SCIENCE CHINA
Information Sciences

November 2021, Vol. 64 210104:1–210104:15
https://doi.org/10.1007/s11432-021-3325-6

• RESEARCH PAPER •
Special Focus on Visual Computing with Machine Learning

iHairRecolorer: deep image-to-video hair color transfer

Keyu WU[1†], Lingchen YANG[1†], Hongbo FU[2] & Youyi ZHENG[1*]

[1]*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou* 310058*, China;*
[2]*School of Creative Media, City University of Hong Kong, Hong Kong* 999077*, China*

Received 14 April 2021/Revised 24 June 2021/Accepted 9 August 2021/Published online 26 October 2021

Abstract In this paper, we present iHairRecolorer, the first deep-learning based approach for example-based hair color transfer in videos. Given an input video and a reference image, our method automatically transfers the hair color in the reference image to the hair in the video while keeping other hair attributes (e.g., shape, structure, and illumination) untouched, producing vivid color-transferred dynamic hair in the video. Our method performs the color transfer purely in the image space, without any form of intermediate 3D hair reconstruction. The key enabler of our method is a carefully designed conditional generative model that explicitly disentangles various hair attributes into their corresponding sub-spaces, which are implemented as conditional modules integrated into a generator. We introduce a novel spatially and temporally normalized luminance map to represent the structure and illumination of the hair. Such a representation can largely ease the burden of the generator to synthesize temporally coherent vivid dynamic hairs in the video. We further introduce a cycle consistency loss to enforce the faithfulness of the generated results with respect to the reference. We demonstrate our system's superiority in video hair color transfer by extensive experiments and comparisons to alternative methods.

Keywords hair color transfer, video manipulation, luminance map, cycle consistency

Citation Wu K Y, Yang L C, Fu H B, et al. iHairRecolorer: deep image-to-video hair color transfer. Sci China Inf Sci, 2021, 64(11): 210104, https://doi.org/10.1007/s11432-021-3325-6

1 Introduction

Human hair is essential for characterizing portraits, motivating many successful studies in computer graphics and computer vision, from hairstyle transfer [1], interactive hair shape manipulation [2] to fully controllable portrait hair editing [3]. The existing studies have mainly focused on static hair. Hair editing in videos remains relatively under-explored. In this article, we consider the problem of exemplar-based hair color transfer in video clips, and aim to transfer the hair color in a reference image to the dynamic hair in a source video.

However, different from most parts of a human face, hair is remarkably delicate, variable, and complicated. It consists of thousands of tiny strands and is subject to illumination, motion, and occlusion, making it intractable to analyze, represent, and generate. The challenges mainly arise from three aspects. First, the intricate hair shape and varying hair opacity, either induced by its structure or the complex motion, could lead to the difficulty of seamless hair-background separation across frames, i.e., imperfect hair segmentation. Second, hair strands' fine-grained features tend to vary intrinsically for their anisotropic nature, as well as extrinsically for their sensitivity to light variations (e.g., a subtle light change could cause conspicuous alteration of hair appearance). These make it arduous to distill a hair palette of the source video and then instill that of the reference without affecting other hair attributes, such as shape, structure, and illumination. Third, no ground truth information (e.g., paired video clips of the same content but exhibiting different hair colors) is available. Thus, finding a self-supervised mechanism, which can disentangle the information of hair into background, color, and other attributes, is essential to our method.

* Corresponding author (email: youyizheng@zju.edu.cn)
† Wu K Y and Yang L C have the same contribution to this work.

Luckily, with the advances of conditional generative adversarial networks (GANs), disentangling different hair attributes in a self-supervised manner becomes promising. Recently, Tan et al. [3] introduced the MichiGAN to flexibly edit hair shape, structure, and appearance in a single image. Despite the compelling results in individual frames, their solution leads to flickering artifacts when applied to our task on each frame due to a lack of temporal coherence. Refashioning their framework by leveraging optical flow [4] might seem like a feasible solution. However, due to the sophisticated structure and motion blur of dynamic hair, the predicted flow is much less accurate, and thus inappropriate for building temporal consistency. Moreover, re-synthesizing realistic moving strands conditional on a 2D hair orientation map [3] could also be extremely challenging since plenty of hair details are not captured by the orientation map. Another potential solution is to formulate this task as an exemplar-based video colorization problem [5], by directly replacing the chrominance of the source with that of the reference while keeping the luminance intact. This method could preserve the illumination and hair structure well but prone to color inaccuracy and spuriousness since color information still more or less exists in the luminance map.

Therefore, we need a system to bridge the hair color difference between the reference and the source as much as possible while simultaneously preserving the original background and other hair attributes, and more importantly, to ensure the realism and temporal coherence of transferred results. To this end, we introduce iHairRecolorer, a novel example-based system for hair color transfer from a reference image to a video. Our iHairRecolorer is based on generative neural networks and consists of three delicate conditioning mechanisms within the video generation pipeline. Specifically, spatially and temporally variant hair attributes of shape, structure, and illumination are collectively represented as a masked luminance map in the LAB color space, which is further normalized across frames to obliterate color information while maintaining temporal coherence. In contrast to less informative orientation maps [3], such a luminance map perfectly preserves hair's fine-grained structures and lighting variations, thus significantly easing the burden of our generator. More globally, hair color is represented as a feature vector encoded from a given hair region of the reference, and this feature vector serves as the generator's latent input to chromatically guide the generation process of every frame. Additionally, we use a background encoder to progressively achieve mask-aware hair-background blending in the feature domain so as to refine the imperfect hair segmentation. We integrate the aforementioned condition modules into our generator to realize the disentanglement of these attributes. The entire network is trained with our elaborately-designed loss functions, of which a novel one, considering cycle consistency, is proposed to enforce the faithfulness of the generated results to the reference.

We conduct extensive experiments on the open dataset FaceForensics [6] and our newly-collected dataset from YouTube, including qualitative and quantitative comparisons, as well as an ablation study. The results show that the proposed method outperforms all existing alternative methods on image-to-video hair color transfer. In summary, the main contributions of our work include the following.

- We introduce iHairRecolorer, the first deep learning-based approach for transforming the hair color from a reference image to a video clip.
- We for the first time exploit a hair luminance map in the LAB color space to represent hair structure and illumination, and prove its effectiveness in preserving fine-grained hair geometries than other alternative representations.
- We employ a novel cycle consistency loss to better match the colors between generated results and the reference image.

2 Related work

Hair manipulation. Hair is a critical component of human portraits and yet is challenging to analyze and synthesize due to its intricate structure and severe self-occlusions therein. Various techniques have been proposed for hair manipulation, such as interactive hair shape editing [2], hair transfer [1], morphing [7], neural hair rendering [8, 9], and manipulation of multiple attributes of hair in a single image [3]. However, most of them are based on a coarse 2D orientation map to represent the hair structure, which lacks temporal coherence and fine-grained details, negatively affecting visual quality for video manipulation. Although Chai et al. [9] calculated a warping field to maintain temporal coherence, their method cannot generalize to arbitrary hairstyles due to the dependence on 3D hair models.

Conditional image generation. The unprecedented power of GANs [10] inspires assorted successful studies. For example, Refs. [11,12] designed unconditional GANs to generate super-realistic images based

on randomly sampled hidden vectors. To impose specific constraints within the generation process, Mirza and Osindero [13] proposed the conditional GAN (cGAN). Pix2Pix [14] further generalizes this idea to diverse conditional tasks, motivating many intriguing applications [15–18]. However, Pix2Pix and its variants do not support easy control of multiple attributes for a specific object during generation. To achieve this, the disentanglement of different attributes becomes extremely important, as demonstrated in the following studies. CariGANs [19] explicitly model geometric exaggeration and appearance stylization through two separate networks, achieving photo-to-caricature translation. iOrthoPredictor [20] decouples the teeth appearance and geometry in an unsupervised manner, enabling visual prediction of orthodontics. MichiGAN [3] utilizes multiple condition modules to control disparate attributes of the hair orthogonally. Our solution is inspired by MichiGAN, but is different from it in the following aspects. First, they focus on hair editing in images whereas we concentrate on image-to-video hair color transfer. Second, they represent hair structures using orientation maps [1, 8, 21], which get rid of textures completely but in turn filter out many details. In contrast, our approach utilizes normalized luminance maps, which are capable of preserving original fine-grained hair structures and illumination, and thus are more suitable for synthesizing realistic dynamic hair.

Video-to-video synthesis. Extensive studies have been conducted on video synthesis. Refs. [22–24] extended the GAN framework for unconditional video synthesis, achieving excellent results. However, these solutions have difficulty in generating long videos and cannot control specific video elements due to their unconditional settings. Subsequently, a lot of studies then focus on synthesizing videos under the control of a given specific sequence (e.g., semantic segmentation masks, body poses, or even images). Refs. [25, 26] designed the systems for motion transfer, where the motions of a character in a source video are transferred to a target character. Refs. [27, 28] transformed a low-resolution video into a realistic super-resolution video. Refs. [29–32] transferred the style of a reference image to a natural scene video. While these methods are subject to problem-specific constraints and designs, vid2vid [4] introduces a unified framework for video-to-video synthesis by imposing temporal inconsistency penalty. Analogously, our work can also be regarded as a video-to-video problem, and thus we use the framework of vid2vid to generate temporally coherent videos. Different from vid2vid, we introduce three delicate conditioning mechanisms within the video generation pipeline for the hair color transfer.

Video colorization. Most recently, a few studies [33–36] used one colored frame as an example and learned temporal propagation through deep neural networks to colorize the subsequent frames in a video sequence. However, due to the propagation mechanism, these methods would be problematic if it fails on a particular frame. Thus Refs. [5, 37, 38] proposed to establish the correspondences of objects between a reference image and a source image or video, based on which they could colorize the source in the LAB color space, namely transferring the chrominance (AB channels) of the reference to that of the source with the luminance (L channel) of the source kept intact. Despite high-fidelity results, they are prone to color inaccuracy and spuriousness since color information still more or less exists in the luminance map. Inspired by them, we extract a luminance map and further normalize it across frames to obliterate color information and maintain temporal coherence.

3 Overview

Given a source video with T frames $I(T) = \{I^1, I^2, \ldots, I^T\}$ and a reference image I_{ref}, our goal is to remove the original hair palette in $I(T)$ and then paint the reference's into it while keeping other hair attributes unchanged, e.g., shape, structure and illumination. The output video is denoted by $\hat{I}(T) = \{\hat{I}^1, \hat{I}^2, \ldots, \hat{I}^T\}$.

Figure 1 shows the pipeline of our system, where we transfer the hair color of a video frame by frame given a reference image. Our network architecture consists of a backbone generator and three condition modules to control three different components related to hair: background, color, and a collection of other attributes, including shape, structure, and illumination. The inputs to condition modules are respectively a background image B^t, the reference image I_{ref}, and k consecutive normalized luminance maps $I_l^t(k) = \{I_l^{t-k+1}, I_l^{t-k+2}, \ldots, I_l^t\}$ where $\{I_l^{t-k+1}, I_l^{t-k+2}, \ldots, I_l^{t-1}\}$ and I_l^t are calculated from the previous colorization results and current frame, respectively, which will be processed and injected into the backbone generator, producing the processed tth frame $\hat{I}^t$. By virtue of the disentanglement of disparate components, our backbone generator provides flexible control over the hair color in the video without affecting other components. More formally, we formulate the video generation process as the following

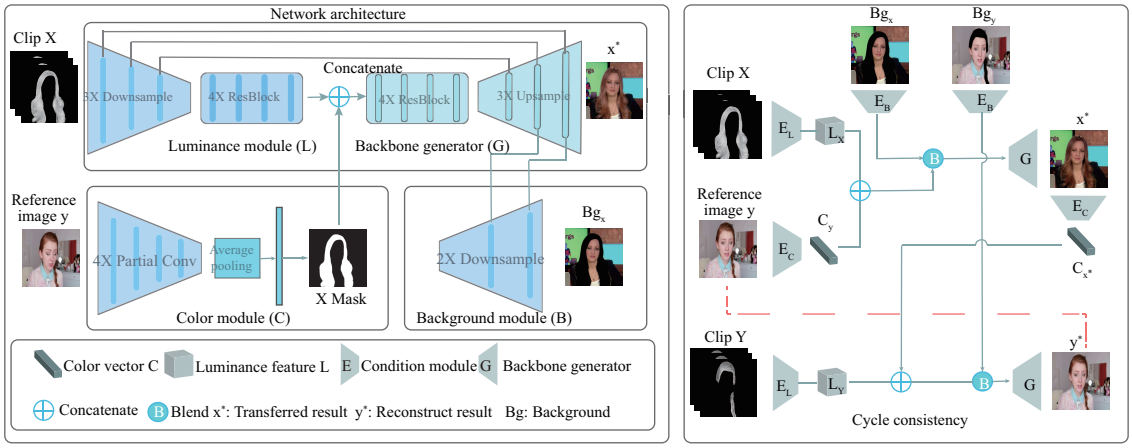

Figure 1 (Color online) The overall pipeline of iHairRecolorer. It consists of three condition modules and a backbone generator. The inputs to these condition modules are respectively a background image, a reference image, and 3 consecutive normalized luminance maps, which will be processed and injected into the backbone generator in a fully disentangled manner. A cycle consistency training strategy is also added to our pipeline to enforce the faithfulness of the generated results w.r.t the reference image.

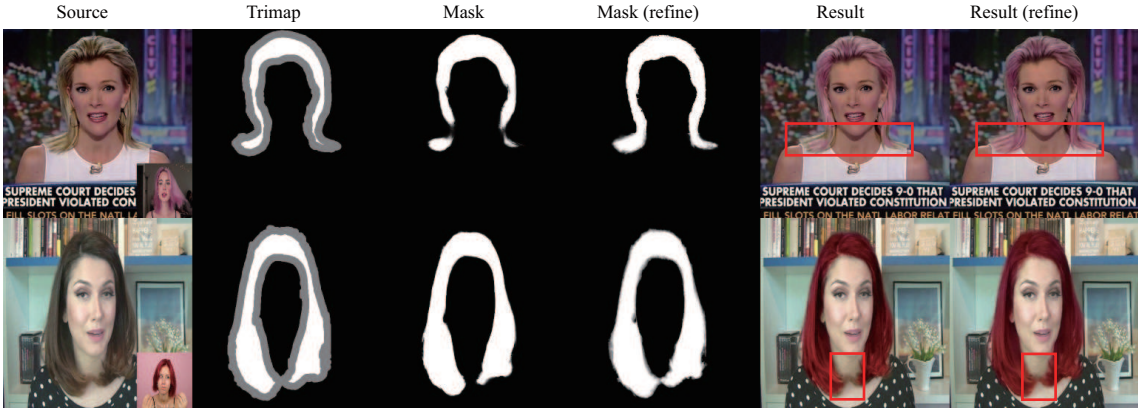

Figure 2 (Color online) We generate the trimap based on the coarse hair mask and impose the matting algorithm [40] to refine it, which can significantly improve the quality of recoloring.

deterministic function for each frame t:

$$\hat{I}^t = G\left(I_l^t(k), I_{\text{ref}}, B^t\right). \tag{1}$$

We will then elaborate on our method and prove its feasibility. We first introduce our three condition modules (Section 4). Then, we present the entire pipeline and discuss its training strategies and loss functions (Section 5). Finally, we show our method's effectiveness by experimental comparisons (Subsection 6.2) and ablation studies (Subsection 6.3).

4 Conditional modules

In this section, we will describe our three condition modules to handle disparate hair components in detail, including their condition representations, architectures, and integration into the backbone generator.

4.1 Luminance module

The luminance module is designed to process hair shape, structure and illumination, which are collectively represented using a map with the same resolution as the original image for each frame t.

Hair shape. We utilize a semantic segmentation network [39] to obtain a hair mask M for each frame, since we only focus on the hair region. However, the segmentation network often generates imperfect or even incorrect masks, as shown in the third column in Figure 2. Such masks often provide a coarse cue for

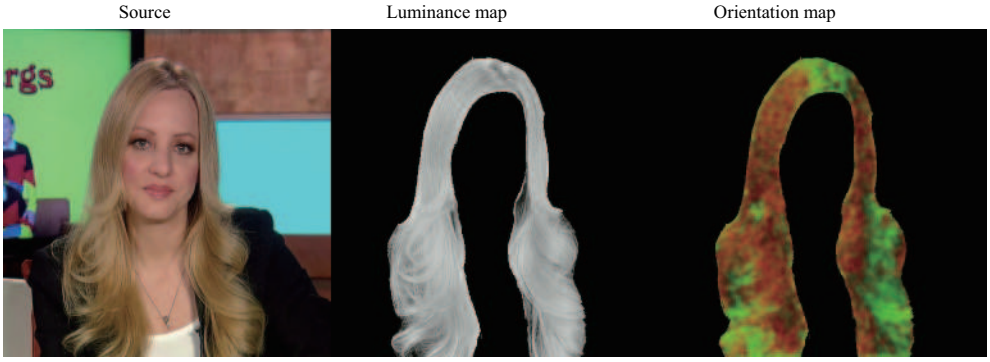

Figure 3 (Color online) The orientation map only retains local growing directions for a source hair image while our luminance map encompasses more detailed information, such as local structures and illumination.

the hair shape, thus leading to artifacts in the recoloring results, especially along the hair boundaries. To address this issue, we first erode the hair mask and then dilate it to generate a trimap (second column in Figure 2). White, black, and gray represent the foreground, background, and unknown area, respectively, where the unknown area means the boundary area between the hair and the background in our task. Finally, we impose the matting algorithm [40] to refine the hair mask (fourth column in Figure 2). It can be seen from Figure 2 that the refined masks lead to better recoloring results.

Hair structure & illumination. One possible way to represent hair structure is to use 2D hair orientation map as in [3], which gets rid of hair textures completely but in turn becomes less informative. For instance, varying hair structures and illumination will be relentlessly filtered out, becoming univocal 2D growing directions. Instead, we convert the image into the LAB color space consisting of a chrominance map (AB channels) and a luminance map (L channel). The luminance map is commonly used to represent the structures and illumination of scenes [5]. It has several advantages over a 2D hair orientation map. First, it is obtained directly from the color space conversion, and thus more stable and robust, whereas calculating 2D hair orientation maps is vulnerable to noise and image quality, leading to flickering videos as shown in Figure 3. Second, compared to an orientation map, which only contains the local hair growing directions, a luminance map also encompasses local details and lighting variations, which are instrumental in synthesizing stable and realistic dynamic hair.

However, there is more or less color information entangled in the luminance map, mainly because different luminances will lead to color fluctuation for the same chrominance, as shown in Subsection 6.2. Therefore, we cannot directly achieve color transformation within the LAB color space by replacing the chrominance of the source with that of the reference, as in [5].

In light of this, we normalize the luminance map to the standard normal distribution. In this way, the luminance map of any sample will have the same distribution, thus ensuring color consistency. However, we find that normalizing the luminance map frame by frame would cause the problem of temporal incoherence, since the pixel values for the same local feature might jump across frames after normalization. To avoid this problem, we thus normalize the the luminance map both spatially and temporally as follows:

$$\overline{L} = \frac{\sum_t^T \sum_i (M_i^t \cdot L_i^t)}{\sum_t^T \sum_i M_i^t}, \tag{2}$$

$$L_{\text{norm}}^t = \frac{L^t - \overline{L}}{\sqrt{\frac{\sum_t^T \sum_i (M_i^t \cdot (L_i^t - \overline{L})^2)}{\sum_t^T \sum_i M_i^t}}}, \tag{3}$$

where $L^t = I_l^t$ denotes the luminance map of I^t for simplicity and L_{norm}^t denotes the normalized version; M^t is the hair mask of the tth frame; the subscript i denotes the ith pixel in the map. Note that, the normalization happens only within a sampled video clip during training while during testing the entire video.

We remove the influence of background by $M \cdot L_{\text{norm}}$ and then concatenate k consecutive frames' as the module's input. Here, we use $k = 3$. In this way, when transferring each frame, we also include its previous 2 frames' normalized luminance maps to enforce temporal coherence. The luminance module consists of several downsampling and residual blocks, as well as the skip connections to the backbone generator.

4.2 Color module

Inspired by MichiGAN [3], we represent hair color as a feature vector extracted from the reference hair region for its global consistency and shape invariance. Different from MichiGAN, which needs to learn multiple factors, such as intrinsic albedo color, shading variations and wisp styles, we only learn the color style since other types of information have been provided by our luminance map.

Representing hair color as a global vector has several merits. First, this is a learnable high-level feature vector, which could hopefully extract the color palette and the mixing style, rather than a low-level hand-crafted vector, such as the averaged color of the hair region. Second, it can be designed to be invariant to hair shape and structure, and thus can be applied to any target hair region.

To achieve this, we first utilize several partial convolution [41] layers to extract color features only within the hair region and compress the feature map to a global vector by adding an instance-wise average pooling layer to compel the abandonment of hair shape, structure and illumination, similar to [3]. Then the output of color module serves as the latent input to our backbone generator. It is worth mentioning that we only use one reference image to chromatically guide the generation of the entire video. During training, we randomly choose a reference frame in a sampled video clip while during testing, we can use arbitrary hair images as reference images.

4.3 Background module

Finally, we must blend together the background and the synthesized hair regions to get a complete video. The simplest way is to achieve this in the image domain, via e.g., a direct copy and paste or Poisson blending [42], which however, would lead to either disharmony or discoloration. Therefore, we merge the hair region and background progressively in the feature domain to learn a more natural blending, similar to [3]. Different from MichiGAN, we do not randomly dilate or erode the background mask before encoding the background. What is more, we blend the background features only at the last two layers of the backbone generator. This is mainly due to that the diversity of hairstyles in our training set (about 400 videos) is much less than the training data used in [3] (about 56000 hair images), and the generator is liable to overfitting by synthesizing the hair color based on the background features if the blending begins at a deeper layer.

5 Backbone generator

In this section, we will describe our backbone generator, which integrates the information from the aforementioned conditional modules. We will introduce its architecture, training strategy, and loss functions (Subsections 5.1–5.3, respectively).

5.1 Architecture

Our backbone generator contains several residual blocks and up-sampling blocks. It integrates disparate hair features to generate the final results according to the following schemes. First, the color module encodes the reference image into a global vector and duplicates it to the down-sampled target hair region, producing the color latent input, which is further concatenated with the luminance module's output before fed into the generator. Then, after several residual blocks, the backbone generator will ingest multi-scale luminance features through skip connections from the luminance module, so as to obtain more detailed information. In parallel, we blend the background features into the last two up-sample layers of the generator to produce our result in a mask-aware manner [3]. Note that, unlike [3], which utilizes SPADE layers [43] to inject structure information into the generator, we directly adopt skip connections, since we did not find any significant improvement with SPADE layers.

5.2 Training strategy

Considering there is no paired ground-truth, we could let the background, color and other attributes come from the same source to train our model, i.e., in a self-supervised manner, similar to previous studies [3, 37]. However, this strategy does not have an explicit constraint for each part of the system, thus weakening the generalization ability of the model to diverse references. Therefore, we propose to further refine the trained model with a cycle-consistency loss. Specifically, after training our model under

self-supervision, we fix our color condition module and choose two different video clips: with one (clip X) as the source and a randomly chosen frame of the other (clip Y) as the reference, to obtain a recolored clip X*. Then, we utilize a random frame in X* as the reference and transfer its hair color to the hair in clip Y, and the model should reconstruct the reference video clip Y under the constraint of cycle consistency. As the cycle-consistency loss is imposed separately after the self-supervised training step, we refer the training stages with them as 'stage I' and 'stage II', respectively.

5.3 Loss functions

The goal of our network is to produce realistic dynamic hair whose color matches with the reference image with other attributes kept intact. For simplicity, we define $G(I^t, I_{\text{ref}}) = G(I_l^t(k), I_{\text{ref}}, B^t)$, representing the transformation of the hair color of frame t according to the reference image I_{ref} using our network G. In order to accomplish these objectives, we impose the following losses.

Reconstruction loss. For a self-supervised training strategy, our goal is to decompose different hair components of a sampled image I in a video clip and recombine them to reconstruct I. Thus, we use L_1 loss $\mathcal{L}_1$ and perceptual loss $\mathcal{L}_p$ to impose such penalty, as in [3].

Adversarial loss. We directly adopt two adversarial losses from [4], which are respectively adversarial loss $\mathcal{L}_{\text{adv}}$ for single images and spatio-temporal adversarial loss $\mathcal{L}_{\text{temp}}$ for video clips, to generate realistic and temporally coherent results, where $\mathcal{L}_{\text{temp}}$ is calculated similar to the first two items in (5). To achieve more robust training, we also take the discriminator feature matching loss $\mathcal{L}_{\text{FM}}$, as in [44].

Cycle consistency loss. As mentioned in Subsection 5.2, the above losses are not enough to make the network faithfully transfer the hair color. To better match the color of the reference I_{ref}, we propose a chromatic cycle consistency loss $\mathcal{L}_{\text{chromatic}}$, which can ensure that the color code of $\hat{I} = G(I, I_{\text{ref}})$ can reconstruct the reference image I_{ref}:

$$\begin{aligned}\mathcal{L}_{\text{chromatic}} = &\left\|I_{\text{ref}} - I_{\text{cyc}}\right\|_1 \\ &+ \left\|\phi_i\left(I_{\text{cyc}}\right) - \phi_i\left(I_{\text{ref}}\right)\right\|_1 \\ &+ \left\|\frac{\sum \hat{I} \cdot M}{\sum M} - \frac{\sum I_{\text{ref}} \cdot M_{\text{ref}}}{\sum M}\right\|_1,\end{aligned} \tag{4}$$

where $I_{\text{cyc}} = G(I_{\text{ref}}, \hat{I})$, ϕ_i represents the ith layer of the VGG network [45]. The first and second items measure the difference between I_{ref} and I_{cyc} while the third item ensures that the averaged color of $\hat{I}$ matches that of I_{ref}, which collectively ameliorates the situation of discoloration. What is more, we propose $\mathcal{L}_{\text{stable}}$ to enhance temporal coherence of the synthesized video clips at this stage, using the same spatiotemporal discriminator D as in stage I:

$$\begin{aligned}\mathcal{L}_{\text{stable}} = &\mathbb{E}\left[D\left(\hat{I}^t(k), I_l^t(k)\right)\right]^2 \\ &+ \mathbb{E}\left[\left(1 - D\left(I^t(k), I_l^t(k)\right)\right]^2\right. \\ &+ \mathbb{E}\left[D\left(I_{\text{cyc}}^t(k), I_{\text{ref},l}^t(k)\right)\right]^2 \\ &+ \mathbb{E}\left[\left(1 - D\left(I_{\text{ref}}^t(k), I_{\text{ref},l}^t(k)\right)\right]^2\right.,\end{aligned} \tag{5}$$

where k represents k $(= 3)$ consecutive frames in the video. We also skip 3 frames to take longer temporal coherence into consideration. In summary, the overall training objective we aim to optimize is

$$\begin{aligned}\mathcal{L}_{\text{all}} = &\lambda_1\mathcal{L}_1 + \lambda_p\mathcal{L}_p + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}} \\ &+ \lambda_{\text{temp}}\mathcal{L}_{\text{temp}} + \lambda_{\text{chromatic}}\mathcal{L}_{\text{chromatic}} + \lambda_{\text{stable}}\mathcal{L}_{\text{stable}},\end{aligned} \tag{6}$$

where λ_1, λ_p, λ_{adv}, λ_{FM}, λ_{temp}, $\lambda_{\text{choromatic}}$, and λ_{stable} control the weight of each loss, and we set $\lambda_1 = 10.0$, $\lambda_p = 10.0$, $\lambda_{\text{adv}} = 0.1$, $\lambda_{\text{FM}} = 0.1$, $\lambda_{\text{temp}} = 1.0$, $\lambda_{\text{choromatic}} = 10.0$, $\lambda_{\text{stable}} = 1.0$ in our experiments.

6 Experiments

In this section, we first show diverse results generated by our method and then introduce the implementation details of iHairRecolorer (Subsection 6.1). Finally we verify the effectiveness of our system

through visual and comparative experiments (Subsection 6.2), along with studies on the necessity of each algorithmic component (Subsection 6.3).

With varicolored reference images, we tested our method on the source videos with diverse hairstyles ranging from short to long, simple to complex, where hairs undergo different types of illumination and motions, induced either by head movements or wind blow. In general, our method can realistically and robustly transfer the hair color in each video, which matches well with the given reference images. By virtue of our normalized luminance map, our method can recover not only globally consistent illumination but also realistic motions for the dynamic hair, even under large motions and complex hair illumination, as seen in the 4th, 8th, and 9th rows. Thanks to our proposed cycle consistency constraint and the normalization of luminance map, our generator can faithfully transfer the hair color, as revealed in the color consistency of the results in the same column. All video results can be found in our supplementary materials.

In summary, our method has the following advantages. First, since the normalized luminance map contains rich and stable structure information, it works fine for diverse hairstyles and even under large hair motions. Second, our method can ignore the illumination of the reference and keep the illumination of the original video unchanged, due to the disentanglement of illumination from color. Third, the diverse hairstyles can be transferred to the consistent color using the same reference, attributed to our cycle consistency loss and normalized luminance map.

6.1 Implementation details

Dataset. In order to cover a wide range of hair structure and appearance, we train our model on public dataset FaceForensics [6] and our newly collected dataset from YouTube. The videos in these datasets are of high quality and contain diverse hair colors and structures. We crop and align them to 256×256 through a face detector. Finally, we get 407 videos, of which 370 are for training and the rest for testing. Besides, we also augment the data by randomly flipping or rewinding the videos. For each video, we randomly sample 50 clips, at the beginning of training: we only used 7 consecutive frames for training in each clip and doubled the number of frames after every 3 epochs.

Network structure. Now, we introduce the structures of iHairRecolorer, as shown in Figure 1. The luminance module contains 3 consecutive downsampling and four residual blocks with the same kernel size of 3×3 and finally increasing the feature channels to 512. The color module downsamples the reference image by 4 downsampling partial convolutions with an instance normalization and leaky ReLu activation to avoid the influence of other samples in the same batch. It also involves a global instance average pooling to compress the color features to a global vector. Then the mask-guided block duplicates the color code spatially to the downsampled target hair mask. For the background module, we only design two downsampling layers since the background is easy to learn, and as the receptive field increases, it also increases the risk of boundary overfitting during self-supervised training. Similar to the luminance module, the backbone generator contains 4 residual blocks and 3 upsampling blocks, and accepts all the condition features to generate final results. Besides, we also design skip connections between the luminance module and the backbone generator to obtain more detailed information.

6.2 Comparisons

To our best knowledge, we are the first deep learning-based approach for image-to-video hair color transfer. Thus, we compare our approach with the state-of-the-art methods, which could achieve similar results, including vid2vid [4], MichiGAN [3], and video colorization [5]. In addition, we also compare the performance of different hair structure representation methods (luminance map and orientation map) when coloring a static image.

Comparison with video-to-video synthesis. Firstly, we compare to vid2vid. Since their off-the-shell framework could not support color condition generation, we compare with them by producing our results based on the references sampled from their generated results. Figure 4 shows the comparison results. Despite excellent temporal coherence in their results, their approach is susceptible to image quality, since they must transform the original video into a series of edge maps as input. What is more, since they generate the results in a recurrent manner, structural distortion or color inconsistency across frames can be observed due to error accumulation, especially as the number of generated frames increases. In contrast, our method retains complete structural information and guarantees color consistency no matter how long the video is, and keeps the rest regions unchanged.

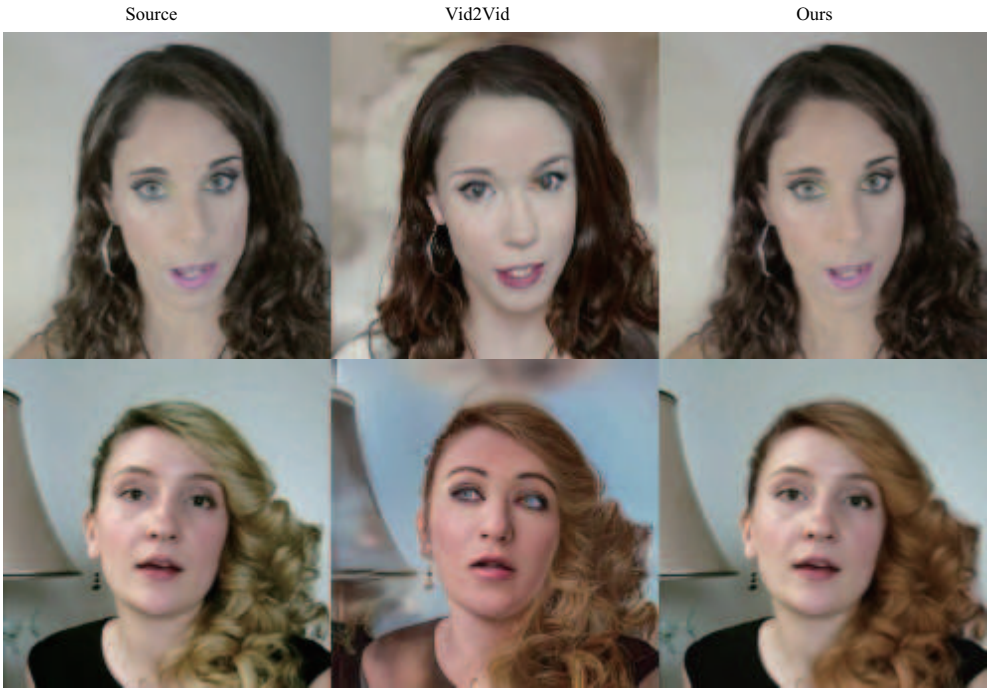

Figure 4 (Color online) Due to error accumulation, vid2vid [4] is prone to color inconsistency and structure distortion. In contrast, our results are more consistent, realistic and temporally coherent.

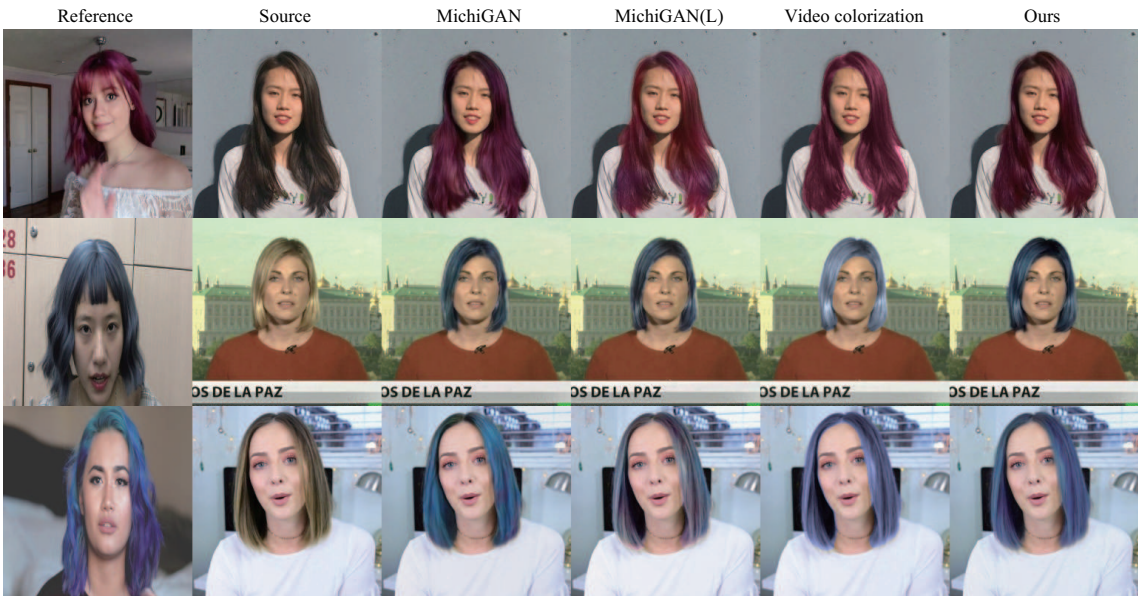

Figure 5 (Color online) Comparison with state-of-the-art methods. Due to a lack of temporal coherence, the results generated by MichiGAN [3] and MichiGAN(L) are flickering. On the other hand, the results generated by video colorization [5] cannot match the color in the reference image.

Comparison with MichiGAN. As shown in the first row in Figure 5, the videos generated by MichiGAN suffer from the problem of serious flickering artifacts and unrealistic visual effects, when the hair structure of the source is intricate or the hair undergoes large motions. A major reason is because their method generates images frame-by-frame and does not guarantee the temporal coherence. Besides, since their method relies on coarse 2D orientation maps and the extraction of such maps also does not ensure a temporal coherence, their method will lead to flickering artifacts when extended to videos. On the other hand, since their method does not explicitly disentangle the luminance from hair color, the illumination from the reference is transferred to an arbitrary position of the hair in consecutive

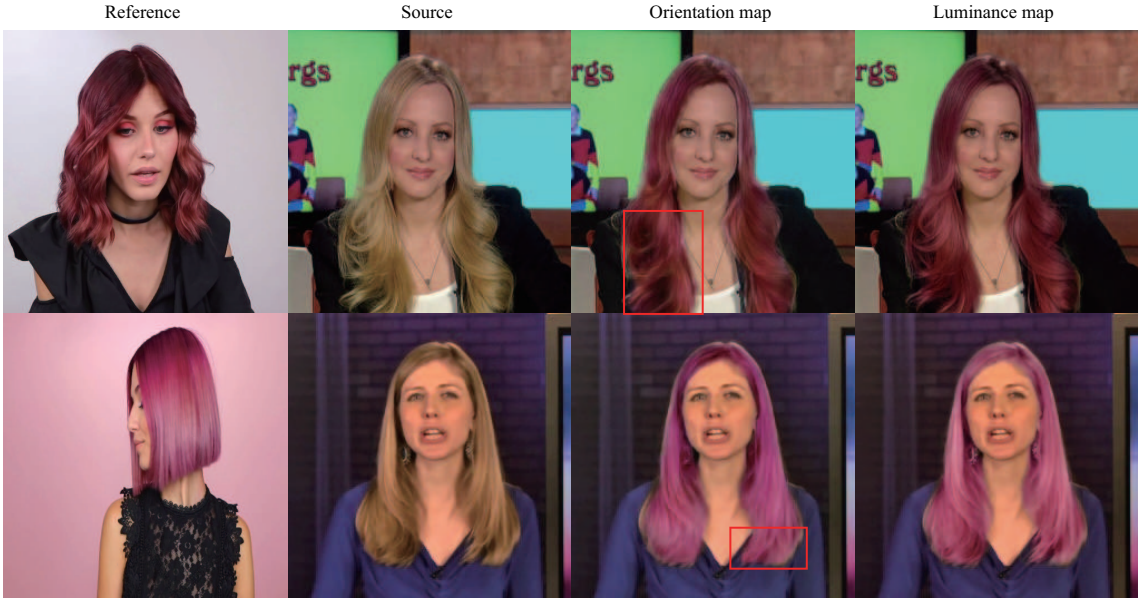

Figure 6 (Color online) Compared with orientation map, the luminance map can retain more fine-scale structural features.

sequences, resulting in a negative effect on the authenticity of their results, as seen in the second row of Figure 5. Moreover, since they only adopt the self-supervised strategy to train the model, it is difficult for their method to generalize to diverse hairstyles or reference colors, especially mixed colors as shown in the third row of Figure 5. Furthermore, we also replace the orientation map with the luminance map (denoted by MichiGAN(L)) for comprehensive comparison. As shown in the fourth column, the luminance map significantly improves the stability of the hair structure and retains more details, as we mentioned in Subsection 4.1. However, the color of the transferred results lacks temporal coherence and changes with the movement of the hair. In contrast, our method can adapt to diverse extreme conditions such as large motion, complex illumination, and mixed colors, and generate realistic, high-fidelity and temporally-coherent video sequences, as well as being not affected by the reference's illumination while retaining the illumination in the source video.

Comparison with video colorization. Another work similar to ours is video colorization [5], which colorizes a grayscale video given a reference image. They design a network to find the semantic correspondence between the reference image and the source video, based on which they replace the AB channels of the source with that of the reference. Considering that their method is a generic colorization method, we did not fine-tune their network on portrait dataset. However, for a fair comparison, we directly assign the semantic corresponding relationship (i.e., between the hair regions) to their network to remove the influence of other parts on the hair color and only compare with their approach in terms of the coloring performance and authenticity. As shown in the fourth column in Figure 5, the hair color transferred by their method is significantly different from the reference image because directly duplicating the reference's AB channel to the source video will be affected by the discrepancy of luminance distribution. In addition, their method cannot refine the boundary if the correspondence relationship is not completely correct, and it will color the background and lead to boundary artifacts as seen in the first row of Figure 5. In contrast, we design the color module to extract global color features and fuse with structural (luminance map) features at the feature level instead of directly swapping AB channels, thus making the results more faithful to the reference image. Moreover, we proposed the normalized luminance map and cycle consist loss further to improve the color accuracy and authenticity of the generated results. Therefore, our method leads to better recoloring results.

Performance on static images. As shown in Figure 6, when coloring static images, the luminance map can better maintain fine-scale structure of source image than orientation map, because the orientation map is calculated by Gabor filtering and Gaussian blurring, which describes the approximate growth direction of the hair, and fine-scale features are vulnerable to noise and easily lost. The results are similar to coloring videos as describes in Subsection 6.3.

User study. We conducted a simple user study to further evaluate the realism of our generated

Table 1 Compare the quantitative results and human preference scores (user study) of the results generated by different methods

Method	MichiGAN	MichiGAN(L)	Video colorization	Ours
FID	23.73	19.94	27.07	16.73
Human preference (%)	11.46	20.55	15.41	52.58

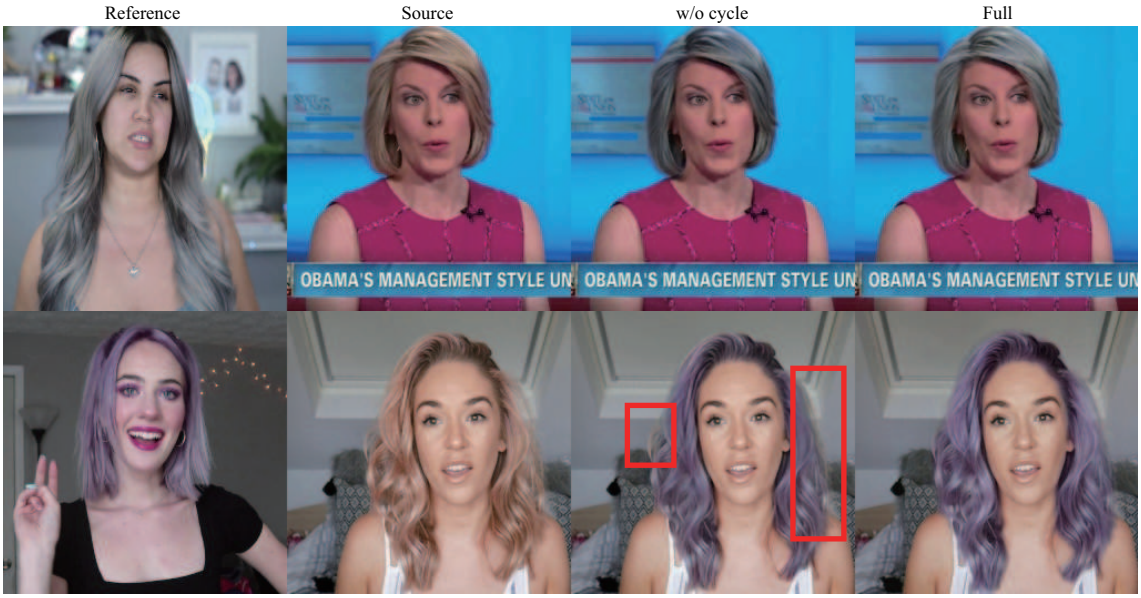

Figure 7 (Color online) Our cycle consistency loss helps the network to generate the hair color that better matches with the reference image, compared to the network trained without it (w/o cycle). Additionally, it also refines the hair boundaries.

results. We used 25 videos randomly selected from the test dataset and colored them by different reference images. Users are required to choose the best results generated by different methods in terms of temporal consistency, visual photo-realism, and color faithfulness. Note that vid2vid cannot generate arbitrary hair colors because the models they provide do not support conditional modules, so we exclude this method in the study. Table 1 shows the results based on the feedback from 28 users, 52.58% of the users think our results are the best.

Quantitative comparisons. We also employ the Frechet inception distance (FID) to measure the distribute distance between the colorized output and the realistic natural frames. We first use a variety of reference images to color each test video, and then randomly sample 10000 frames from all generated videos and their corresponding source images to calculate the FID value. As shown in Table 1, our method achieves the lowest FID, proving that our method provides the most realistic results. Besides, MichiGAN(L) achieved a lower FID value than MichiGAN, which again shows the effectiveness of our proposed normalized luminance map.

6.3 Ablation study

In this subsection, we show the effectiveness of each component in our system.

Cycle consistency loss. To validate the proposed cycle consistency loss $\mathcal{L}_{\text{Cyc}}$ ($\mathcal{L}_{\text{chromatic}}$ and $\mathcal{L}_{\text{stable}}$), we compare our full model with the simplified version which is trained without this term (denoted by NC). The results shown in Figure 7 reveal two advantages of $\mathcal{L}_{\text{Cyc}}$. First, the full model transfers the hair color from the references to the videos more faithfully than NC. This is mainly because there is no such explicit constraint in the training process of NC, making it generalize worse to diverse references. By adding $\mathcal{L}_{\text{Cyc}}$, the network is forced to reconstruct the reference using the color code of the transferred source, thus ensuring its faithfulness to the reference. Second, $\mathcal{L}_{\text{Cyc}}$ could more or less alleviate the hair boundary flickering problem caused by the imperfect hair mask. When training NC, the self-reconstruction training manner makes this problem covert. In contrast, after adding $\mathcal{L}_{\text{Cyc}}$, the color disharmony around the hair boundaries of the cross-transferred results becomes overt and can be discriminated by the spatiotemporal discriminator.

Spatially and temporally normalized luminance map. To demonstrate the importance of our

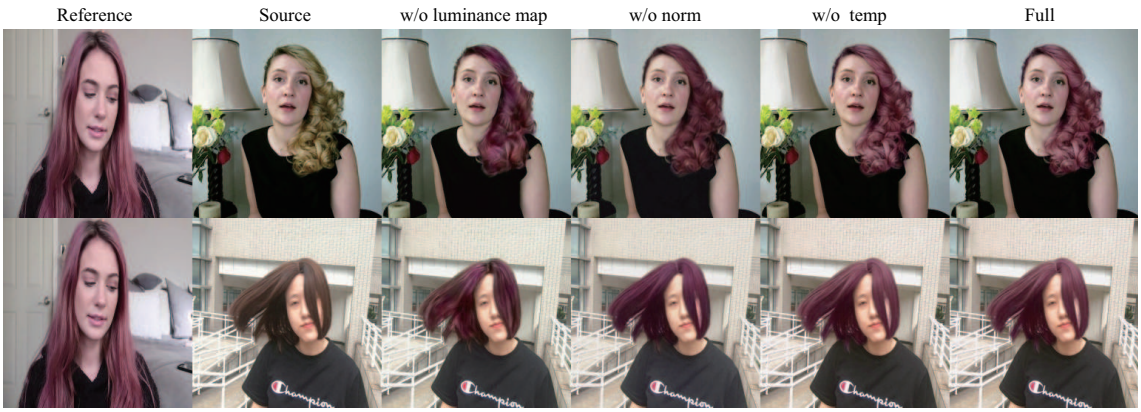

Figure 8 (Color online) The results generated using the orientation map (w/o luminance map) are not temporally coherent and realistic enough. When we replace the orientation map with the regular luminace map (w/o norm), those problems are alleviated a lot but the problem of color inconsistency with the reference occurs. After normalizing the regular luminance map spatially (w/o temp), the transferred color matches the reference better. Moreover, a temporally and spatially normalized luminance map can generate more temporally coherent dynamic hair (Full).

Figure 9 (Color online) Ablation study for background module. Obviously, directly blending the background in the image domain (w/o background) will produce serious boundary artifacts.

spatially and temporally normalized luminance map, we conduct four sets of experiments: (i) replacing luminance map with orientation map (w/o luminance map), (ii) not normalizing the luminance map (w/o norm), (iii) normalizing luminance map only spatially (w/o temp), and (iv) normalizing the luminance map spatially and temporally (Full). As shown in Figure 8, the orientation map contains only the growth direction of the hair, which is coarse structure information, and the extracted structure may lack temporal coherence in consecutive frames, especially under large hair motions or when the hair structure is intricate (see the results in the 3rd column). At the fourth column of Figure 8, we show that without normalizing the luminance map, different source videos are transferred into different colors under the same color condition due to the discrepancy of luminance distributions. On the other hand, only normalizing the luminance map spatially would still cause flickering of the luminance maps acorss frames and thus flickering artifacts in the transferred results, when the hair undergoes large motions (the 5th column in the second row). In comparison, the spatially and temporally normalized luminance map ensures a much more faithful transfer of the hair colors.

Blend the background based on the mask. To verify the necessity of the background module, we compare with the following alternative: only generating the hair region and blending it with the background based on the hair mask (w/o background). As shown in the middle-right of Figure 9, the mix of background and hair produces serious boundary artifacts. Similar phenomenon can also be observed in a single frame.

7 Limitations

Although our method can achieve compelling results, it also has several limitations. First, due to different hairstyles between the reference image and the source video, we compress the color features into a global latent vector and transfer the source video based on the hair mask. Therefore, it produces unreal videos

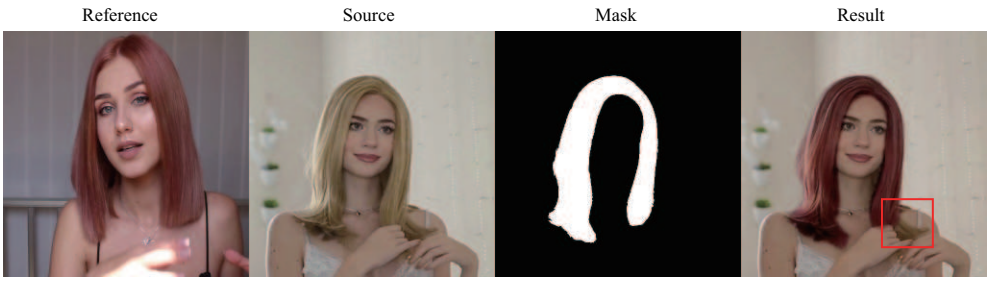

Figure 10 (Color online) The limitation of iHairRecolorer. When there is a large incorrect region produced by hair segmentation network, the corresponding hair color will not be transferred.

Figure 11 (Color online) Diverse results generated by our method. It can be seen that our method could generate realistic and temporally-coherent dynamic hair whose color closely matches with the reference images.

when the hair segmentation network produces incorrect masks, especially when there is a large region of incorrect segmentation as shown in Figure 10. On the other hand, imperfection and temporally inconsistent hair segmentation, especially around the hair boundary regions, may also lead to suspicious

artifacts when the hair undergoes large motions (e.g., the fringe area in the 9th row of Figure 11). Moreover, the hair segmentation network we employ is mainly trained on front-facing photos, and thus may not extend well for non-frontal photographs. Developing a more robust and temporally accurate hair parsing method could largely alleviate the aforementioned problems. Second, our method currently only focuses on the hair color transfer, and it cannot manipulate other hair attributes in the video, such as hair style or hair structure. Editing dynamic hairstyles in a video or even in 3D [46] is an interesting but challenging task to be explored in the future. This demands anticipating faithful rigid head motions and non-rigid hair motions.

8 Conclusion

To conclude, we have introduced iHairRecolorer, the first deep generative adversarial network for image-to-video hair color transfer. It contains three meticulous design condition modules and a backbone generator. Different from existing conditional hair editing methods, our approach uses normalized luminance maps instead of orientation maps to represent hair structure and illumination. Under the constraint of our proposed cycle consistency loss, our method faithfully transfers the color from the reference image to the source video. Our ablation study proves the importance of each component of our proposed method and our comparison experiments demonstrate that the proposed method significantly performs existing alternative approaches on transferring video hair color.

Acknowledgements This work was supported in part by National Key Research & Development Program of China (Grant No. 2018YFE0100900) and National Natural Science Foundation of China (Grant No. 62172363).

Supporting information The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chai M, Wang L, Weng Y, et al. Single-view hair modeling for portrait manipulation. ACM Trans Graph, 2012, 31: 1–8
2 Chai M, Wang L, Weng Y, et al. Dynamic hair manipulation in images and videos. ACM Trans Graph, 2013, 32: 1–8
3 Tan Z, Chai M, Chen D, et al. MichiGAN: multi-input-conditioned hair image generation for portrait editing. 2020. ArXiv:2010.16417
4 Wang T C, Liu M Y, Zhu J Y, et al. Video-to-video synthesis. 2018. ArXiv:1808.06601
5 Zhang B, He M, Liao J, et al. Deep exemplar-based video colorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 8052–8061
6 Rossler A, Cozzolino D, Verdoliva L, et al. Faceforensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 1–11
7 Weng Y, Wang L, Li X, et al. Hair interpolation for portrait morphing. In: Proceedings of Computer Graphics Forum, 2013. 79–84
8 Wei L, Hu L, Kim V, et al. Real-time hair rendering using sequential adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 99–116
9 Chai M, Ren J, Tulyakov S. Neural hair rendering. 2020. ArXiv:2004.13297
10 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014. 2672–2680
11 Zhang H, Goodfellow I, Metaxas D, et al. Self-attention generative adversarial networks. In: Proceedings of International Conference on Machine Learning, 2019. 7354–7363
12 Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 4401–4410
13 Mirza M, Osindero S. Conditional generative adversarial nets. 2014. ArXiv:1411.1784
14 Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017
15 Wang T C, Liu M Y, Zhu J Y, et al. High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018
16 Jo Y, Park J. SC-FEGAN: face editing generative adversarial network with user's sketch and color. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 1745–1753
17 Yu R, Wang X, Xie X. VTNFP: an image-based virtual try-on network with body and clothing feature preservation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 10511–10520
18 Han X, Wu Z, Wu Z, et al. VITON: an image-based virtual try-on network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7543–7552
19 Cao K, Liao J, Yuan L. CariGANs: unpaired photo-to-caricature translation. 2018. ArXiv:1811.00222
20 Yang L, Shi Z, Wu Y, et al. iOrthoPredictor: model-guided deep prediction of teeth alignment. ACM Trans Graph, 2020, 39: 1–15
21 Paris S, Briceño H M, Sillion F X. Capture of hair geometry from multiple images. ACM Trans Graph, 2004, 23: 712–719
22 Saito M, Matsumoto E, Saito S. Temporal generative adversarial nets with singular value clipping. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 2830–2839
23 Tulyakov S, Liu M Y, Yang X, et al. MoCoGAN: decomposing motion and content for video generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1526–1535

24 Vondrick C, Pirsiavash H, Torralba A. Generating videos with scene dynamics. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 613–621
25 Liu W, Piao Z, Min J, et al. Liquid warping GAN: a unified framework for human motion imitation, appearance transfer and novel view synthesis. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 5904–5913
26 Chan C, Ginosar S, Zhou T, et al. Everybody dance now. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 5933–5942
27 Shechtman E, Caspi Y, Irani M. Space-time super-resolution. IEEE Trans Pattern Anal Machine Intell, 2005, 27: 531–545
28 Shi W, Caballero J, Huszár F, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 1874–1883
29 Chen D, Liao J, Yuan L, et al. Coherent online video style transfer. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 1105–1114
30 Gupta A, Johnson J, Alahi A, et al. Characterizing and improving stability in neural style transfer. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 4067–4076
31 Huang H, Wang H, Luo W, et al. Real-time neural style transfer for videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 783–791
32 Ruder M, Dosovitskiy A, Brox T. Artistic style transfer for videos. In: Proceedings of German Conference on Pattern Recognition. Berlin: Springer, 2016. 26–36
33 Jampani V, Gadde R, Gehler P V. Video propagation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 451–461
34 Vondrick C, Shrivastava A, Fathi A, et al. Tracking emerges by colorizing videos. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 391–408
35 Liu S, Zhong G, de Mello S, et al. Switchable temporal propagation network. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 87–102
36 Meyer S, Cornillère V, Djelouah A, et al. Deep video color propagation. 2018. ArXiv:1808.03232
37 He M, Chen D, Liao J, et al. Deep exemplar-based colorization. ACM Trans Graph, 2018, 37: 1–16
38 He M, Liao J, Chen D, et al. Progressive color transfer with dense semantic correspondences. ACM Trans Graph, 2019, 38: 1–18
39 Chai M, Shao T, Wu H, et al. AutoHair: fully automatic hair modeling from a single image. ACM Trans Graph, 2016, 35: 1–12
40 Hou Q, Liu F. Context-aware image matting for simultaneous foreground and alpha estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 4130–4139
41 Liu G, Reda F A, Shih K J, et al. Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 85–100
42 Pérez P, Gangnet M, Blake A. Poisson image editing. ACM Trans Graph, 2003, 22: 313–318
43 Park T, Liu M Y, Wang T C, et al. Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019
44 Wang T C, Liu M Y, Zhu J Y, et al. High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 8798–8807
45 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556
46 Guo J, Li M, Zong Z, et al. Volumetric appearance stylization with stylizing kernel prediction network. ACM Trans Graph, 2021, 40: 1–15