Computer-Aided Design 115 (2019) 111–121

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Vectorization Based Color Transfer for Portrait Images☆,☆☆

Qian Fu [a], Ying He [a,*], Fei Hou [b], Juyong Zhang [c], Anxiang Zeng [d,e], Yong-Jin Liu [f]

[a] School of Computer Science & Engineering, Nanyang Technological University, Singapore
[b] Institute of Software, Chinese Academy of Sciences, China
[c] School of Mathematical Sciences, University of Science and Technology of China, China
[d] Alibaba Group, China
[e] NTU-Alibaba Joint Research Institute, Singapore
[f] MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, China

ARTICLE INFO

Article history:
Received 26 April 2019
Accepted 3 May 2019

Keywords:
Color transfer
Portrait images
Optimal mass transportation
Diffusion curves
Poisson vector graphics

ABSTRACT

This paper introduces a method for transferring colors between portrait images. Using a trained neural network to extract facial mask, we vectorize each image with a set of sparse diffusion curves to encode the low-frequency colors, and use the Laplacian of residual colors to represent the high-frequency details. Then we apply optimal mass transport to transfer the boundary colors between the diffusion curves of the source and reference images. Finally, the original or modified Laplacians of colors are added to the transferred diffusion curve image. Unlike the existing methods that either require 3D information or assume the source and reference images have similar poses and dense correspondence, our method is computationally efficient and flexible, which can work for portrait images with large pose and color differences.

1. Introduction

In the digital age, portrait and self-portrait photographs are extremely popular and have spread to every corner of the world. Although these images are casual in nature, many users would like to retouch them before sharing online. Since manual color tuning is time consuming even for experienced users, color transfer is a practical method to change the appearance of the source image to the color pattern of a reference image.

In computer graphics and image processing fields, color transfer has been studied for almost two decades since the seminal paper by Reinhard et al. [1]. There are two classes of algorithms for automatic color transfer: the traditional statistic-based methods that match the color distributions of the source and references images [2], and more recently the learning-based methods that can establish semantically-meaningful correspondence between visually different image pairs [3,4]. Although these methods work very well for landscape photos, directly applying them to portrait images often produce poor results, since people are more sensitive to the artifacts of human faces than landscape photos. To tackle the challenges, Shih et al. [5] decomposed portrait images into multiple layers of varying scales, and transferred the local statistics of each layer by computing a local energy map. Matching the local statistics over multiple scales can effectively transfer the skin textures, the local contrast and the overall lighting direction. However, due to the requirement of a dense correspondence between the input and the reference, it works only for the pair with similar poses and facial expressions. Recently, Shu et al. [6] developed an algorithm for transferring the illumination of one portrait to another. Fitting a 3D morphable face model to the portrait, their method combines the color values with 3D positions and normals, and transfers the color and illumination by solving an optimal mass transport (OMT) problem. Although their method is able to handle a variety of portraits and illumination conditions, it is computationally expensive due to the formulation in 8-dimensional space.

In this paper, we aim at developing a simple yet effective method for transferring colors between portrait images. Using a trained neural network to extract facial mask, we first vectorize each image using a set of sparse diffusion curves to encode the low-frequency colors, and use the Laplacian of residual colors to represent high-frequency details. Then we apply optimal mass transport to transfer the boundary colors between the diffusion curves of the source and reference images. Finally, the original or

☆ This paper has been recommended for acceptance by Pierre Alliez, Yong-Jin Liu and Xin Li.

☆☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cad.2019.05.005.

* Corresponding author.
E-mail addresses: qfu004@e.ntu.edu.sg (Q. Fu), YHe@ntu.edu.sg (Y. He), houfei@ios.ac.cn (F. Hou), juyong@ustc.edu.cn (J. Zhang), renzhong@taobao.com (A. Zeng), liuyongjin@tsinghua.edu.cn (Y.-J. Liu).

https://doi.org/10.1016/j.cad.2019.05.005
0010-4485/

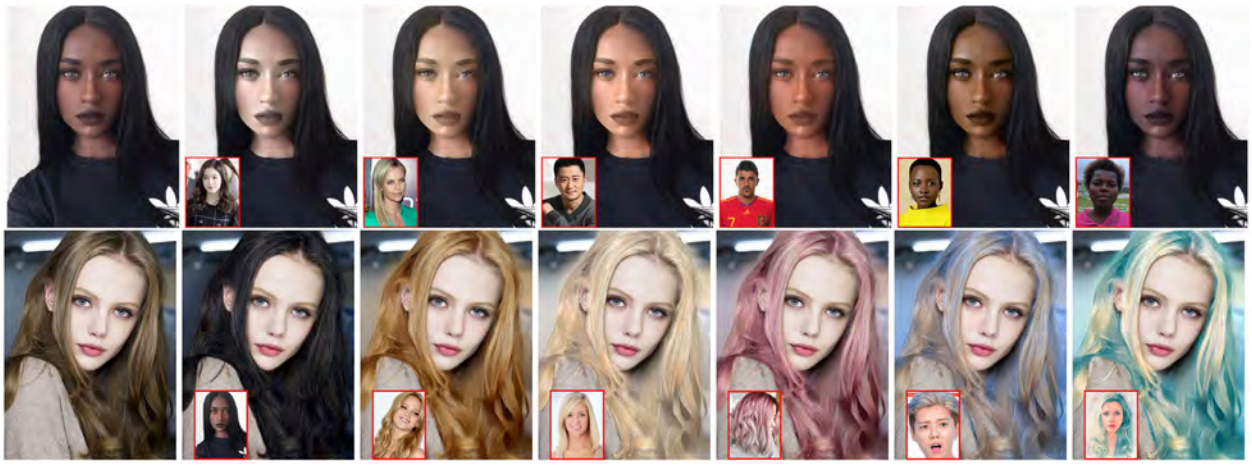

Fig. 1. Examples of skin and hair color transfer with our method. The original images are shown in the first column and the reference images are shown in small insets. Note that our method does not require a dense correspondence between the input and the reference, therefore it can handle portraits with different postures.

modified Laplacians of colors are added to the transferred diffusion curve image. Using a local Poisson solver, the color changes can be updated in a local manner. Unlike the existing methods that either require 3D information or assume the source and reference images have similar poses and dense correspondence, our method is computationally efficient and flexible, which can work for portrait images with large pose and color differences. Since our method formulates OMT using sparse diffusion curves, it is computationally efficient, and does not require a dense correspondence between the input and the example. As a result, our method is able to handle portraits of different postures (including side views) and special facial textures (e.g., freckle, tattoo), and work for both photographs and paintings. Our method also allows the user to adjust PRs to produce interesting visual effects such as watercolor or painting styles. Fig. 1 shows examples of skin and hair color transfer using our method.

2. Related work

This section reviews the closely related work on color transfer, vector graphics, and image vectorization.

2.1. Vector graphics

Vector graphics provides several practical benefits over traditional raster graphics, including sparse representation, compact storage, geometric editability, and resolution-independence. Early vector graphics supports only linear or radial color gradients, diminishing their applications for photo-realistic images. Orzan et al. [7] proposed diffusion curve images (DCI) that consist of sparse curves with user-specified colors on both sides. By solving a Laplace equation, the colors are smoothly diffused from the curves across the image domain. Bezerra et al. [8] proposed diffusion barriers, diffusion anisotropy, and spatially varying color intensity to control the diffusion process. Their method can diffuse both colors and normal maps to produce non-realistic effects. Finch et al. [9] extended diffusion curves to provide smooth interpolation through color constraints by using thin-plate splines, which allows direct control of color gradient. Lieng et al. [10] developed shading curves that associate shading profiles to each side of the curve. These shading profiles, which can be manually manipulated, represent the color gradient out from their associated curves. Recently, Hou et al. [11] presented Poisson vector graphics (PVG) which extends DCI with non-zero Laplacians. Together with two new types of vector primitives, called Poisson regions and Poisson curves, they demonstrated that PVG can effectively produce photo-realistic images.

2.2. Image vectorization

There are mainly two kinds of vector image formulations, region based and curve based vector graphics. In terms of region based vector formulation, Sun et al. [12] introduced an automatic vectorization algorithm of gradient mesh using Ferguson patches. Later, Lai et al. [13] generalized the method for constructing high-quality gradient meshes for multiply-connected domains using slit map [14]. In terms of curve based formulation, Xie et al. [15] proposed hierarchical diffusion curve to vectorize natural images. Zhao et al. [16] proposed to use shape optimization to generate more compact vector representation of images. With the development of deep learning, Song et al. [17] proposed a texture compression algorithm that represented the image by a deep neural network.

2.3. Color/style transfer

Reinhard et al. [1] discussed that RGB channels are not independent of each other but LAB channels are, so they used LAB three channels to color transfer for natural images based on a linear equation for mean and standard deviation. Pitie [2] presented N-dimensional probability density function for color transfer. Hwang et al. [18] applied moving least squares to color transfer and image denoising in a same scene because of camera setting or time reason. Lee et al. [19] provided an example-based color transfer by using regularized color and tone mapping functions and an illuminance correction step to reduce artifacts. Shih et al. [5] proposed a color transfer for headshots that can match skin texture, local contrast and overall lighting direction by matching multiple-dimensional local statistics, but required a dense correspondence and similar beard, hairstyle, and age of the input and reference. Shu et al. [6] applied massing transport approach over 8D data of facial regions including RGB color, position, and normal to achieve a good relighting and color transfer result. There are several other methods based on object-to-object color transfer with semantically correspondences between source and reference images. Wu et al. [20] proposed a content-based method for separately transferring the color patterns between the corresponding semantic regions which extracted by a composition-based detection method or a face detection method for portraits. Yang et al. [21] presented a semantic color transfer especially for portraits, that they first manually extracted semantic facial parts and then perform color transfer algorithm proposed by [1] between corresponding parts with the same semantic information. He et al. [22] used neural representations

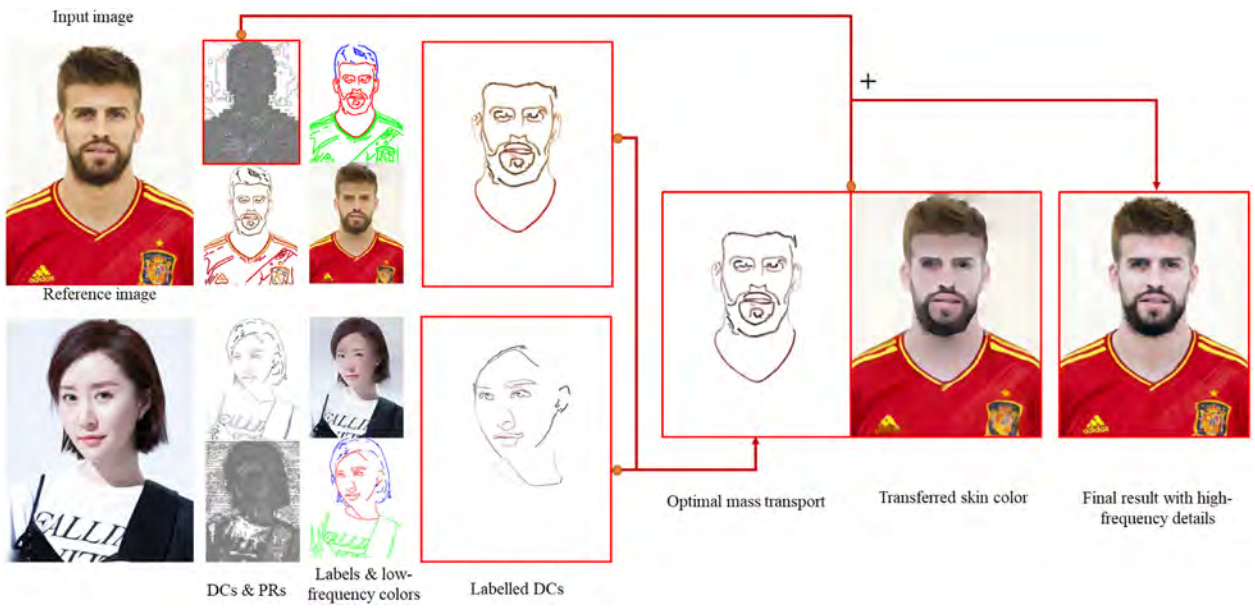

Fig. 2. Our algorithmic pipeline. Vectorizing the input and reference images, we obtain a set of sparse diffusion curves, which encode the smooth, low-frequency signals (e.g., the skin color), and pixel-level Poisson regions, which encode the high-frequency details. Using a trained convolution neural network, we segment the facial features and label the DCs into three categories: skin (red), hair (blue) and background (green). Then we formulate the color transfer problem as an optimal mass transport between the labeled DCs. After solving Poisson's equation using an efficient random-access solver, we obtain the retouched portrait . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for matching semantically-meaningful dense correspondence between images to obtain a more accurate object-to-object color transfer. Neural network style transfer methods [3,4] could be used as color transfer successfully. However, they were easy to cause distortion that made the results look like paintings that was not good for real photos' color transfer, and Luan et al. [23] improved the effect of photo style transfer by using energy function to constrain local affine transfer.

3. Overview

Our method consists of four steps. First, we vectorize the input and reference pair with diffusion curves and Poisson regions. Diffusion curves are sparse curves representing the salient features and each curve is two sided and assigned boundary colors on each side. Intuitively speaking, vectorization is to "push" the low-frequency colors of the non-feature regions into the boundary colors of the extracted diffusion curves. Poisson regions are the Laplacian of color residual, which encode the high-frequency details. Second, we apply a convolution neural network to label the diffusion curves, such as skin, hair and background. Third, we formulate the color transfer problem as an optimal transport between the diffusion curves.

So the vector images are separated into two levels — the low-frequency signals such as the main color and the high-frequency details. The vectorization provides both the target of color transfer and more degree of control like adjusting global contrast by decreasing or increasing Laplacian values. Second, by using a trained convolution neural network, we segment the facial features and label the DCs into three categories: skin, hair and background. Then, we formulate the color transfer problem as an optimal mass transport between the labeled DCs, and implement it by the sliced Wasserstein distance algorithm. Finally, we apply the local Poisson solver rendering the regions of interested to obtain the output image.

Fig. 2 illustrates the pipeline of our color transfer method and the corresponding pseudo-code is given in Algorithm 1.

Algorithm 1 Vectorization based color transfer on portrait images

Require: The input image I, the reference image R
Ensure: The output image O

1: Vectorize I and R using diffusion curves DC and Poisson regions PR (Section 4)
2: Label the diffusion curves as skin, hair, and background (Section 5)
3: Use optimal mass transport to transfer the boundary colors of DC_R to DC_I (Section 6)
4: Apply a local Poisson solver to render the diffusion curves with new boundary colors DC_I' and the original or modified Poisson regions PR_I (Section 7)

4. Vectorization

To achieve a vector image that can change color conveniently and resume the original image, we combine diffusion curve and Poisson image. If only use diffusion curves, there need very dense curves to basically resume the effect of the original image, but too many curves hinder the editing and color change. To decrease the number of curves and get complete image details, we use edge detection method to obtain enough DCs and calculate its Poisson region to maintain details. And we apply a local Poisson solver to update the color of vector image locally.

For a proper vectorization based Poisson's equation, its boundary condition given by DCs is extracted automatically. To conveniently edit DCs, we should control the number of DCs as less as possible. But to effectively color transfer, DCs need to include enough low-frequency information of the main color, and can render a result as true as original image with adding high-frequency details from Poisson region.

First, we calculate Poisson regions of the whole image as Laplacian values. In the discrete case, it is easy to obtain Laplacian value of each pixel p_i through the RGB color of its neighborhood. Let p_j be the left, right, top, and bottom position of p_i, the 3D

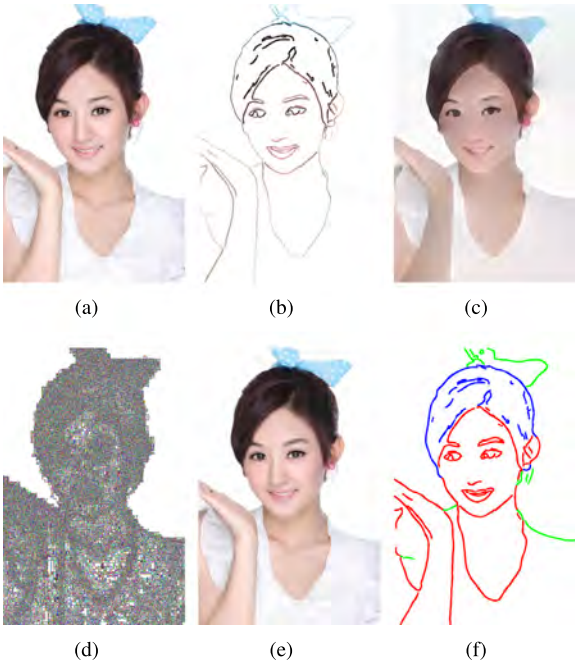

Fig. 3. Vectorization. (a) The original image. (b) Extracted diffusion curves with boundary colors. (c) The diffusion curves encode only the low-frequency colors. (d) The Poisson regions are the pixel-level regions that encode high-frequency details. (e) Combining both diffusion curves and Poisson regions, we obtain a highly accurate vector representation. (f) Label the diffusion curves into three categories: skin (red), hair (blue) and background (green) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 4. We segment the portrait image into three parts: background, skin, and hair.

Laplacian value of p_i can be calculated as

$$Lap(p_i) = \sum_{j=1}^{4} \big(RGB(p_j) - RGB(p_i)\big). \tag{1}$$

Fig. 3 shows an example of vectorizing portrait image.

Then, we extract DCs as boundary condition through the edge detection method [24]. All the edges with non-zero probabilities are deemed as DC candidates and we reserve the edges, whose sum of Laplacians around the edges above a given threshold, as DCs. Because large gradient means leap and comparative colors representing the main color around at the greatest extent. Only the edges with sharp color variations are necessary for our color transfer algorithm, so our DC extraction method is robust for the application.

Finally, associating with Laplacian values f and boundary condition g, we solve Poisson's equation (5) by using our local Poisson solver (see Section 7) and render the vector image.

5. Labeling Facial Features

To automatically label portraits with appropriate facial parts, we adopt the deep learning based method proposed in [25], which is one of the state-of-the-art face parsing methods.

The network structure is the FCN-8s [26] which is commonly used for semantic image segmentation. Here we train the labeling CNN model on HELEN dataset [27], which contains 2330 facial images of arbitrary size, and each pixel of its image is labeled as 11 classes (background, hair, facial skin, left eyebrow, right eyebrow, left eye, right eye, nose, inner mouth, upper lip, and lower lip). Fig. 4 shows three main labels of the above portrait in Fig. 3.

As above, after vectorization, DCs represent the main color of the original image, and the target of color transfer becomes these sparse curves. Generally, not all of DCs belong to the interest class like skin or hair, so we use the segmentation results to label DCs automatically. Specifically, we use these segmentation masks of background, hair, and skin (combining with facial skin, brows, eyes, nose, and mouths) to label each control point of DC. Because some DCs are very long, they are not always belonging to a single class. And we can see that DCs are sparsely distributed and its number is not much, that makes the labeling result is relatively good even if its segmentation is not that accurate.

However, this labeling result is not satisfied enough, especially for skin color transfer like neck labeled as background instead of skin. Because the facial skin area of segmentation is only covering facial skin and it cannot separate face and background perfectly. But the facial skin label still provides the basic information of skin color, and it helps to improve recognizing other DCs that actually belong to the skin like neck and hands.

Take skin labeling for example. At first, sort the color of DCs labeled as facial skin, and break off both ends to remove the superfluous parts brought by inaccurate facial skin mask. Then we have a range of skin color to measure other DCs whether they belong to skin or not. Actually, this method can apply to the situation that skin color has a big difference comparing to hair and background. When their color is similar, this method cannot help labeling automatically, so our system allows the user to assign and change labels where necessary. In the process, we do not need perfect labels, especially for references.

6. Color transfer with optimal mass transport

Optimal mass transportation is a useful method for color transfer by operating the color histograms of the input and output. In simple terms, it solves the problem that transports the color histogram of input to references by using minimum cost.

Specifically, let $I = \{c_i \,|c_i \in \mathbb{R}^{m\times 3}\}_{i=1}^{m}$, $R = \{c_j \,|c_j \in \mathbb{R}^{n\times 3}\}_{j=1}^{n}$ be the RGB color of the input and reference images, and H_I and H_R be their normalized color histograms. A transportation plan is σ: $\{c_i\} \to \{c_j\}$, $\sigma = \{\sigma_{ij}\}$, σ_{ij} means how many pixels of color c_i changed to color c_j. And the optimal mass transportation plan is defined as

$$\arg\min_{\sigma} \sum_i \sum_j \sigma_{ij} \|c_i - c_j\|^2 \tag{2}$$

with constraints:

$$\sigma_{ij} \geq 0, \quad \sum_{j=1}^{n} \sigma_{ij} = H_{c_i}, \quad \sum_{i=1}^{m} \sigma_{ij} = H_{c_j}.$$

Fig. 5. Color transfer for skin and hair with different iterations N. The first column shows the input images and the last column is the references. The middle columns (from left to right) are the results with $N = 1, 5, 10, 20, 30, 50$, respectively.

This problem is known as Kantorovich's formulation [28], and it can be solved by linear programming method. However, its solution may cause some pixels of the same color transported to different colors that is not a good feature for the application of color transfer.

Monge's formulation [29] is to find a transport map $T : \{c_i\} \to \{c_j\}$ that realizes

$$\arg\min_{T} \sum_{i=1}^{m} H_{c_i} \|c_i - T(c_i)\|^2 \tag{3}$$

with constraints:

$$H_{T(I)} = H_R.$$

It can provide a result that all pixels of each color transferred to a single color. But Monge's formulation may be no solution, because sometimes there is no T satisfying $H_{T(I)} = H_R$, for example, where the number of color I is not equal to color R.

Here we apply the sliced Wasserstein distance algorithm [30] to solve this problem. This algorithm uses an iteration method to give an approximate solution such that $H_{T(I)} \approx H_R$:

$$T^{k+1}(I) = (1-\alpha)T^k(I) + \alpha T(T^k(I)),\ \ k = 1, 2, \ldots, N. \tag{4}$$

with initial condition:

$$T^1(I) = I$$

where N is the number of iterations, and α is the linear parameter. By using this algorithm, we can always obtain a solution. Meanwhile, its solution makes sure to transport all pixels of a same color into a single color. For one-dimensional data, we can obtain results easily by sorting and interpolating values. But for multi-dimensional data such as 3D data in RGB color transfer, the sliced Wasserstein distance algorithm applies its one-dimensional algorithm to each projection of multi-dimensional data with a random basis and then iterates to obtain a final approximate result. Comparing with directly multi-dimensional transportation, the sliced algorithm is more efficient and consume less time. In our experiment, we set $\alpha = 0.2$ as in [31] and $N = 30$ whose result is good enough on labeled diffusion curves. Fig. 5 shows that the transfer results with different iterations.

To reduce color leap in skin color transfer in practice, we add Gaussian noises as in [6] to their colors to make that a color can be transferred to a relatively similar color. It increases the data like five times in my experiment, though it has less effect on running time because the data is only some sparse curves and it brings more smooth and natural results. However, it is not necessary for hair color transfer, because hair is supposed to be more closed to the effect of reference with direct transfer.

After clarifying DCs as skin, hair, and background, the target of color transfer becomes labeled DCs from one classification, e.g. skin. So the objectives of optimal mass transportation are some sparse curves. Apply the Sliced Wasserstein Distance algorithm on these labeled DCs of input and change their colors referring to DCs with the same label. Then locally render of the region of interest to obtain the final result of the color transfer. In actual, the process is changing the boundary condition and with the same Poisson region. Therefore, the details of the input remain unchanged that keep the original texture and meanwhile change color.

7. Rendering

To render and update the vector graphics, we solve Poisson's equation

$$\Delta u(\mathbf{x}) = f, \quad u(\mathbf{x})|_{\mathbf{x}\in\partial\Omega} = g, \tag{5}$$

where f is the Laplacian constraints, and g is the Dirichlet boundary condition of colors. We develop a harmonic B-spline [32] based local solver for (5).

Let $\Omega \subset \mathbb{R}^2$ be a 2D compact domain and $\mathcal{T} = \{\mathbf{t}_i | \mathbf{t}_i \in \Omega\}_{i=1}^n$ a set of knots. Taking $\{\mathbf{t}_i\}$ as the generators, we construct a Voronoi diagram $\Omega = \bigcup_{i=1}^n \mathcal{V}_i$, where $\mathcal{V}_i$ is the Voronoi cell of knot $\mathbf{t}_i$. For arbitrary points $\mathbf{x}, \mathbf{y} \in \Omega$, Green's function of the Laplace operator Δ satisfies

$$\Delta\phi_{\mathbf{y}}(\mathbf{x}) = \delta_{\mathbf{y}}(\mathbf{x}), \tag{6}$$

where $\delta_{\mathbf{y}}(\mathbf{x})$ is the Dirac delta function centered at $\mathbf{y}$. In 2D space, we can give its analytic solution

$$\phi_{\mathbf{y}}(\mathbf{x}) = \frac{1}{2\pi} \log(\|\mathbf{x} - \mathbf{y}\|).$$

For a Voronoi cell $\mathcal{V}_j$, applying Green's theorem to Eq. (6) yields

$$\int_{\mathcal{V}_j} \Delta\phi_{\mathbf{y}}(\mathbf{x})\, \mathrm{d}\sigma = \int_{\partial\mathcal{V}_j} \frac{\partial\Delta\phi_{\mathbf{y}}(\mathbf{x})}{\partial\mathbf{n}}\, \mathrm{d}s, \tag{7}$$

where $\mathbf{n}$ is the outward unit normal to the boundary $\partial\mathcal{V}_j$, $\mathrm{d}\sigma$ and $\mathrm{d}s$ are the area and line integral elements, respectively.

We define a basis function ψ_j for each Voronoi cell $\mathcal{V}_j$ as

$$\psi_j(\mathbf{x}) = \sum_i w_{ij}\phi_{\mathbf{t}_i}(\mathbf{x}), \tag{8}$$

Fig. 6. Skin color transfer results. The original images are in red frames, and the rest are the transferred results.

Fig. 7. Skin color transfer between oil/watercolor paintings and real portraits. Row 1 is the recoloring results of watercolor painting according to the real faces (the first and these mini images are original), the middle row is the recoloring results of according to the same reference as above row (the first image is original), and the skin transfer results of these two paintings are shown at the left lower column. The third row shows the transfer results from watercolor painting (the first three) and oil painting (the last three). Then by decreasing or increasing their Poisson region, we can obtain the results like watercolor or oil-painting style (the last row).

where w_{ij} is the discrete Laplacian weight and $\sum_i w_{ij}\phi_{\mathbf{t}_i}(\mathbf{x})$ is a boundary sum that approximates the line integral on the right hand side of Eq. (7). Feng and Warren [32] showed that the functions ψ_j are approximately local, nonnegative, and satisfying partition of unity.

Green's third identity provides an analytical solution for Poisson's equations in the form of integral

$$u(\mathbf{x}) = \iint_{\Omega} \phi_{\mathbf{y}}(\mathbf{x})\Delta u(\mathbf{y})d\sigma_{\mathbf{y}} + \oint_{\partial\Omega} \left(u(\mathbf{y})\frac{\partial \phi_{\mathbf{y}}(\mathbf{x})}{\partial \mathbf{n}} - \phi_{\mathbf{y}}(\mathbf{x})\frac{\partial u(\mathbf{y})}{\partial \mathbf{n}} \right) dl_{\mathbf{y}}, \tag{9}$$

where $d\sigma$ and dl are the surface and line elements, $\mathbf{n}$ is the outward pointing unit normal of dl.

Using harmonic B-splines [33], we compute the approximate solution by

$$u(\mathbf{x}) = \sum \lambda_j \psi_j(\mathbf{x})$$

where if λ_j is a boundary, use the Dirichlet boundary condition g to assign it; and if not, calculate it by solving a sparse linear system.

By using this Poisson solver on vector images, we can render and update them in a local manner.

8. Results

By using our method, we can obtain natural results of skin color transfer with the remaining other regions unchanged except for interested regions. Fig. 6 shows the skin color transfer between six real persons of different skin color from white to black.

Rather than human photos, our method can be applied to skin color transfer between painting-to-painting and paining-to-photo. In Fig. 7, the first two rows are the skin color transfer results of watercolor and oil paintings referring to real portraits, the rest two of the first column are between these two paintings, and the last six of the third row is the results of photos referring to watercolor painting (the first three) and oil painting (the last three). To create the effect like the style of watercolor or oil painting, we half and double their Poisson region based on the color transfer results of the above row.

Moreover, the portrait lighting method proposed in [6] can also be used in skin color transfer. To transfer portrait lighting and colors, they apply a mass transport method on 8D data of facial areas including RGB colors, 2D coordinates, and 3D normals. So their interested regions are the whole facial areas, whose data mount is much more than sparse curves as ours. Due to the high dimension problem they solved, their method is computationally

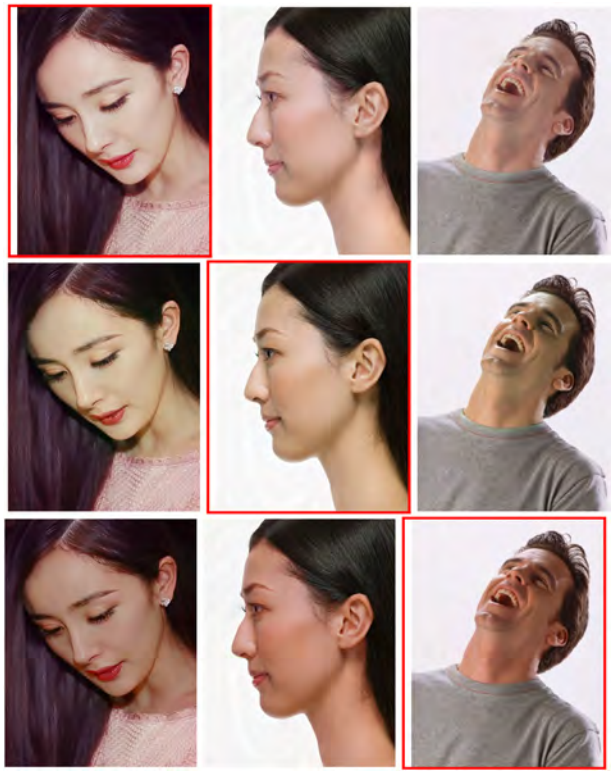

Fig. 8. Color transfer among portraits with different head positions. The images in red frames are the original images.

expensive. It took roughly 3 min to process a portrait image of size 500 × 600. In our method, vectorization is the bottleneck, which takes 30–60 s. After that, labeling the diffusion curves and transfer colors are highly efficient, taking only a few seconds. Rendering the diffusion curves and Poisson regions using our random-access solver also takes a few seconds. Therefore, our method is more efficient than Shu et al.'s approach [6]. Furthermore, since our results are vector primitives, it is relatively easy for downstream applications, such as editing the geometry of facial features, and cloning.

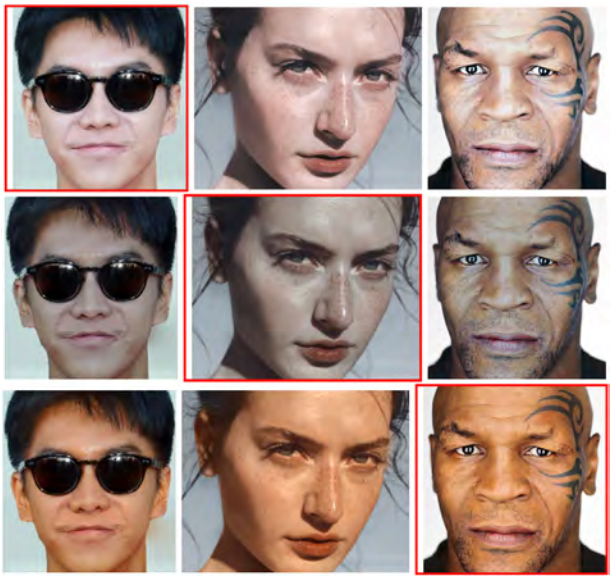

Fig. 9. Our method can preserve features such as glasses, freckles, and tattoos.

Our method can produce good visual effects on not only front faces but also other postures including side faces dropped heads, and raised heads as shown in Fig. 8. Vectorization decomposes the image into two levels where the color change is in the low-frequency information, so our method is able to maintain the texture and high-frequency details of the original images. As shown in Fig. 9, whether freckles or tattoos on faces or covered by something like sunglasses, we can obtain a good result maintaining original details.

Note that Shih et al.'s method [5] assumes a dense correspondence between the input-reference pair, therefore, it works only for portrait photos with similar postures and facial features. In

(a) Reference (b) Input (c) Our result (d) Color transfer in [6] (e) Relighting in [6]

(f) Reference (g) Input (h) Our result (i) [5]

Fig. 10. Comparison with the state-of-the-art methods [6] and [5]. The three methods have slightly different objectives: our method focuses on skin and hair colors, Shu et al.'s method [6] deals with lighting transfer and Shih et al.'s method [5] targets skin colors and textures. Our method is more computationally efficient than the other approaches and it does not require dense correspondence between the input and the reference images.

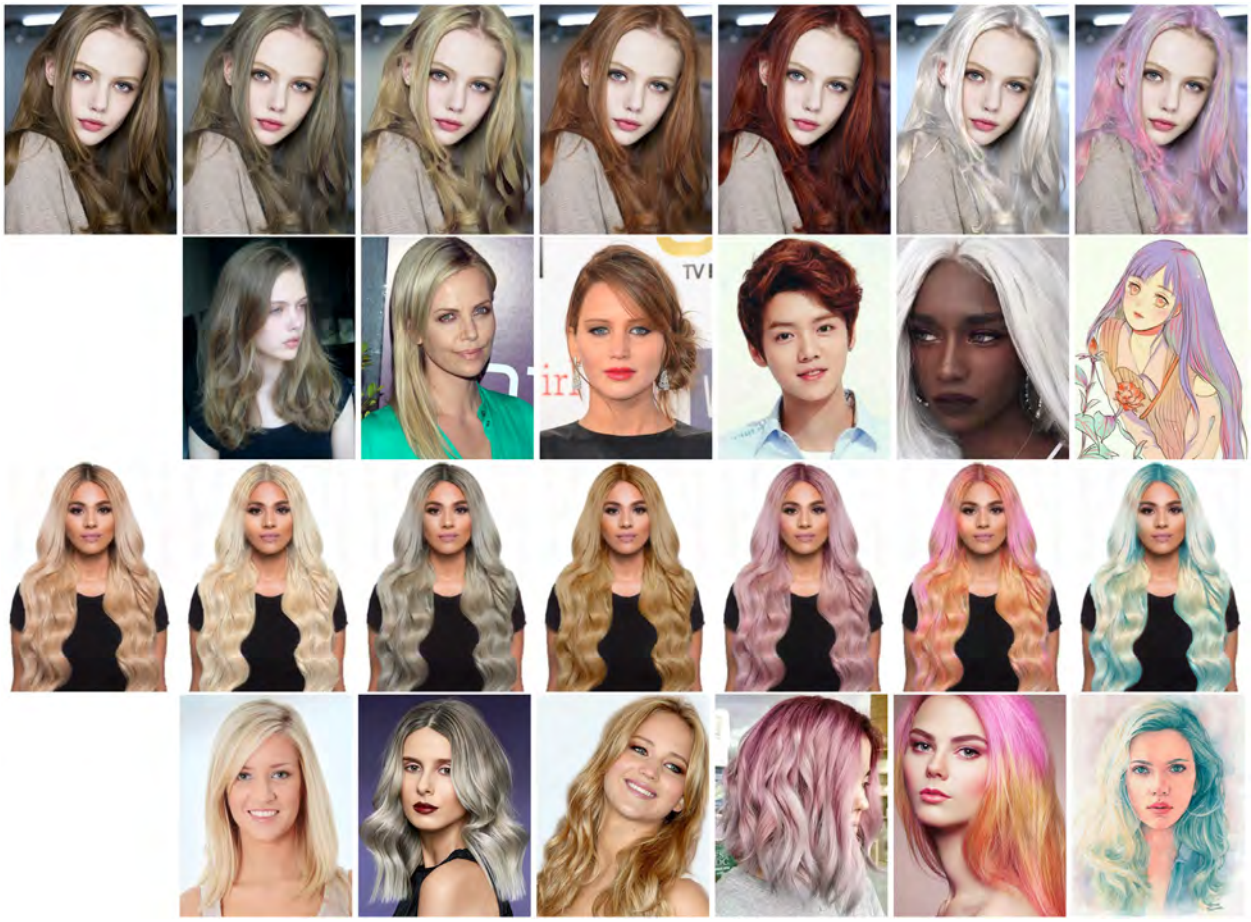

Fig. 11. Hair color transfer for photos and paintings. The input images are in the first column and the reference images are in the second and fourth rows.

Fig. 12. By labeling the diffusion curves of eyes and mouths, our method can also transfer their colors. Left: the original image; middle and right: the transferred results.

contrast, our method does not require such a correspondence, hereby is flexible and can handle portraits with different head positions and facial features (see Fig. 8). Moreover, our method also works for non-photorealistic images, such as cartoon and paintings.

In addition, we compare our method with [6] and [5]. As shown in Fig. 10, our method can achieve comparable results by keeping skin textures and highlights. Besides skin color, our method can be used on hair color transfer too. Fig. 11 shows two girls' hair color (the first column) changed referring to different hair colors whether their hair colors are similar or not. And we can generate good results not only on whose hair color has only one color but also multiple colors. And rather than real persons, the reference can be painting too (the last column). Our current method labels the regions in three main categories — skin, hair, and background. By labeling eyes and mouth, our method can also transfer their colors (see Fig. 12).

9. Conclusions

This paper presents a simple yet effective method for transferring skin and hair color between portrait images. First, we

Fig. 13. Our method fails on portraits with significantly different color distributions. Note that the input image has a very low contrast, whereas the reference image has a high contrast. Tuning the PRs can reduce the artifacts.

apply vectorization to split original images into low and high frequencies, where low frequency given by sparse DCs including main color information as the target of color transfer, and high frequency provided by Poisson region including other details. Next, we use a segmentation algorithm to label DCs to obtain the specific target of optimal mass transportation to change their color. Finally, locally render the output with our Poisson local solver. The experimental results provide some color transfer results of portraits skin and hair, but actually, color transfer of smaller regions like mouth or eyes can be transferred in a similar way. Furthermore, our method can be applied to other objects or sceneries too.

However, the presented method has limitations. When the colors of the interested regions of the input and reference have a massive difference, the color transfer results may be artificial, even if adjusting their Poisson region can make moves to improve their effects as shown in Fig. 13. The segmentation results are not accurate, and we can completely avoid labeling by the user if we have accurate segmentation masks in the future. Moreover, there is not a numerical or quantitative evaluation to qualify the results besides the visualization images provided in this paper. We plan to propose a numerical evaluation of the color transfer results that can also be used to find the best matching reference images automatically.

Acknowledgments

This project was partially supported by Singapore Ministry of Education Grant RG26/17, NTU-Alibaba Joint Research Institute, Singapore, National Natural Science Foundation of China Grants (61872347, 61672481, 61725204, and U1736220), Special Plan for the Development of Distinguished Young Scientists of ISCAS, China (Y8RC535018), Youth Innovation Promotion Association CAS, China (No. 2018495) and the Royal Society-Newton Advanced Fellowship, China (NA150431).

References

[1] Reinhard E, Adhikhmin M, Gooch B, Shirley P. Color transfer between images. IEEE Comput Graph Appl 2001;21(5):34–41.
[2] Pitie F, Kokaram AC, Dahyot R. N-dimensional probablility density function transfer and its application to colour transfer. In: Proceedings of ICCV '05. IEEE; 2005, p. 1434–9.
[3] Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 2414–23.
[4] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. Springer; 2016, p. 694–711.
[5] Shih Y, Paris S, Barnes C, Freeman WT, Durand F. Style transfer for headshot portraits. ACM Trans Graph 2014;33(4):148.
[6] Shu Z, Hadap S, Shechtman E, Sunkavalli K, Paris S, Samaras D. Portrait lighting transfer using a mass transport approach. ACM Trans Graph 2018;37(1):2.
[7] Orzan A, Bousseau A, Winnemöller H, Barla P, Thollot J, Salesin D. Diffusion curves: A vector representation for smooth-shaded images. ACM Trans Graph 2008;27(3):92:1–8.
[8] Bezerra H, Eisemann E, DeCarlo D, Thollot J. Diffusion constraints for vector graphics. In: Proceedings of the 8th international symposium on non-photorealistic animation and rendering. NPAR '10, New York, NY, USA: ACM; 2010, p. 35–42.
[9] Finch M, Snyder J, Hoppe H. Freeform vector graphics with controlled thin-plate splines. ACM Trans Graph 2011;30(6):166:1–166:10.
[10] Lieng H, Tasse FP, Kosinka J, Dodgson NA. Shading curves: Vector-based drawing with explicit gradient control. Comput Graph Forum 2015;34(6):228–39.
[11] Hou F, Sun Q, Fang Z, Liu Y-J, Hu S-M, Qin H, Hao A, He Y. Poisson vector graphics (pvg). IEEE Trans Vis Comput Graphics 2018;1. http://dx.doi.org/10.1109/TVCG.2018.2867478, URL doi.ieeecomputersociety.org/10.1109/TVCG.2018.2867478.
[12] Sun J, Liang L, Wen F, Shum H-Y. Image vectorization using optimized gradient meshes. ACM Trans Graph 2007;26(3):Article 11.
[13] Lai Y-K, Hu S-M, Martin RR. Automatic and topology-preserving gradient mesh generation for image vectorization. ACM Trans Graph 2009;28(3):85:1–8.
[14] Yin X, Dai J, Yau S, Gu X. Slit map: Conformal parameterization for multiply connected surfaces. In: Advances in geometric modeling and processing, 5th international conference, GMP 2008, Hangzhou, China, April 23-25, 2008. Proceedings. 2008, p. 410–22. http://dx.doi.org/10.1007/978-3-540-79246-8_31.
[15] Xie G, Sun X, Tong X, Nowrouzezahrai D. Hierarchical diffusion curves for accurate automatic image vectorization. ACM Trans Graph 2014;33(6):230:1–230:11.
[16] Zhao S, Durand F, Zheng C. Inverse diffusion curves using shape optimization. IEEE Trans Vis Comput Graphics 2018;24(7):2153–66.
[17] Song Y, Wang J, Wei L-Y, Wang W. Vector regression functions for texture compression. ACM Trans Graph 2015;35(1):5:1–5:10.
[18] Hwang Y, Lee J-Y, So Kweon I, Joo Kim S. Color transfer using probabilistic moving least squares. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, p. 3342–9.
[19] Lee J-Y, Sunkavalli K, Lin Z, Shen X, So Kweon I. Automatic content-aware color and tone stylization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 2470–8.
[20] Wu F, Dong W, Kong Y, Mei X, Paul J-C, Zhang X. Content-based colour transfer. In: Computer graphics forum, Vol. 32. Wiley Online Library; 2013, p. 190–203.
[21] Yang Y, Zhao H, You L, Tu R, Wu X, Jin X. Semantic portrait color transfer with internet images. Multimedia Tools Appl 2017;76(1):523–41.
[22] He M, Liao J, Chen D, Yuan L, Sander PV. Progressive color transfer with dense semantic correspondences. ACM Trans Graph 2019;38(2). Article No. 13.
[23] Luan F, Paris S, Shechtman E, Bala K. Deep photo style transfer, CoRR, abs/1703.07511 2 (2017) 4990–4998.
[24] Dollár P, Zitnick CL. Structured forests for fast edge detection. In: Proceedings of the 2013 IEEE international conference on computer vision. ICCV '13, Washington, DC, USA: IEEE Computer Society; 2013, p. 1841–8.
[25] Li Z, Zhang J. Pixel-level guided face editing with fully convolution networks. In: 2017 IEEE international conference on multimedia and expo, ICME 2017, Hong Kong, China, July 10-14, 2017. 2017, p. 307–12.

[26] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[27] Smith BM, Zhang L, Brandt J, Lin Z, Yang J. Exemplar-based face parsing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.
[28] Villani C. Topics in optimal transportation, Vol. 58. American Mathematical Soc.; 2003.
[29] Monge G. Mémoire sur la théorie des déblais et des remblais. Hist. Acad. Roy. Sci. Paris 1781;666704.
[30] Rabin J, Peyré G, Delon J, Bernot M. Wasserstein barycenter and its application to texture mixing. In: International conference on scale space and variational methods in computer vision. Springer; 2011, p. 435–46.
[31] Bonneel N, Tompkin J, Sunkavalli K, Sun D, Paris S, Pfister H. Blind video temporal consistency. ACM Trans Graph 2015;34(6):196.
[32] Feng P, Warren J. Discrete bi-laplacians and biharmonic b-splines. ACM Trans Graph 2012;31(4):115:1–115:11.
[33] Hou F, He Y, Qin H, Hao A. Knot optimization for biharmonic b-splines on manifold triangle meshes. IEEE Trans Vis Comput Graphics 2017;23(9):2082–95.