Pacific Graphics (2020)
E. Eisemann, A. Jacobson, and F.-L Zhang (Editors)

An Energy-Conserving Hair Shading Model Based on Neural Style Transfer

Zhi Qiao, Takashi Kanai

The University of Tokyo, Graduate School of Arts and Sciences

Abstract

We present a novel approach for shading photorealistic hair animation, which is the essential visual element for depicting realistic hairs of virtual characters. Our model is able to shade high-quality hairs quickly by extending the conditional Generative Adversarial Networks. Furthermore, our method is much faster than the previous onerous rendering algorithms and produces fewer artifacts than other neural image translation methods. In this work, we provide a novel energy-conserving hair shading model, which retains the vast majority of semi-transparent appearances and exactly produces the interaction with lights of the scene. Our method is effortless to implement, faster and computationally more efficient than previous algorithms.

CCS Concepts

• ***Computing methodologies*** → ***Image-based rendering;*** *Neural networks;*

1. Introduction

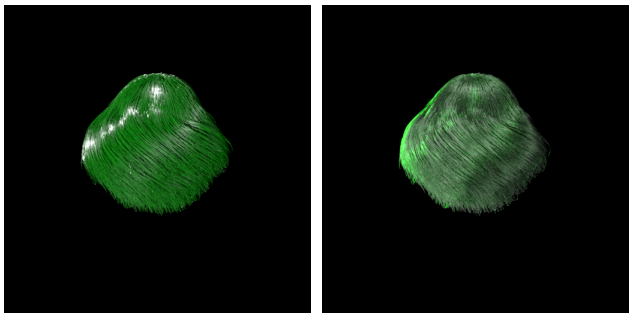
Photorealistic hair rendering is one of the first noticeable aspects of virtual characters that conveys stunning visual satisfaction. Hair structure and texture are conceived with a certain personality to impressive virtual characters. However, current real-time hair rendering approaches produce unrealistic output and tend to be artificial. As we know, humans generally have numerous strands of hair that form an extremely complicated geometric structure. Each fiber has a complex shape, especially in the case of long hair. Therefore, it is quite hard to represent each and every hair detail accurately by using any of the available modeling schemes. High-quality hair rendering has several challenging problems that are difficult to solve, and pose as barriers to achieve realistic appearances. Three main properties need to be taken into account: single scattering, multiple scattering, and thin visibility. However, rendering hairs with all of these properties in mind is computationally expensive. To find a way to resolve the contradiction between performance and quality, we introduce a novel method to achieve photorealistic hair animation, where high-performance hair and high-quality hair are both provided as references. Current machine learning researches have made great progress in image translation, which can translate images from one domain to another. Similarly, our key idea is to train a generator, to transfer low-quality hair shading images to photorealistic hair shading ones. By consolidating current researches on image translation, our method applies an unsupervised model to achieve photorealistic hair shading. The next challenge is keeping the results temporally coherent. Because our application scenarios comprise of a set of frame sequences, the temporal coherence problem is a gap that we must bridge.

This work is based on conditional Generative Adversarial Networks (cGAN), and is also inspired by recent style transfer researches. We propose here an unsupervised learning method, which builds on the Cycle-GAN [ZPIE17a] architecture. In our application, the Cycle-GAN directly applied to hair shading transfer comes with a temporal unstable problem that yields abnormal highlight appearances. To resolve this problem, other modules are added such as temporal coherence modules and highlight correction modules.

2. Related Work

Hair rendering. An early typical technology in hair rendering was proposed by Kajiya et al. [KK89, YTJR15]. This model uses a single hair fiber for scattering light and is composed of a diffuse term and a specular term. Because of the simplicity of this model, it is widely used in real-time applications such as games or interactive movies. However, this method results in a flat hair due to inadequate prediction of the azimuthal dependence of the scattering intensity and its diffusion term. Thereafter, Marschner et al. [MJC*03] proposed a model that treats each hair fiber as a translucent cylinder.

On the other hand, several simplified hair rendering models were proposed for real-time scenes. Zinke [Zin08] designed a model to handle simple light sources, but this model does not directly consider the light integration and transport complexities under environment lighting. Ren et al. [RZL*10] proposed an algorithm for real-time hair rendering with both single and multiple scattering effects under complex environment lighting. This method approximates the environment light by a set of spherical radial basis functions. Erik et al. [JCLR19] presented an approximation of strand-



(a) (b)

Figure 1: *The left figure is rendered by the fast shading model, the material consists of roughness, highlight size and whiteness attributes. The right figure is rendered by state-of-the-art reflectance model based on d'Eon and a Zinke's model. Even in the same scene, illumination appearances are totally different. It means that our method cannot apply supervised learning.*

based hair for hybrid hair rendering. All of these methods do not sufficiently consider significant phenomena such as directionality of multiple scattering, inter-reflections, subtle blurring and color-shifting effects.

General Style Transfer. A variety of works addressed the general style transfer in the past. Inspired by recent advances of GANs [GPAM*14], various types of advance style transfer methods based on the image-to-image translation [GEB16] have been proposed. Isola et al. [IZZE17] pioneeringly proposed a general purpose solution that leverages cGANs for image-to-image translation called as Pix2pix-GAN. This method is widely applied in many applications such as semantic labels and edges to photos. As an extension of this method in unsupervised learning, Cycle-GAN [ZPIE17b] was introduced for achieving bijective consistency between two domains, thereby providing photorealistic and diverse results.

As a method which uses a Cycle-GAN for video-to-video translation, Aayush et al. [BMRS18] proposed Recycle-GAN, which applies a predictor to synthesize the next frame as shown in Figure. 2 (b). This method achieves the temporal coherence for video style transfer. Yang et al. [CPY*19] also devised an unpaired video-to-video translation based on Aayush's method as shown in Figure. 2 (c). They use the FlowNet to generate an optical flow instead of the recurrent temporal predictor in [BMRS18].

In this paper, we extend a Cycle-GAN to photorealistic hair transfer, which is faster than previous offline hair rendering. Instead of the predictor of Recycle-GAN and the FlowNet of Mocycle-GAN, we directly apply the motion vectors generated from the rendering pipeline to predict the next frame. In addition, the highlight correction module is also introduced in our model. Lastly, our work uses reference images from the target domain to synthesize the specified desired hair appearances.

3. Our Model

Figure 1 (a) and (b) denote images rendered by using a fast shading model and a high-quality reflectance model in the same lighting environment, respectively. Obviously, it can be shown that the highlight appearance is totally different between two images.

Suppose we use Pix2pix-GAN to train our model and the Figure 1 (b) is taken as the ground truth, the neural network will tend to learn a mapping that translates the highlight appearance from Figure 1 (a) to Figure 1 (b). However, the correct mapping in the training stage does not mean that this map is also correct in the inference stage. Even though highlight appearances of the ground truth is correct in this case, the trained mapping is still unstable and will cause unpredictable inference results. Because in different scenes, the highlight position and appearances are too complex and irregular to build a correct mapping between the hair image rendered by a fast model and a high-quality model. That is to say, in the inference stage, if the lighting or position or viewpoint is slightly changed, the trained model would fail to map the correct highlight position and appearance. Above all, in this application, the hair rendered by a high-quality model like Figure 1 (b) cannot be treated as the ground truth. This is why we choose to use Cycle-GAN as the unsupervised model.

We also apply a small number of references as styles, which comes from the target domain. In the following section, we will show how the references affect the results by the same input. Our model would need to know where is the appropriate highlight area; otherwise, it could generate wrong highlight appearance in the test and applications. For this reason, we formulate a specular generator to extract a specular map (highlight map) from RGB images. By using this generator, the highlighted area could be constrained in the appropriate position that is close to the input highlighted area. Furthermore, the temporal coherence always inhibits the video style transfer. In contrast to conventional complicated methods, our approach directly uses the motion vector extracted from the rendering pipeline to predict the next frame. An illustration of the overall network structure is depicted in Figure 2.

The original Cycle-GAN uses two generators to map between domain X and domain Y by the adversarial loss. Otherwise, the cycle consistency loss also constrains the mapping to be the one-to-one mapping, which forces different samples in the source domain to translate in the target domain. That is to say, Cycle-GAN achieves a bijective mapping between two domains. On the other hand, unlike Cycle-GAN, our model builds a feedforward hair shading transformation network. We firstly introduce the reference to the adversarial and cycle consistency losses.It should be observed in the training or inference stage, whether the reference consists of only one image obtained from the corresponding domain for one kind of hairstyle. These losses constrain the results of G_X and G_Y according to the reference: r_x and r_y which come from the corresponding domain X and Y. The adversarial loss for G_X and G_Y

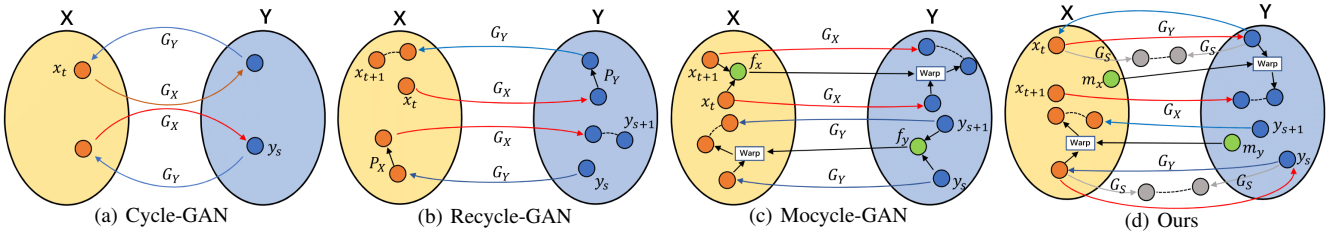


Figure 2: *(a) Cycle-GAN uses two generators (G_X and G_Y) to map between domain X and domain Y by the adversarial loss and cycle consistency loss. (b) Recycle-GAN introduces predictors (P_X and P_Y) to synthesize the future frame for ensuring temporal coherence. (c) Mocycle-GAN utilizes generated optical flows (f_x and f_y) to predict the future frame by warping the current frame. (d) Instead of generated optical flows, we introduce the more precise motion vector (m_x and m_y) to predict the future frame and the highlight constraint to keep the faithful illumination appearances by the pre-trained specular generator G_S.*

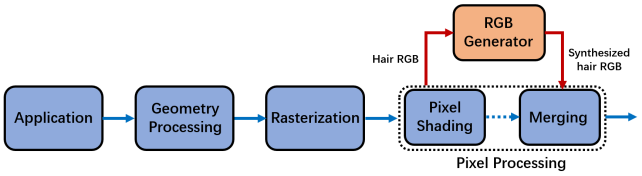


Figure 3: *In our pipeline, the hair RGB is extracted before the merging stage and imported to the trained generator. The output are synthesized photorealistic hair RGB that is sent back to the merging stage.*

are:

$$L_{GAN}(G_X, D_Y, X, Y) = \mathbb{E}_{y\sim P_Y}[\log D_Y(y)] + \mathbb{E}_{x\sim P_X, r_y\sim P_Y}[\log(1 - D_Y(G_X(x, r_y))], \quad (1)$$

$$L_{GAN}(G_Y, D_X, X, Y) = \mathbb{E}_{x\sim P_X}[\log D_X(x)] + \mathbb{E}_{y\sim P_Y, r_x\sim P_X}[\log(1 - D_X(G_Y(y, r_x))]. \quad (2)$$

The references are also applied in the cycle consistency loss:

$$L_{cyc}(G_X, G_Y) = \mathbb{E}_{x\sim P_X}[||G_Y(G_X(x, r_y), r_x) - x||_1] + \mathbb{E}_{y\sim P_Y}[||G_X(G_Y(y, r_x), r_y) - y||_1]. \quad (3)$$

A fundamental weakness of the Cycle-GAN model is that it learns mapping only at the frame level. The hairstyle in 3D animation is represented by a contiguous frame sequence. So the generator must learn to further achieve temporal coherence mapping to ensure smooth visual appearance in contiguous frames. Previous researches based on Recycle-GAN [BMRS18] and Mocycle-GAN [CPY*19] were proposed. Both methods predict future frames by training a temporal predictor or an optical flow predictor. However, Recycle-GAN does not apply the motion information from adjacent frames to promote video-to-video translation. Mocycle-GAN explicitly models motion across frames with optical flow, but the optical flow is predicted by another trained FlowNet. This will cause deviations in optical flow and distortions occur in the warping step.

Unlike previous researches, we propose a method that also utilizes the motion information, which is rendered by the graphics pipeline instead of an additional neural network. In our approach, this motion information is called *motion vector*. In the graphics pipeline, the renderer will encode a 2D vector representing the object motion of X-axis as green and Y-axis as red. Since the motion vector is directly generated from the pipeline, it can be undoubtedly regarded as a very accurate optical flow. By the motion vector, the synthetic frame can be warped with the transferred motion to the subsequent frame.

We give here two contiguous frames from domain X: x_t and x_{t+1}. As mentioned above, x_t and x_{t+1} are reconstructed as $G_Y(G_X(x_t, r_y), r_x)$ and $G_Y(G_X(x_{t+1}, r_y), r_x)$. Then, the frame x_t is warped as $W(G_Y(G_X(x_t, r_y), r_x), m_x)$, which should be similar to the next reconstructed frame $G_Y(G_X(x_{t+1}, r_y), r_x)$. Correspondingly, for two contiguous frames y_t and y_{t+1} in domain Y, the warped frame $W(G_X(G_Y(y_t, r_x), r_y), m_y)$ also should be similar to the next reconstructed frame $G_X(G_Y(y_{t+1}, r_x), r_y)$. So the temporal cycle consistency constraint is applied in the L1 distance between the warped frame and the synthetic next frame:

$$L_{tempo}(G_X, G_Y) = \mathbb{E}_{x\sim P_X}[||W(G_Y(G_X(x_t, r_y), r_x), m_x) - G_Y(G_X(x_{t+1}, r_y), r_x)||_1] + \mathbb{E}_{y\sim P_Y}[||W(G_X(G_Y(y_t, r_x), r_y), m_y) - G_X(G_Y(y_{t+1}, r_x), r_y)||_1]. \quad (4)$$

Another problem is highlight distortion, which is addressed by the specular constraint in our model. Here we introduce the specular map that is used to define a surface's highlight color, as shown in Figure 4. This specular map is rendered by the graphics pipeline together with RGB image and motion vectors. Before training the primary network, a specular generator G_S that synthesizes a specular map from an RGB image should be pre-trained. For this pre-trained network we use a standard Pix2pix-GAN model, and the dataset consists of RGB image as the input and specular map as the ground truth. Its specific architecture will be illustated in the next section. In the training, G_S extracts the specular map from synthetic RGB image: $G_X(x, r_y)$, $G_Y(y, r_x)$. The L1 loss is used to minimize the error which is the sum of all the absolute differences between the extracted specular map of the fake RGB image: $G_S(G_X(x, r_y))$, $G_S(G_Y(y, r_x))$ and the generated specular map of the real input:

$G_S(x)$ and $G_S(y)$:

$$\begin{aligned} L_{spe}(G_X, G_Y) =& \mathbb{E}_{x\sim P_X}[||G_S(G_X(x, r_y)) - G_S(x)||_1] \\ &+ \mathbb{E}_{y\sim P_Y}[||G_S(G_Y(y, r_x)) - G_S(y)||_1]. \end{aligned} \tag{5}$$

Above all, our total loss is:

$$\begin{aligned} &L(G_X, G_Y, D_X, D_Y) = L_{GAN}(G_X, D_Y, X, Y) + L_{GAN}(G_Y, D_X, X, Y) \\ &+ \lambda_C L_{cyc}(G_X, G_Y) + \lambda_T L_{tempo}(G_X, G_Y) + \lambda_S L_{spe}(G_X, G_Y). \end{aligned} \tag{6}$$

where λ_C, λ_T, and λ_S are tradeoff parameters. The complete network structure is shown in Figure. 2 (d).

4. Implementation

Network Architecture. Our model is based on Cycle-GAN [ZPIE17b], which have shown impressive results for unsupervised neural style transfer. For generators G_X, G_Y and G_S, the networks contain several Unet blocks, which are identical except in the skip connections between each layer i in the encoder and layer $n-i$ in the decoder, where n is the total number of layers. We use 70×70 PatchGANs structure [IZZE17] for the discriminator D_X and D_Y networks, which discriminate each 70×70 overlapping patches in the image that are real or fake.

Training details. We totally trained the model for 200 epochs. The learning rate is kept at 0.0002 for the first 50 epochs and linearly decay to 0 over the following 150 epochs. The solver is based on Adam optimization algorithm that is initialized from a Gaussian distribution with a standard deviation of 0.02 and a mean of 0. In the training, we set tradeoff parameters $\lambda_C = \lambda_T = 10$ and $\lambda_S = 1$.

For the training and running of the trained networks, we used a single NVIDIA GeForce RTX 2080Ti GPU with 11GB VRAM memory. The inference time is about 20ms per 512×512 frame.

Pipeline. In this application, the scene consists of hairs and other objects like head and body. All of these are passed to the pipeline in a usual way. Firstly, hairs are rendered by a fast shading model with low quality. We only require to extract the hairs RGB from the standard graphics pipeline before the merging step as shown in Figure. 3. Then, these data are set as input to the pre-trained generator G_X. G_X generates the photorealistic hairs RGB, which are finally taken back to the pipeline and the merging process is continued exporting to the frame buffer.

Experimental Setup. We need the training set that only consists of hairs instead of the entire head or body. To this end, datasets are rendered by ourselves. For the input domain, we choose a poor effect material to fastly render hair image. In principle, the inputs have highlight appearances that interact with the environment rather than static texture. Each of the hair strands should be clear instead of polygon patches. For the target domain, we used Maya Arnold's standard hair shader, which is an advanced Monte Carlo ray tracing renderer built for the demands of visual effects, and hair is rendered based on the d'Eon model for specular and Zinke model for diffuse.

We designed different kinds of hairstyles with diverse colors. For each hairstyle, we simulated and rendered a 400-frames sequence with a size of 512×512. Each frame consists of three RGB channels for domain transfer, two motion vector channels for temporal constraint, three specular map channels for highlight constraint, one depth map channel and one alpha map channel for merging with the scene. The number of total images is 6000, 4000 for the training and 2000 for the testing.

5. Results and Discussion

Figure 5 first shows our results by different inputs and references. Our model can faithfully produce photorealistic hair appearances. In the right column, the generated hairs not only exhibit realistic shapes but also inherit similar clusters from the reference. The results also show our model can synthesize various hairstyles reasonably well. We can also analyze whether the highlight constraint is necessary. Notice that the highlight appears in the same position between input and output. Concequently, our method can faithfully produce highlight appearances.

Figure 6 next shows our results and the results of several previous works by excluding the influence from reference for comparison. All the models are trained by the same dataset. In Figure 6, the right columns are results generated by Cycle-GAN, Recycle-GAN, Mocycle-GAN and ours, respectively. Obviously, our model achieves less distortions and more faithful hair appearances. The results also show the output by previous models cannot reproduce the exact highlight field. On the contrary, our method can faithfully produce highlight appearances. Without the highlight constraint, the results become more uncontrollable, this problem could lead artificial sensation in the application.

Figure 7 shows the optical flow from Mocycle-GAN in the left and our motion vectorin the right. Obviously, instead of synthesizing the fuzzy optical flow, our motion vector contains more detailed information about motion directions and speeds.

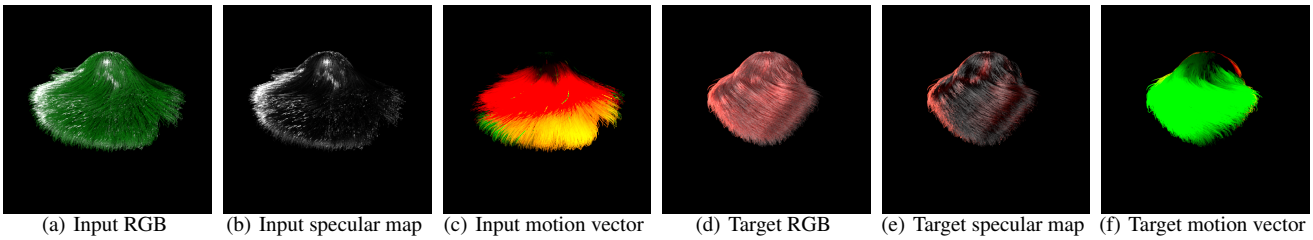
In Figure 8, we analyze whether the highlight constraint can be applied in different illumination conditions. In this experiment, we used inputs rendered by the different scenes that the illumination intensity and direction are changed. The results illustrate that our method can faithfully produce highlight appearances in different illumination conditions.

	Rendering	Synthesis	Total
Arnold Renderer	> 50s	N/A	> 50s
Ours	**< 0.70s**	**20ms**	**< 0.72s**

Table 1: *Hardware Performance. We render the dataset of target domain by Arnold Renderer and input domain by a fast rendering model. Our model Generate one frame took 10 milliseconds on average and dozens of times faster than Maya Arnold.*

Hardware performance. For the problems faced with our model, we need a state-of-the-art rendering method to get the target domain dataset. To this end, we produced the target domain hair image by Maya Arnold. Each image took more than 50 seconds. In another domain, we used a fast rendering method that took less than 0.7 seconds for each frame.

Table 1 shows that our model is dozen times faster than the direct photorealistic rendering of hairs. It should be noticed that the rendering time is an indeterminate value because the time budget is different in various hairstyles and motions.



(a) Input RGB (b) Input specular map (c) Input motion vector (d) Target RGB (e) Target specular map (f) Target motion vector

Figure 4: *Our dataset consist of three channels RGB for domain transfer, two channels motion vector for temporal constraint, three channels specular map for highlight constraint*

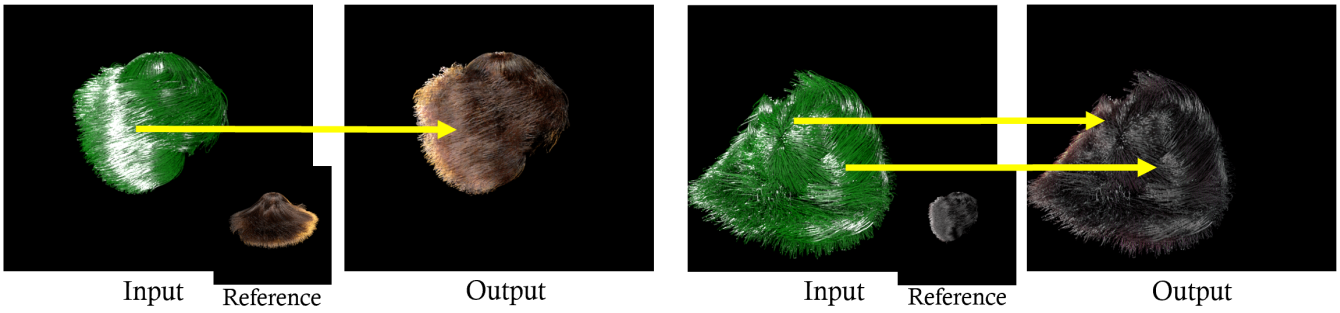


Figure 5: *Examples of various kinds colors results. The original inputs, the output results, and the different kinds of references are given.*

	FID	PSNR	SSIM
Cycle-GAN	3.97	24.99dB	0.91
Recycle-GAN	3.72	25.11dB	0.91
Mocycle-GAN	3.84	25.84dB	0.93
Ours	**3.70**	**27.07dB**	**0.98**

Table 2: *FID scores on translation quality for RGB channels hair-to-hair synthesis. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) on specular map translation quality for hair illumination appearances.*

Evaluation metrics. We firstly adopt Fréchet Inception Distance (FID) [HRU*17] for the evaluation of RGB images. FID has been widely applied to the evaluation of generated images and is a metric for calculating the feature distance between real and generated images using a pre-trained inception network. By extracting features from the RGB images and calculating the FID score, a lower score indicates that the result is closer to the target domain.

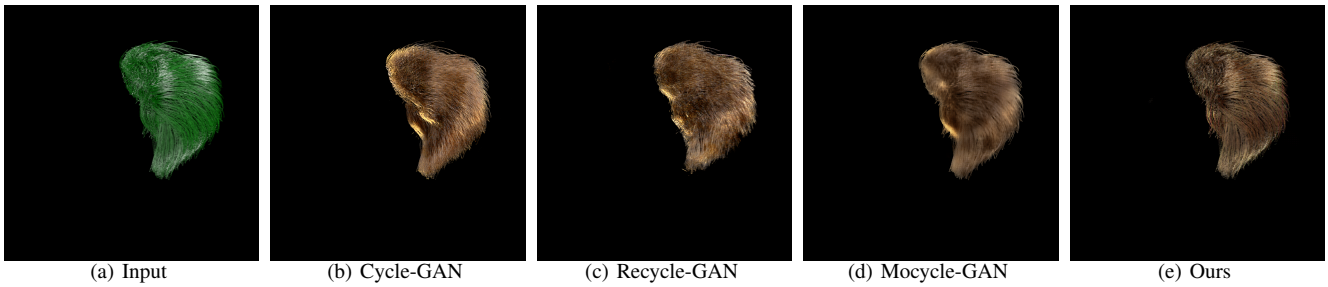
Table 2 shows the FID scores of Cycle-GAN, Recycle-GAN and our method. Note that Recycle-GAN performs better than Cycle-GAN when considering the consistency of the domain and time cycles through spatio-temporal constraints. Furthermore, Mocycle-GAN promotes pixel-wise temporal consistency by warping the synthetic frame with optical flow, which also achieves better performance than Cycle-GAN. However, our method performs best by using motion vectors that are more accurate than Mocycle-GANs and constrained in time.

In addition, the specular map has to be evaluated to provide realistic illumination to the hair. Here, Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) are introduced as image quality evaluation metrics. In this scenario, given a pair of RGB images in the output and input domains, a specular generator synthesizes the output and input specular maps, respectively. As shown in Table 2, by encouraging specular constraint, the results are consistently better for the two metrics than the other methods. This confirms the effectiveness of our specular constraints in retaining more highlighting information in hair synthesis.

6. Conclusions

We have presented an energy-conserving model that synthesizes photorealistic hair images from low-quality hair images. Our method transfers hair images to those with desired appearances according to the reference hair. We explore the continuity for the translation of a hair image sequence by our temporal constraints, which is the first time motion vectors have been applied to improve the structure and temporal continuity of hair. Our method also introduced the specular constraint to ensure faithful highlight appearances.

Compared with previous image translation methods, our model generates more convincing results and improves the preservation of illumination from the input. In particular, our model is dozens of times faster than traditional state-of-art hair rendering models. Our work suggests that the use of unsupervised image translation can faithfully reproduce photorealistic hair animation and significantly reduce computational expenses. We expect that our novel framework can be applied to the shading of semi-transparent objects like sunset glow and colored glass.



(a) Input (b) Cycle-GAN (c) Recycle-GAN (d) Mocycle-GAN (e) Ours

Figure 6: *Examples of previous works results and our results.*

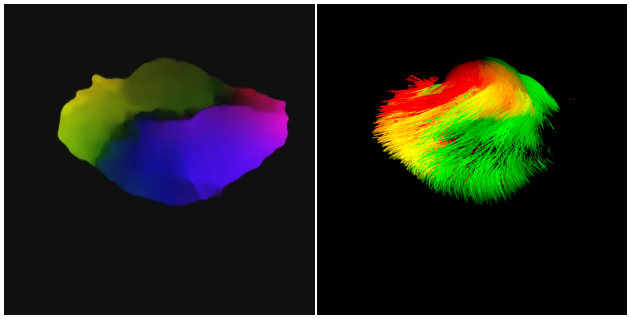


Figure 7: *The left figure is synthesized by Flownet2 from Mocycle-GAN model. The right figure is directly generated by Arnold renderer and used for predicting the future frame in our model.*

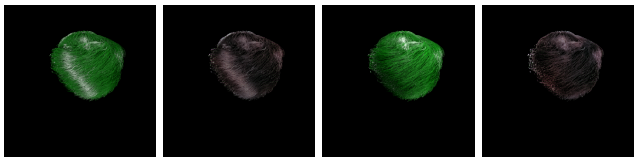


Figure 8: *Examples of illumination appearances. By encouraging the highlight constraint, the highlight has appeared in the position that is close to the input highlight. It demonstrates that our method can faithfully produce highlight appearances in different illumination conditions.*

Acknowledgements

Zhi Qiao acknowledges receipt of Japanese government (MEXT) scholarships. This work was partially supported by JSPS KAKENHI, Grant Number JP19K11990.

References

[BMRS18] BANSAL A., MA S., RAMANAN D., SHEIKH Y.: Recycle-GAN: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision (ECCV)* (September 2018), pp. 122–138. doi:10.1007/978-3-030-01228-1_8. 2, 3

[CPY*19] CHEN Y., PAN Y., YAO T., TIAN X., MEI T.: Mocycle-GAN: Unpaired video-to-video translation. In *Proceedings of the 27th ACM International Conference on Multimedia* (2019), p. 647–655. doi:10.1145/3343031.3350937. 2, 3

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2414–2423. doi:10.1109/CVPR.2016.265. 2

[GPAM*14] 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* (2014), pp. 2672–2680. doi:10.5555/2969033.2969125. 2

[HRU*17] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), p. 6629–6640. doi:10.5555/3295222.3295408. 5

[IZZE17] ISOLA P., ZHU J., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5967–5976. doi:10.1109/CVPR.2017.632. 2, 4

[JCLR19] JANSSON E. S. V., CHAJDAS M. G., LACROIX J., RAGNEMALM I.: Real-time hybrid hair rendering. In *Eurographics Symposium on Rendering - DL-only and Industry Track* (2019). doi:10.2312/sr.20191215. 1

[KK89] KAJIYA J. T., KAY T. L.: Rendering fur with three dimensional textures. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 271–280. doi:10.1145/74334.74361. 1

[MJC*03] MARSCHNER S. R., JENSEN H. W., CAMMARANO M., WORLEY S., HANRAHAN P.: Light scattering from human hair fibers. *ACM Trans. Graph. 22*, 3 (July 2003), 780–791. doi:10.1145/882262.882345. 1

[RZL*10] REN Z., ZHOU K., LI T., HUA W., GUO B.: Interactive hair rendering under environment lighting. *ACM Trans. Graph. 29*, 4 (July 2010). doi:10.1145/1778765.1778792. 1

[YTJR15] YAN L.-Q., TSENG C.-W., JENSEN H. W., RAMAMOORTHI R.: Physically-accurate fur reflectance: Modeling, measurement and rendering. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 185:1–185:13. doi:10.1145/2816795.2818080. 1

[Zin08] ZINKE A.: *Photo-Realistic Rendering of Fiber Assemblies*. Dissertation, Universität Bonn, Jan. 2008. URL: http://nbn-resolving.de/urn:nbn:de:hbz:5N-13198. 1

[ZPIE17a] ZHU J., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR abs/1703.10593* (2017). URL: http://arxiv.org/abs/1703.10593, arXiv:1703.10593. 1

[ZPIE17b] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 2242–2251. doi:10.1109/ICCV.2017.244. 2, 4