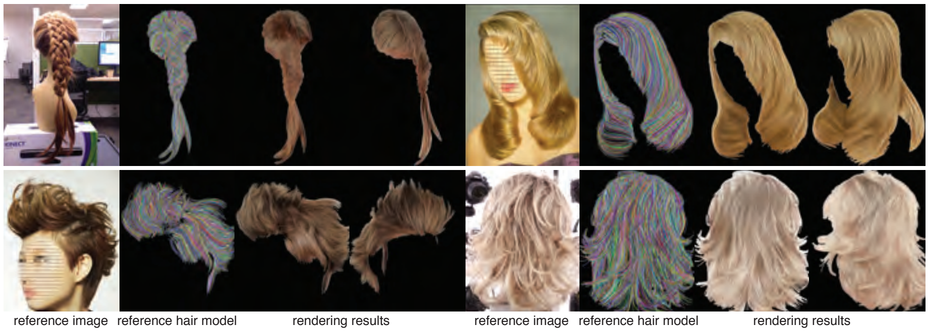

Fig. 1. We propose a real-time hair rendering method. Given a reference image, we can render a 3D hair model with the referenced color and lighting in real-time. Faces in this paper are obfuscated to avoid copyright infringement

Abstract. We present an adversarial network for rendering photorealistic hair as an alternative to conventional computer graphics pipelines. Our deep learning approach does not require low-level parameter tuning nor ad-hoc asset design. Our method simply takes a strand-based 3D hair model as input and provides intuitive user-control for color and lighting through reference images. To handle the diversity of hairstyles and its appearance complexity, we disentangle hair structure, color, and illumination properties using a sequential GAN architecture and a semi-supervised training approach. We also introduce an intermediate edge activation map to orientation field conversion step to ensure a successful CG-to-photoreal transition, while preserving the hair structures of the original input data. As we only require a feed-forward pass through the network, our rendering performs in real-time. We demonstrate the synthesis of photorealistic hair images on a wide range of intricate hairstyles and compare our technique with state-of-the-art hair rendering methods.

Keywords: Hair rendering · GAN

1 Introduction

Computer-generated (CG) characters are widely used in visual effects and games, and are becoming increasingly prevalent in photo manipulation and in virtual reality applications. Hair is an essential visual component of virtual characters. However, while significant advancements in hair rendering have been made in the computer graphics community, the production of aesthetically realistic and desirable hair renderings still relies on a careful design of strand models, shaders, lights, and composites, generally created by experienced look development artists. Due to the geometric complexity and volumetric structure of hair, modern hair rendering pipelines often combine the use of efficient hair representation, physically-based shading models, shadow mapping techniques, and scattering approximations, which not only increase the computational cost, but also the difficulty for tweaking parameters. In high-end film production, it is not unusual that a single frame for a photorealistic hair on a rendering farm takes several minutes to generate. While compelling real-time techniques have been introduced recently, including commercial solutions (e.g., NVIDIA HairWorks, Unity Hair Tools), the results often appear synthetic and are difficult to author, even by skilled digital artists. For instance, several weeks are often necessary to produce individualized hair geometries, textures, and shaders for hero hair assets in modern games, such as Uncharted 4 and Call of Duty: Ghosts.

Inspired by recent advances in generative adversarial networks (GANs), we introduce the first deep learning-based technique for rendering photorealistic hair. Our method takes a 3D hair model as input in strand representation and uses an example input photograph to specify the desired hair color and lighting. In addition to our intuitive user controls, we also demonstrate real-time performance, which makes our approach suitable for interactive hair visualization and manipulation, as well as 3D avatar rendering.

Compared to conventional graphics rendering pipelines, which are grounded on complex parametric models, reflectance properties, and light transport simulation, deep learning-based image synthesis techniques have proven to be a promising alternative for the efficient generation of photorealistic images. Successful image generations have been demonstrated on a wide range of data including urban scenes, faces, and rooms, but fine level controls remain difficult to implement. For instance, when conditioned on a semantic input, arbitrary image content and visual artifacts often appear, and variations are also difficult to handle due to limited training samples. This problem is further challenged by the diversity of hairstyles, the geometric intricacy of hair, and the aesthetic complexity of hair in natural environments. For a viable photorealistic hair rendering solution, we need to preserve the intended strand structures of a given 3D hair model, as well as provide controls such as color and lighting.

Furthermore, the link between CG and real-world images poses another challenge, since such training data is difficult to obtain for supervised learning. Photoreal simulated renderings are time-consuming to generate and often difficult to match with real-world images. In addition, capturing photorealistic hair models is hard to scale, despite advances in hair digitization techniques.

In this work, we present an approach, based on a sequential processing of a rendered input hair models using multiple GANs, that converts a semantic strand representation into a photorealistic image (See Fig. 1 and Section 5). Color and lighting parameters are specified at intermediate stages. The input 3D hair model is first rendered without any shading information, but strand colors are randomized to reveal the desired hair structures. We then compute an edge activation map, which is an important intermediate representation based on adaptive thresholding, which allows us to connect the strand features between our input CG representation and a photoreal output for effective semi-supervised training. A conditional GAN is then used to translate this edge activation map into a dense orientation map that is consistent with those obtained from real-world hair photographs. We then concatenate two multi-modal image translation networks to disentangle color and lighting control parameters in latent space. These high-level controls are specified using reference hair images as input, which allows us to describe complex hair color variations and natural lighting conditions intuitively. We provide extensive evaluations of our technique and demonstrate its effectiveness on a wide range of hairstyles. We compare our rendered images to ground truth photographs and renderings obtained from state-of-the-art computer graphics solutions. We also conduct a user study to validate the achieved level of realism.

Contributions: We demonstrate that a photorealistic and directable rendering of hair is possible using a sequential GAN architecture and an intermediate conversion from edge activation map to orientation field. Our network decouples color, illumination, and hair structure information using a semi-supervised approach and does not require synthetic images for training. Our approach infers parameters from input examples without tedious explicit low-level modeling specifications. We show that color and lighting parameters can be smoothly interpolated in latent space to enable fine-level and user-friendly control. Compared to conventional hair rendering techniques, our method does not require any low-level parameter tweaking or ad-hoc texture design. Our rendering is computed in a feed forward pass through the network, which is fast enough for real-time applications. Our method is also significantly easier to implement than traditional global illumination techniques. We plan to release the code and data to the public.[‡]

2 Related Work

In this section we provide an overview of state-of-the-art techniques for hair rendering and image manipulation and synthesis.

Fiber-level hair renderings produce highly realistic output, but incurs substantial computational cost [31, 8, 43, 42, 44], but also require some level of expertise for asset preparation and parameter tuning by a digital artist. Various simplified models have been proposed recently, such as dual scattering [54], but

[‡] Available on project page: `http://cosimo.cn/#hair_render`

its real-time variant have a rather plastic and solid appearance. Real-time rendering techniques generally avoid physically-based models, and instead rely on approximations that only mimics its appearance, by modeling hair as parametric surfaces [25, 24], meshes [46, 18], textured morphable models [2], or multiple semi-transparent layers [40, 45]. Choosing the right parametric model and setting the parameters for the desired appearance requires substantial artist expertise. Converting across different hair models can be casted as a challenging optimization or learning problem [43]. Instead, in this work, we demonstrate that one can directly learn a representation for hair structure, appearance, and illumination using a sequence of GANs, and that this representation can be intuitively manipulated by using example images.

Substantial efforts have been dedicated to estimating hair structures from natural images, such as with multi-view hair capturing methods [47, 17, 15, 29, 28, 30, 35, 34]. Recently, single-view hair reconstruction methods [3, 5, 6, 4, 16] are becoming increasingly important because of the popularity in manipulating internet portraits and selfies. We view our work as complementary to these hair capturing methods, since they rely on existing rendering techniques and do not estimate the appearance and illumination for the hair. Our method can be used in similar applications, such as hair manipulation in images [5, 6], but with simpler control over the rendering parameters.

Neural networks are increasingly used for the manipulation and synthesis of visual data such as faces [39, 19], object views [50], and materials [27]. Recently, Nalbach et al. [32] proposed how to render RGB images using CNN, but it requires aligned attributes e.g. normal and reflectance per pixel, which are not well-defined for hair strands with sub-pixel details. Generative models with an adversary [12, 36] can successfully learn a data representation without explicit supervision. To enable more control these models have been further modified to consider user input [51] or to condition on a guiding image [20]. While the latter provides a powerful manipulation tool via image-to-image translation, it requires strong supervision in the form of paired images. This limitation has been further addressed by enforcing cycle-consistency across unaligned datasets [52]. Another limitation of the original image translation architecture is that it does not handle multimodal distributions which are common in synthesis tasks. This is addressed by encouraging bijections between the output and latent spaces [53] in a recently introduced architecture known as BicycleGAN. We assess this architecture as part of our sequential GAN for hair rendering.

Our method is also related to unsupervised learning methods that remove part of the input signal, such as color [48], and then try to recover it via an auto-encoder architecture. However, in this work we focus on high quality hair renderings instead of generic image analysis, and unlike SplitBrain [48], we use image processing to connect two unrelated domains (CG hair models and real images).

Variants of these models have been applied to many applications including image compositing [26], font synthesis [1], texture synthesis [41], facial texture synthesis [33], sketch colorization [38], makeup transfer [7], and many more.

Hair has a more intricate appearance model due to its thin semi-transparent structures, inter-reflections, scattering effects, and very detailed geometry.

3 Method

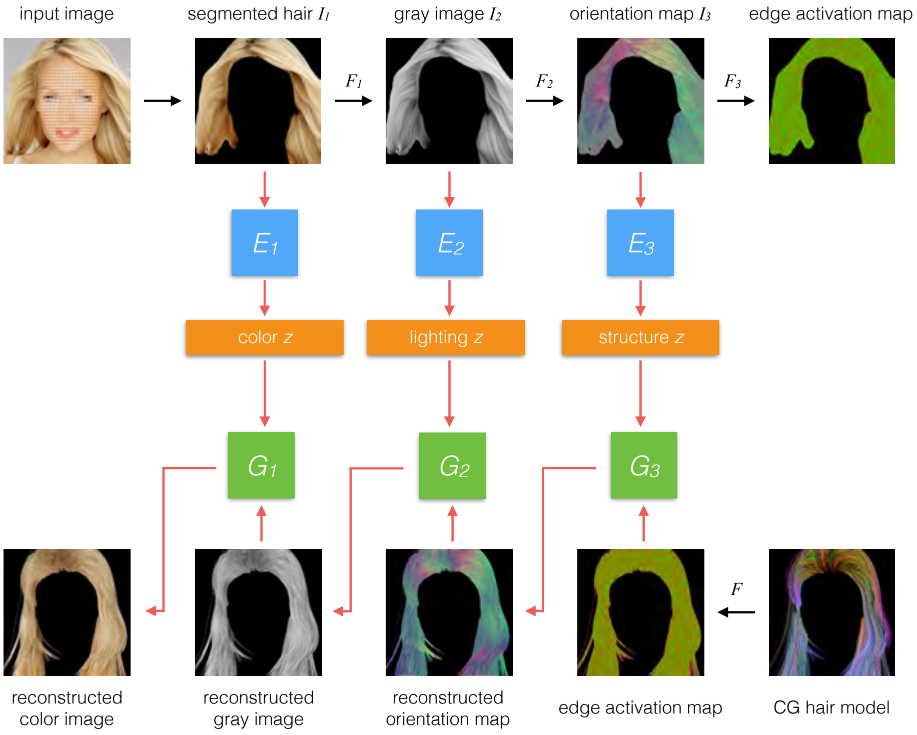

Fig. 2. Overview of our method. Given a natural image, we first use simple image processing to strip it off salient details such as color, lighting, and fiber-level structure, leaving only coarse structure captured in activation map (top row). We encode each simplified image into its own latent space which are further used by generators. The CG hair is rendered in a style mimicing the extracted coarse structure, and generators are applied in inverse order to add the details from real hair encoded in the latent space yielding the realistic reconstruction (bottom row)

We propose a semi-supervised approach to train our hair rendering network using only real hair photographs during training. The key idea of our approach is to gradually reduce the amount of information by processing the image, eventually bringing it to a simple low dimensional representation, *edge activation map*. This representation can also be trivially derived from a CG hair model, enabling us to connect two domains. The encoder-decoder architecture is applied to each simplified representation, where the encoder captures the information removed by the image processing and the decoder recovers it (see Fig. 2).

Given a 2D image I_1 we define image processing filters $\{F_i\}_{i=1}^3$ that generate intermediate simplified images $I_{i+1} := F_i(I_i)$. Each intermediate image I_i is first encoded by a network $E_i(I_i)$ to a feature vector z_i and then decoded with a generator $G_i(z_i, I_i)$. The decoder is trained to recover the information, that is lost in a particular image processing step. We use a conditional variational autoencoder GAN [53] for each encoder-decoder pair.

Our sequential image processing operates in the following three steps. First, $I_2 := F_1(I_1)$ desaturates a segmented hair region of an input photograph to produce a *grayscale image*. Second, $I_3 := F_2(I_2)$ is the *orientation map* using the maximal response of a rotating DoG filter [28]. Third, $F_3(I_3)$ is an *edge activation map* obtained using adaptive thresholding, and each pixel contains only the values 1 or -1 indicating if it is activated (response higher than its neighboring pixels) or not. This edge activation map provides a basic representation for describing hair structure from a particular viewpoint, and the edge activation map derived from natural images or rendering of CG model with random strand colors can be processed equally well with our generators. Fig. 3 demonstrates some examples of our processing pipeline applied to real hair.

At the inference time, we are given an input 3D hair model in strand representation (100 vertices each) and we render it with randomized strand colors from a desired viewpoint. We apply the full image processing stack $F_3 \circ F_2 \circ F_1$ to obtain the edge activation map. We then can use the generators $G_1 \circ G_2 \circ G_3$ to recover the realistic looking image from the the edge activation map. Note that these generators rely on encoded features z_i to recover the desired details, which provides an effective tool for controlling rendering attributes such as color and lighting. We demonstrate that our method can effectively transfer these attributes by encoding an example image and by feeding the resulting vector to the generator (where fine-grained hair structure is encoded in z_3, natural illumination and hair appearance properties are encoded in z_2, and detailed hair color is encoded in z_1).

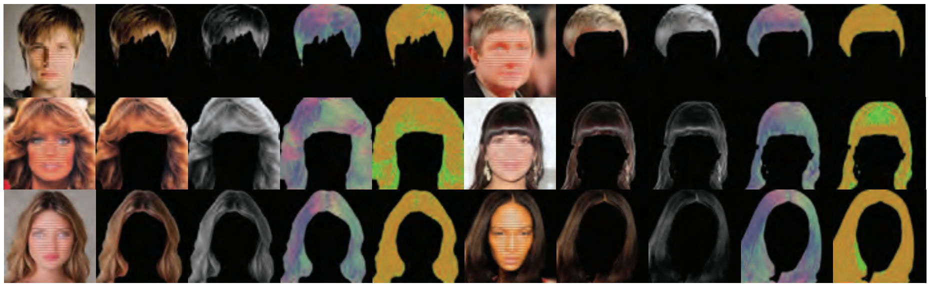

Fig. 3. Examples of our training data. For each set of images from left to right, we show input image; segmented hair; gray image; orientation map; and edge activation map

4 Implementation

Given a pair of input/output images for each stage in the rendering pipeline (e.g. the segmented color image I_1 and its grayscale version I_2) we train both the encoder network and generator network together. The encoder E_1 extracts the color information $z_1 := E_1(I_1)$, and the generator reconstructs a color image identical to I_1 using only the gray scaled image I_2 and the parameter z_1, $I_1 \approx G_1(z_1, I_2)$. These pairs of encoder and generator networks enable us to extract information available in higher dimensional image (e.g., color), represent it with a vector z_1, and then use it convert an image in the lower-dimensional domain (e.g., grayscale) back to the higher-dimensional representation.

We train three sets of networks (E_i, G_i) in a similar manner, using the training images I_i, I_{i+1} derived from the input image I via filters F_i. Since these filters are fixed, we can treat these three sets of networks independently and train them in parallel. For the rest of this section, we focus on training only a single encoder and generator pair, and thus use a simplified notation: $G := G_i, E := E_i, I := I_i, I' := I_{i+1} = F_i(I_i), z := z_i = E_i(I_i)$.

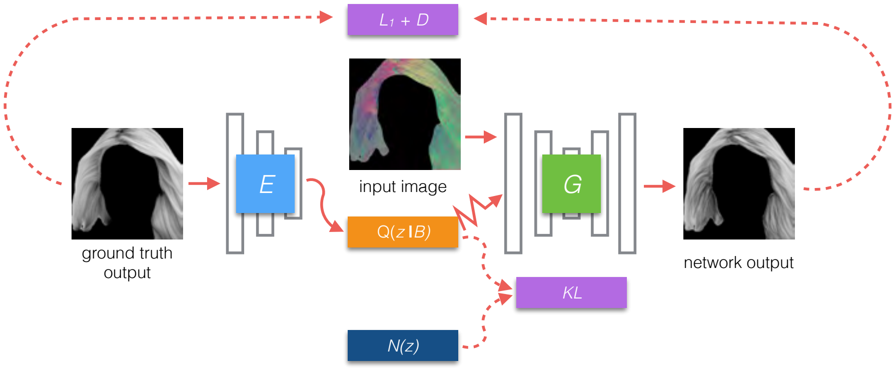

Fig. 4. Our entire network is composed of three encoder and generator pairs (E_i, D_i) with the same cVAE-GAN architecture depicted in this figure

4.1 Architecture

We train the encoder and generator network pair $((E, G))$ using the conditional variational autoencoder GAN (cVAE-GAN) architecture [53] (see Fig. 4). A ground truth image I is being processed by an encoder E, producing the latent vector z. This z and the filtered input image I' are both inputs to the generator G. Our loss function is composed of three terms:

$$\mathcal{L}_1^{\text{VAE}}(G, E) = \|I - G(E(I), I')\|_1$$

penalizes the reconstruction error between I and the generated image produced by $G(z, I')$.

$$\mathcal{L}_{\mathrm{KL}}(E) = \mathcal{D}_{\mathrm{KL}}(E(I) \| \mathcal{N}(0,1))$$

favors $z = E(I)$ to come from the normal distribution in the latent space, where $\mathcal{D}_{KL}$ is the KL-divergence of two probability distributions. This loss preserves the diversity of z and allows us to efficiently re-sample a random z from the normal distribution. Finally, an adversarial loss is introduced by training a patch-based discriminator [21] D. The discriminator takes either I or $G(z, I')$ and classifies whether the input comes from real data or from the generator. A path-based discriminator will learn to distinguish the local feature from its receptive field, and penalize artifacts produced in the local regions of $G(z, I')$.

We use the `add_to_input` method from the work of [53] to replicate z. For a tensor I' with size $H \times W \times C$ and z with size $1 \times Z$, we copy value of z, extending it to a $H \times W \times Z$ tensor, and concatenate this with the tensor I' on the channel dimension, resulting a $H \times W \times (Z + C)$ tensor. This tensor is used as G's input. We considered additional constraints, such as providing a randomly drawn latent vector to the generator [10, 11], but we did not achieve visible improvements by adding more terms to our loss function. We provide a comparison between cVAE-GAN and the BicycleGAN architecture in the results section.

4.2 Data Preparation

Since we are only focusing on hair synthesis, we mask non-hair regions to ignore their effect, and set their pixel values to black. To avoid manual mask annotation, we train Pyramid Scene Parsing Network [49] to perform automatic hair segmentation. We annotate hair masks for 3000 random images from CelebA-HQ dataset [23], and train our network on this data. We use the network to compute masks for the entire 30,000 images in CelebA-HQ dataset, and manually remove images with wrong segmentation, yielding about 27,000 segmented hair images.[‡] We randomly sampled 5,000 images from this dataset to use as our training data.

We apply same deterministic filters F_i on each image in the training data, to obtain the corresponding gray image, orientation maps, and edge activation maps.

4.3 Training

We apply data augmentation including random rotation, translation, and color perturbation (only for input RGB images) to add more variations to the training set. Scaling is not applied, as the orientation map depends on the scale of the texture details from the gray-scaled image. We choose the U-net [37] architecture for generator G, which has an encoder-decoder architecture with symmetric skip connections allowing generation of pixel-level details as well as preserving the global information. ResNet [13] is used for encoder E, which consists of 6 groups of residual blocks. In all experiments, we use a fixed resolution 512 × 512 for all images, and the dimension of z is 8 in each transformation, following the choice from the work of Zhu et al. [53]. We train each set of networks from 5000 images for 100 epochs, with a learning rate gradually decreasing from 0.0001 to zero. The training time for each set is around 24 hours. Lastly, we also add random Gaussian noise withdrawn from $\mathcal{N}(0, \sigma 1)$ with gradually decreasing σ to the image, before feeding them to D, to stabilize the GAN training.

5 Results

Geometric hair models used in Fig. 1, Fig. 6 and Fig. 9 are generated using various hair modeling techniques [16, 17, 15]. The traditional computer graphic models used for comparison in Fig. 10 are manually created in Maya with XGen. We use the model from USC Hair Salon [14] in Fig. 7.

Real-Time Rendering System. To demonstrate the utility of our method, we developed a real-time rendering interface for hair (see Fig. 5 and supplemental video). The user can load a CG hair model, pick the desired viewpoint, and then provide an example image for color and lighting specification. This example-based approach is user friendly as it is often difficult to describe hair colors using a single RGB value as they might have dyed highlights or natural variations in follicle pigmentations. Fig. 1 and Fig. 6 demonstrate the rendering results with our method. Note that our method successfully handles a diverse set of hairstyles, complex hair textures, and natural lighting conditions. One can further refine the color and lighting by providing additional examples for these attributes and interpolate between them in latent space (Fig. 7). This feature provides an intuitive user control when a desired input example is not available.

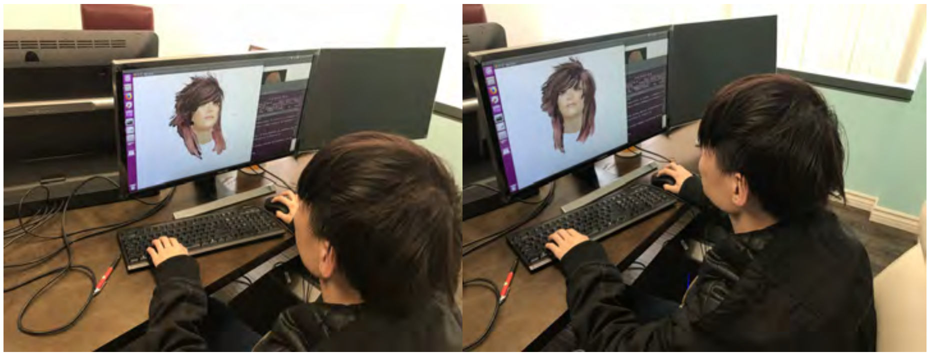

Fig. 5. Our interactive interface

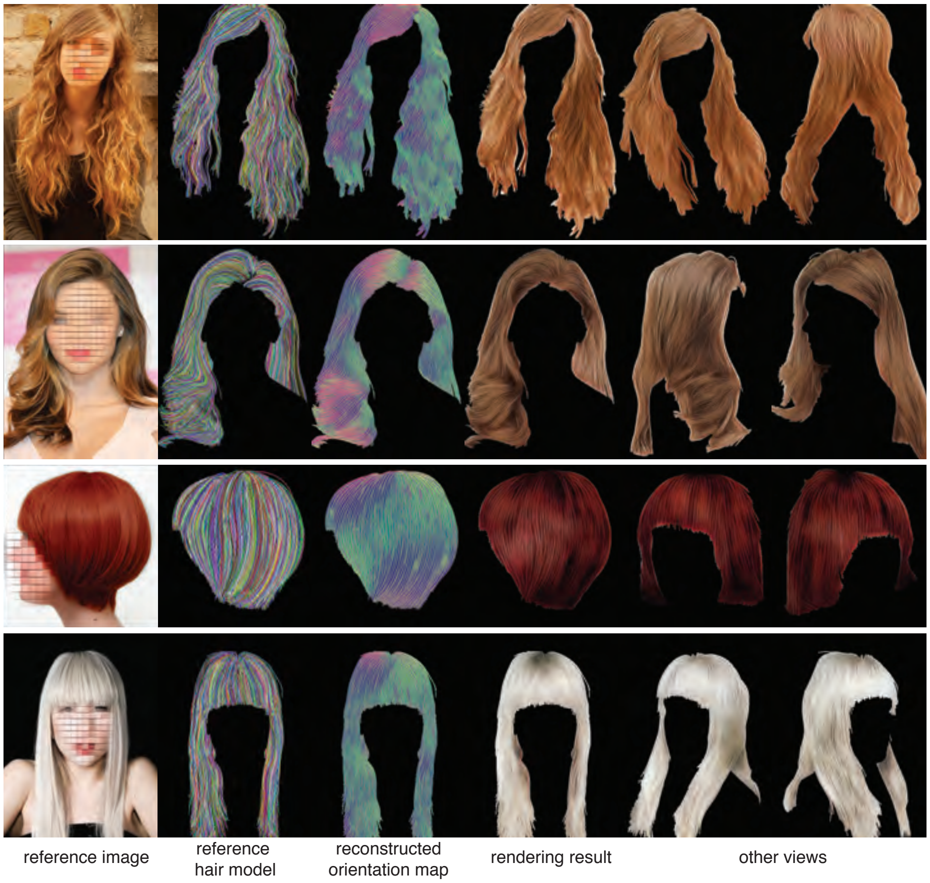

Fig. 6. Rendering results for different CG hair models where color and lighting conditions are extracted from a reference image

Comparison. We compare our sequential network to running BicycleGAN to directly render the colored image from the orientation field without using the sequential network pairs (Fig. 8). Even if we double the number of parameters and training time, we still notice that only the lighting, but not color, is accurately captured. We believe that the lighting parameters are much harder to be captured than the color parameters, hence the combined network may always try to minimize the error brought by lighting changes first, without considering the change of color as an equally important factor. To qualitatively evaluate, our system we compare to state-of-the-art rendering techniques. Unfortunately, these rendering methods do not take reference images as input, and thus lack similar level of control. We asked a professional artist to tweak the lighting and shading parameters to match our reference images. We used an enhanced version of Unity's *Hair Tool*, a high-quality real-time hair rendering plugin based on

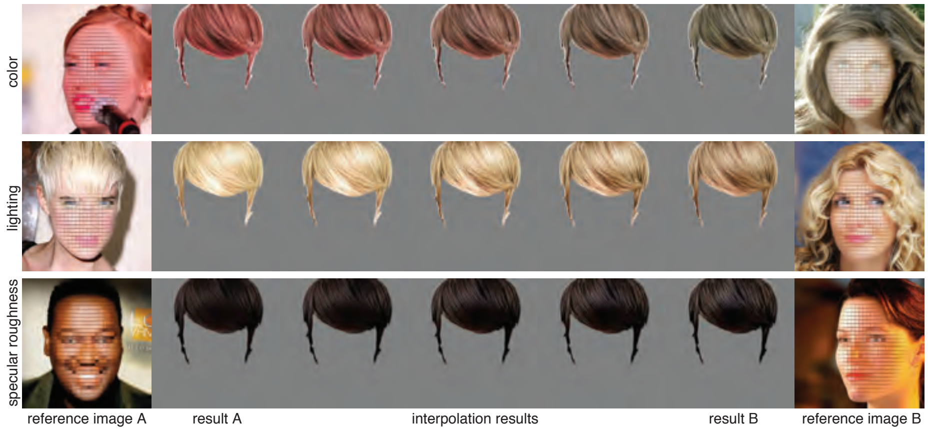

Fig. 7. We demonstrate rendering of the same CG hair model where material attributes and lighting conditions are extracted from different reference images, and demonstrate how the appearance can be further refined by interpolating between these attributes in latent space

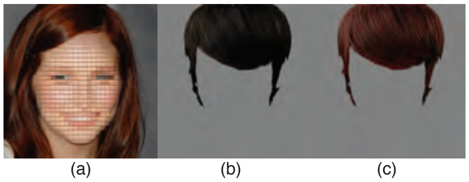

Fig. 8. Comparisons between our method with BicycleGAN (i.e., no sequential network). From left to right, we show (a) reference image; (b) rendering result with BicycleGAN; (c) rendering result with our sequential pipeline

Kajiya-Kay reflection model [22] (Fig. 9). Note that a similar real-time shading technique that approximates multiple scattering components was used in the state-of-the-art real-time avatar digitization work of [18]. Our approach appears less synthetic and contains more strand-scale texture variations. Our artist used the default hair shader in Solid Angle's Arnold renderer, which is a commercial implementation of a hybrid shaded based on the works of [9] and [54], to match the reference image. This is an offline system and it takes about 5 minutes to render a single frame (Fig. 10) on an 8-core AMD Ryzen 1800x machine with 32 GB RAM. While our renderer may not reach the level of fidelity obtained from high-end offline rendering systems, it offers real time performance, and instantaneously, produces a realistic result. It also matching the lighting and hair color of a reference image, a task that took our experienced artist over an hour to perform.

User Study. We further conduct a user study to evaluate the quality of our renderings in comparison to real hair extracted from images. We presented our

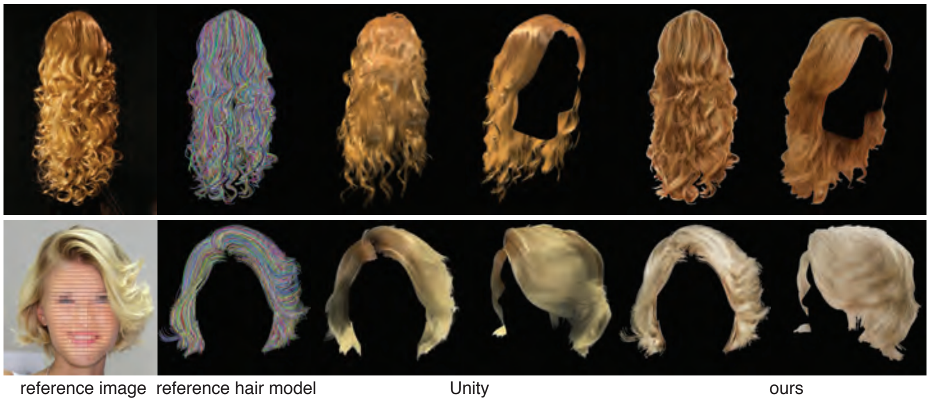

Fig. 9. Comparisons between our method with real-time rendering technique in Unity

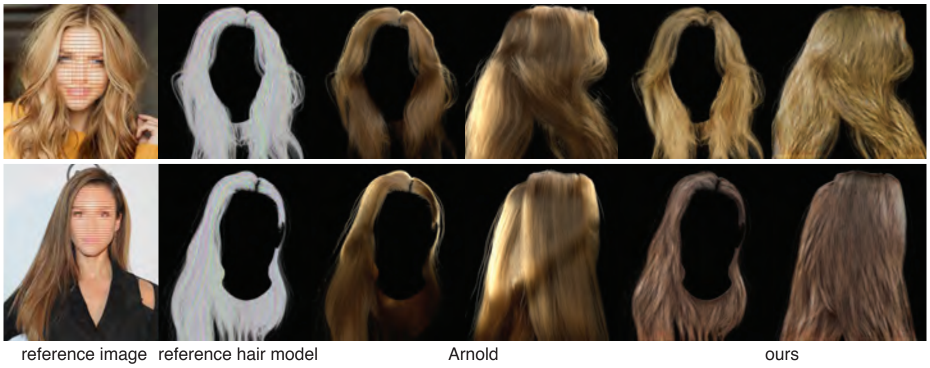

Fig. 10. Comparisons between our method with offline rendering Arnold

synthesized result and an image crop to MTurk workers side by side for 1 second, and asked them to pick an image that contained the real hair. We tested this with edge activation fields coming from either CG model or a reference image (in both cases we used a latent space from another image). If the synthetic images are not distinguishable from real images, the expected fraction of MTurk workers being "fooled" and think they are real is 50%. This is the same evaluation method as Zhu et al.'s work [53], and we showed 120 sets of randomly generated subjects to over over 100 testers (a subset of 10 subjects is provided to each person). We present qualitative results in Fig. 11 and the rate of selecting the synthetic images in Table 1. Note that in all cases, the users had a hard time distinguishing our results from real hair, and that our system successfully rendered hair produced from a CG model or a real image. We further test how human judgments change if people spend more time (3 seconds) evaluating our rendering of a CG model. We found that crowd workers become more accurate at identifying generated results, but still mislabel a significant number of examples as being real.

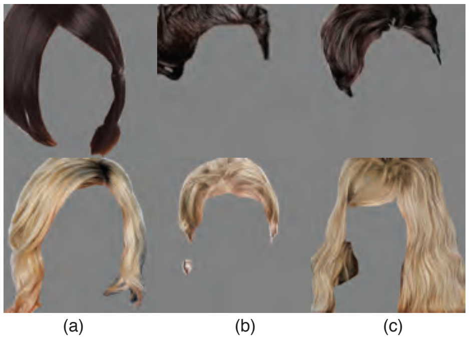

Fig. 11. User study example. From left to right, we show (a) real image, which is used to infer latent space z; (b) rendering result generated with z and the activation map of another real image; (c) rendering result of z and a CG model

Table 1. MTurk workers were provided two images in a random order and were asked to select one that looked more real to them. Either CG models or images were used to provide activation fields. Note that in all cases MTurk users had a hard time distinguishing our results from real hair, even if we extend the time that images are visible to 3 seconds

Hair Source	AMT Fooling Rate [%]
Image	49.9%
CG Model	48.5%
CG Model(3 sec)	45.6%

Performance. We measure the performance of our feed forward networks on an real-time hair rendering application. We use a desktop with 2 Titan Xp each with 12GB of GPU memory. All online processing steps including F_1, F_2, F_3 and all generator are running per frame. The average amount of time used in F_2 is 9ms for looping over all possible rotating angles, and the computation time for F_1 and F_3 are negligible. The three networks have identical architecture and thus runs in consistent speed, each taking around 15ms. For a single GPU, our demo runs at around 15fps with GPU memory consumption 2.7GB. Running the demo on multiple GPUs allows real-time performance with fps varying between 24 to 27fps. Please refer to the video included in the supplemental material.

6 Discussion

We presented the first deep learning approach for rendering photorealistic hair, which performs in real-time. We have shown that our sequential GAN architecture and semi-supervised training approach can effectively disentangle strand-level structures, appearance, and illumination properties from the highly complex and diverse range of hairstyles. In particular, our evaluations show that without our sequential architecture, the lighting parameter would dominate over color, and color specification would no longer be effective. Moreover, our trained latent space is smooth, which allows us to interpolate continuously between arbitrary color and lighting samples. Our evaluations also suggests that there are no significant differences between a vanilla conditional GAN and a state-of-the-art network such as bicycleGAN, which uses additional smoothness constraints in the training. Our experiments further indicate that a direct conversion from a CG rendering to a photoreal image using existing adversarial networks would lead to significant artifacts or unwanted hairstyles. Our intermediate conversion step from edge activation to orientation map has proven to be an effective way for semi-supervised training and transitioning from synthetic input to photoreal output while ensuring that the intended hairstyle structure is preserved.

Limitations and Future Work. As shown in the video demo, the hair rendering is not entirely temporally coherent when rotating the view. While the per frame predictions are reasonable and most strand structure are consistent between frames, there are still visible flickering artifacts. We believe that temporal consistency could be trained with augmentations with 3D rotations, or video training data.

We believe that our sequential GAN architecture for parameter separation and our intermediate representation for CG-to-photoreal conversion could be generalized for the rendering of other objects and scenes beyond hair. Our method presents an interesting alternative and complementary solution for many applications, such as hair modeling with interactive visual feedback, photo manipulation, and image-based 3D avatar rendering.

While we do not provide the same level of fine-grained control as conventional graphics pipelines, our efficient approach is significantly simpler and generates more realistic output without any tedious fine tuning. Nevertheless, we would like to explore the ability to specify precise lighting configurations and advanced shading parameters for a seamless integration of our hair rendering into virtual environments and game engines. We believe that additional training with controlled simulations and captured hair data would be necessary.

Like other GAN techniques, our results are also not fully indistinguishable from real ones for a trained eye and an extended period of observation, but we are confident that our proposed approach would benefit from future advancements in GANs.

Acknowledgments. This work was supported in part by the ONR YIP grant N00014-17-S-FO14, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the Andrew and Erna Viterbi Early Career Chair, the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, and Adobe. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. We thank Radomír Měch for insightful discussions.

References

1. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. CVPR (2018)
2. Cao, C., Wu, H., Weng, Y., Shao, T., Zhou, K.: Real-time facial animation with image-based dynamic avatars. ACM Trans. Graph. **35**(4), 126:1–126:12 (Jul 2016). https://doi.org/10.1145/2897824.2925873, http://doi.acm.org/10.1145/2897824.2925873
3. Chai, M., Luo, L., Sunkavalli, K., Carr, N., Hadap, S., Zhou, K.: High-quality hair modeling from a single portrait photo. ACM Transactions on Graphics (Proc. SIGGRAPH Asia) **34**(6) (Nov 2015)
4. Chai, M., Shao, T., Wu, H., Weng, Y., Zhou, K.: Autohair: Fully automatic hair modeling from a single image. ACM Transactions on Graphics (TOG) **35**(4), 116 (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 (Jul 2013). https://doi.org/10.1145/2461912.2461990, http://doi.acm.org/10.1145/2461912.2461990
6. Chai, M., Wang, L., Weng, Y., Yu, Y., Guo, B., Zhou, K.: Single-view hair modeling for portrait manipulation. ACM Transactions on Graphics (TOG) **31**(4), 116 (2012)
7. Chang, H., Lu, J., Yu, F., Finkelstein, A.: Makeupgan: Makeup transfer via cycle-consistent adversarial networks. CVPR (2018)
8. d'Eon, E., Francois, G., Hill, M., Letteri, J., Aubry, J.M.: An energy-conserving hair reflectance model. In: Proceedings of the Twenty-second Eurographics Conference on Rendering. pp. 1181–1187. EGSR '11, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2011). https://doi.org/10.1111/j.1467-8659.2011.01976.x, http://dx.doi.org/10.1111/j.1467-8659.2011.01976.x
9. d'Eon, E., Marschner, S., Hanika, J.: Importance sampling for physically-based hair fiber models. In: SIGGRAPH Asia 2013 Technical Briefs. pp. 25:1–25:4. SA '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2542355.2542386, http://doi.acm.org/10.1145/2542355.2542386
10. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. CoRR **abs/1605.09782** (2016), http://arxiv.org/abs/1605.09782
11. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., Courville, A.C.: Adversarially learned inference. CoRR **abs/1606.00704** (2016)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 2672–2680. NIPS'14, MIT Press, Cambridge, MA, USA (2014), http://dl.acm.org/citation.cfm?id=2969033.2969125
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
14. Hu, L.: http://www-scf.usc.edu/ liwenhu/shm/database.html (2015)
15. Hu, L., Ma, C., Luo, L., Li, H.: Robust hair capture using simulated examples. ACM Transactions on Graphics (Proc. SIGGRAPH) **33**(4) (Jul 2014)
16. Hu, L., Ma, C., Luo, L., Li, H.: Single-view hair modeling using a hairstyle database. ACM Transactions on Graphics (Proc. SIGGRAPH) **34**(4) (Aug 2015)
17. Hu, L., Ma, C., Luo, L., Wei, L.Y., Li, H.: Capturing braided hairstyles. ACM Trans. Graph. **33**(6), 225:1–225:9 (2014)

18. Hu, L., Saito, S., Wei, L., Nagano, K., Seo, J., Fursund, J., Sadeghi, I., Sun, C., Chen, Y.C., Li, H.: Avatar digitization from a single image for real-time rendering. ACM Trans. Graph. **36**(6), 195:1–195:14 (Nov 2017). https://doi.org/10.1145/3130800.31310887, http://doi.acm.org/10.1145/3130800.31310887
19. Huynh, L., Chen, W., Saito, S., Xing, J., Nagano, K., Jones, A., Debevec, P., Li, H.: Photorealistic facial texture inference using deep neural networks. In: Computer Vision and Pattern Recognition (CVPR). pp. –. IEEE (2018)
20. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2016)
21. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CoRR **abs/1611.07004** (2016), http://arxiv.org/abs/1611.07004
22. Kajiya, J.T., Kay, T.L.: Rendering fur with three dimensional textures. SIGGRAPH Comput. Graph. **23**(3), 271–280 (Jul 1989). https://doi.org/10.1145/74334.74361, http://doi.acm.org/10.1145/74334.74361
23. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Hk99zCeAb
24. Kim, T.Y., Neumann, U.: Interactive multiresolution hair modeling and editing. ACM Trans. Graph. **21**(3), 620–629 (Jul 2002). https://doi.org/10.1145/566654.566627, http://doi.acm.org/10.1145/566654.566627
25. Lee, D.W., Ko, H.S.: Natural hairstyle modeling and animation. Graph. Models **63**(2), 67–85 (Mar 2001). https://doi.org/10.1006/gmod.2001.0547, http://dx.doi.org/10.1006/gmod.2001.0547
26. Lin, C., Lucey, S., Yumer, E., Wang, O., Shechtman, E.: St-gan: Spatial transformer generative adversarial networks for image compositing. In: Computer Vision and Pattern Recognition, 2018. CVPR 2018. IEEE Conference on. pp. –. - (2018)
27. Liu, G., Ceylan, D., Yumer, E., Yang, J., Lien, J.M.: Material editing using a physically based rendering network. In: ICCV (2017)
28. Luo, L., Li, H., Paris, S., Weise, T., Pauly, M., Rusinkiewicz, S.: Multi-view hair capture using orientation fields. In: Computer Vision and Pattern Recognition (CVPR) (Jun 2012)
29. Luo, L., Li, H., Rusinkiewicz, S.: Structure-aware hair capture. ACM Transactions on Graphics (Proc. SIGGRAPH) **32**(4) (Jul 2013)
30. Luo, L., Zhang, C., Zhang, Z., Rusinkiewicz, S.: Wide-baseline hair capture using strand-based refinement. In: Computer Vision and Pattern Recognition (CVPR) (Jun 2013)
31. 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 (Jul 2003). https://doi.org/10.1145/882262.882345, http://doi.acm.org/10.1145/882262.882345
32. Nalbach, O., Arabadzhiyska, E., Mehta, D., Seidel, H.P., Ritschel, T.: Deep shading: convolutional neural networks for screen space shading. In: Computer graphics forum. vol. 36, pp. 65–78. Wiley Online Library (2017)
33. Olszewski, K., Li, Z., Yang, C., Zhou, Y., Yu, R., Huang, Z., Xiang, S., Saito, S., Kohli, P., Li, H.: Realistic dynamic facial textures from a single image using gans. ICCV (2017)

34. Paris, S., Briceño, H.M., Sillion, F.X.: Capture of hair geometry from multiple images. In: ACM Transactions on Graphics (TOG). vol. 23, pp. 712–719. ACM (2004)
35. Paris, S., Chang, W., Kozhushnyan, O.I., Jarosz, W., Matusik, W., Zwicker, M., Durand, F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. In: ACM Transactions on Graphics (TOG). vol. 27, p. 30. ACM (2008)
36. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015), http://arxiv.org/abs/1505.04597
38. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. Computer Vision and Pattern Recognition, CVPR (2017)
39. Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., Samaras, D.: Neural face editing with intrinsic image disentangling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
40. Sintorn, E., Assarsson, U.: Hair self shadowing and transparency depth ordering using occupancy maps. In: Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games. pp. 67–74. I3D '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1507149.1507160, http://doi.acm.org/10.1145/1507149.1507160
41. Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., Hays, J.: Texturegan: Controlling deep image synthesis with texture patches. CVPR (2018)
42. Yan, L.Q., Jensen, H.W., Ramamoorthi, R.: An efficient and practical near and far field fur reflectance model. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017) **36**(4) (2017)
43. Yan, L.Q., Sun, W., Jensen, H.W., Ramamoorthi, R.: A bssrdf model for efficient rendering of fur with global illumination. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2017) (2017)
44. Yan, L.Q., Tseng, C.W., Jensen, H.W., Ramamoorthi, R.: Physically-accurate fur reflectance: Modeling, measurement and rendering. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2015) **34**(6) (2015)
45. Yu, X., Yang, J.C., Hensley, J., Harada, T., Yu, J.: A framework for rendering complex scattering effects on hair. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. pp. 111–118. I3D '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2159616.2159635, http://doi.acm.org/10.1145/2159616.2159635
46. Yuksel, C., Schaefer, S., Keyser, J.: Hair meshes. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009) **28**(5), 166:1–166:7 (2009). https://doi.org/10.1145/1661412.1618512, http://doi.acm.org/10.1145/1661412.1618512
47. Zhang, M., Chai, M., Wu, H., Yang, H., Zhou, K.: A data-driven approach to four-view image-based hair modeling. ACM Transactions on Graphics (TOG) **36**(4), 156 (2017)
48. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: CVPR (2017)
49. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6230–6239 (July 2017). https://doi.org/10.1109/CVPR.2017.660

50. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: European Conference on Computer Vision (2016)
51. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Proceedings of European Conference on Computer Vision (ECCV) (2016)
52. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. ICCV (2017)
53. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems 30 (2017)
54. Zinke, A., Yuksel, C., Weber, A., Keyser, J.: Dual scattering approximation for fast multiple scattering in hair. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) **27**(3), 32:1–32:10 (2008). https://doi.org/10.1145/1360612.1360631, http://doi.acm.org/10.1145/1360612.1360631