Guided Facial Skin Color Correction

Keiichiro Shirai¹, Tatsuya Baba², Shunsuke Ono³, Masahiro Okuda⁴, Yusuke Tatesumi¹, and Paul Perrotin⁵

¹Shinshu University, Japan
²The University of Kitakyushu
³Tokyo Institute of Technology, Japan
⁴Doshisha University, Japan
⁵The University Institute of Technology La Rochelle, France

Abstract

This paper proposes an automatic image correction method for portrait photographs, which promotes consistency of facial skin color by suppressing skin color changes due to background colors. In portrait photographs, skin color is often distorted due to the lighting environment (e.g., light reflected from a colored background wall and over-exposure by a camera strobe), and if the photo is artificially combined with another background color, this color change is emphasized, resulting in an unnatural synthesized result. In our framework, after roughly extracting the face region and rectifying the skin color distribution in a color space, we perform color and brightness correction around the face in the original image to achieve a proper color balance of the facial image, which is not affected by luminance and background colors. Unlike conventional algorithms for color correction, our final result is attained by a color correction process with a guide image. In particular, our guided image filtering for the color correction does not require a perfectly-aligned guide image required in the original guide image filtering method proposed by He et al. Experimental results show that our method generates more natural results than conventional methods on not only headshot photographs but also natural scene photographs. We also show automatic yearbook style photo generation as an another application.

1 Introduction

Portrait photographs sometimes get undesirable color due to a background color reflection, so that we need to make the skin color and brightness uniform for each photograph. In such a situation, a professional cameraman usually arranges lighting conditions with special equipment. Meanwhile, for an amateur, more effort is required to achieve the same quality. This leads to a demand for an automatic and simple method that does not require any special equipment for unifying the color and brightness between multiple images. The undesirable color on portrait photographs is caused by the following.

- Each camera has its own camera response sensitivity. Thus, the distribution of skin color in a color space also depends on the camera used.
- Background color is usually reflected to faces resulting in distorted colors.
- If each subject wears clothes with more than one color, color correction for the whole image region distorts the skin color, and skin color correction will discolor the clothes.

Therefore, a color correction scheme that affects only the face region is required.

In this paper, we attempt to correct an image with various color and brightness conditions with a guide image. Our method has various applications, e.g., the creation of a photo name list such as a yearbook with high quality facial images without much effort or cost. If one tries to gather photographs taken in different lighting conditions with different cameras, correcting the images becomes a very difficult task even using off-the-shelf image processing software. In the situation considered, we need to unify several attributes of all images: the size of cropped facial image, facial skin color, brightness, and background color. Automatic correction of the attributes requires

1. Face detection and Facial skin color extraction.
2. Facial skin color correction.

A simple combination of these techniques gives boundary artifacts between the corrected region and the uncorrected region because the color correction only corrects the extracted region. This is described in Sec. 2.

With the algorithm described in this paper, correction of facial skin color and brightness is performed using a hybrid guided image filtering (GIF) method. Because region extraction and color transfer are individual procedures, color transfer on part of an image generates a color gap between the original color part and the color transformed part. For this problem, we propose the hybrid GIF (the red box in
Fig. 1), which performs color transfer and segmentation. Only the face region is extracted (Fig. 1(1)), followed by adjustment with grading, and then the hybrid GIF corrects the face region of the input image by using the corrected image as a guide image while keeping colors in other regions (Fig. 1(2)). In other words, the hybrid GIF transforms a part of the image as an object based color transfer. In contrast to the fact that the original GIF [14] needs a perfectly-aligned guide image, our GIF only needs roughly aligned pixels. As for non-aligned regions, the filtering is achieved by aligned region propagation such as colorization [16] and matting [17]. Our method carries out nonlinear correction only on the face region and achieves better results than conventional methods. A preliminary version of this study without several improvements or new applications appeared in conference proceedings [3].

The rest of this paper is organized as follows. Sec. 2 introduces related work on color correction and facial image correction. Sec. 3 describes the proposed guide facial skin color correction method. The proposed algorithm is evaluated in experiments by comparison with various conventional methods in Sec. 4. Finally, this paper is concluded in Sec. 5.

2 Related work

Reference based color correction is one of the essential research issues for image editing. Color grading methods [11, 12, 23], which match the color attributes (tone curve and color distribution for each color) of a target image with those of another target image, are effective for color correction among multiple images. However, these existing techniques are global image correction operators and assume that subjects wear the same kinds of clothes. Other related techniques are image matting [13,17]. By using these techniques, one can change colors by specifying desired colors to represent pixels. However, to obtain natural colorization, much coloring information is needed. Another possible approach is to apply color transfer based on GIF with guide images [8,14,22,24], where coloring information is given in a block as a guide image. However, the guide image used in the methods is limited to perfectly aligned images without any position gap, which is often not the case for the situation considered.

As for facial image correction methods, the authors of [4] propose a wrinkle removal method but the purpose is different from color grading of facial skin color. The method proposed in [26] can also transfer the style of the original image to the target one. However, it also edits clothes and performs well only for headshot photos.

3 Proposed method

Our algorithm transforms colors of a part of an input image using a target image color unlike general color transfer methods and corrects surrounding colors of the transformed part. Fig. 1 illustrates the flow chart of our method. Our method mainly consists of two parts.

(1). **Face detection and Facial skin color extraction** (the yellow box in Fig. 1) : It detects the face part and extracts its facial skin color.

(2). **Skin color correction** (green box) : After extracting the skin region\(^1\) in Step (1), the distribution of the facial skin color is rectified using the image (a). The color of the face region is modified by using the image (b) as the guide image.

Note that this paper uses the two phrases, *skin color* and *facial skin color*. *Skin color* refers to the skin color of the whole body, while *facial skin color* is the color of the skin in the face region only.

The novelty of the proposed method lies in (2), where the color correction with grading affects only a part of the image. We use conventional methods for the other steps such as the face detection with some modifications. Each procedure is described in more detail hereafter.

3.1 Face detection

In the first step, we detect an area from head to shoulder (Fig. 2), where we use Haar-cascade detection in OpenCV, also known as the Viola-Jones algorithm [29], for the face detection. The face area is described by \(n\) rectangle windows, where \(n\) is the number of candidates, with a set of barycentric coordinates \((x, y)\) and the sizes of the detected candidate rectangles \((w, h)\) Fig. 2 (center). The blue boxes

---

\(^1\)We distinguish region and area. Area indicates the detected face area in Sec. 3.1, while region indicates the extracted facial skin color region in Sec. 3.2.
in Fig. 2 indicate the detected candidate rectangles. We find the median of \( x, y, w, h \) as

\[
\bar{x} := \text{median}\{x_1, \ldots, x_n\}, \quad \bar{w} := \text{median}\{w_1, \ldots, w_n\}, \\
\bar{y} := \text{median}\{y_1, \ldots, y_n\}, \quad \bar{h} := \text{median}\{h_1, \ldots, h_n\},
\]

where the subscript is the index of the candidate. The rectangle with the median values of the barycentric coordinate \((\bar{x}, \bar{y})\) is adopted as the final face area. Since the size of the face area \((\bar{w}, \bar{h})\) depends on the image, the size of the face area is adjusted by:

\[
(2\bar{w} + 1) \times (2\bar{h} + 1), \tag{2}
\]

where \( l \) is the prescribed scale factor.

### 3.2 Skin color extraction

Rough extraction of the skin color region near the face is performed by a common approach using morphological labeling in the image space and clustering in the color space:

(i) We classify the color of each pixel by clustering the color distribution of the entire image in the HSV color space, and for each pixel we allocate the label of the cluster that each pixel belongs to.

(ii) Some regions are generated by concatenating neighboring pixels with the same labels in the image space, and then we extract regions mainly in the detected face area \( \Omega_\text{rect} \) (Eq. (2)) and assign them to the facial skin region.

Note that, since the skin color in the detected face area (Sec. 3.1) is known, we may obtain satisfactory results by using simpler approaches such as the \textit{k-means} method \cite{18} with stable seed cluster selection \cite{2} (\textit{i.e., k-means++}) for clustering color distributions.

In this paper, for simplicity, we apply \textit{k-means} clustering \cite{2} to the 1-D distribution of the \textit{hue} values of the entire image, and we perform the above-mentioned procedures (i) and (ii) to detect the face region \( \Omega_s^{hue} \). Then we perform thresholding on the region \( \Omega_s^{hue} \) for the \textit{saturation} and \textit{value} components of every pixel. We define the skin color region

\[
\Omega_B = \Omega_s' \setminus \Omega_s
\]

\[
\Omega_{\partial S} := \text{dilation}(\Omega_s), \tag{4}
\]

where dilation(\cdot) is the dilation function with a structuring element, which consists of a circle with a 20 pixel radius. The overline \( \cdot \) is the complement and the operator \( \setminus \) is the set difference. An example of each region can be seen in Fig. 3.

### 3.3 Color grading by \cite{23}

Color grading (Fig. 4) is performed so as to bring the skin color of the extracted region \( \Omega_s \) (more specifically the shape of the color distribution\footnote{Note that we use the RGB color space instead of the HSV color space used in the preceding section. The reason is that the color grading method \cite{1} has been developed for the RGB color space and the color-line image feature used in the next section is a feature related to the RGB color space.}) close to the target facial skin color in Fig. 4(a). Color grading transforms the shapes of color distributions (r,g,b 3D coordinates). One may use simpler techniques such as the estimation of a tone curve to modify each RGB component. However, most of the methods need correspondence between the two images, which is not suitable in our case where one transfers the color between the faces of different persons. In our method we use Pitié et al.’s method \cite{23}, in which pixel correspondences are not
necessary. This method considers the width of the color distribution along a particular axis, iteratively changes it so as to match the specified axis, and then realizes non-linear reshaping of the color distribution. Since the method yields some artifacts such as blurring artifacts around the edges, we need to address this problem, which is described hereafter.

3.4 Guide image filtering via optimization

This section expresses our GIF, which uses a guide image to correct the input image, as an optimization based method. Our energy function actually resembles in part that of the image matting method [17] because we focus on and use its data fidelity term (in [17] that is also known as the local linear model), but the design objective is different. Additionally, a similar method has been proposed in high dynamic range imaging [25].

Our guide image filtering reconstructs an image \( x \in \mathbb{R}^{3N} \) where the facial skin color region has a corrected color by using the input image \( y \in \mathbb{R}^{3N} \) and the color grading result, where \( N \) is a number of pixels. In the RGB color space \( (\mathbb{R}^3) \), let the pixel value at pixel \( j \) of the color corrected image to be solved be \( x_j \), the input image be \( y_j \), and the guide image be \( g_j \) which are given by the color grading in Sec. 3.3. Figure 5 shows each image. Using them, we formulate our guide image filtering as the following convex optimization problem

\[
\min_{x, A, \lambda} \sum_i \left( \sum_{j \in w_i} \|x_j - A_i y_j - b_i\|^2 + \varepsilon \|A_i\|^2_F \right)_{\text{data fidelity}} + \|C_S(x)\| + \|C_B(x)\|_{\text{constraints}}
\]

where \( A_i \in \mathbb{R}^{3 \times 3} \) and \( b_i \in \mathbb{R}^3 \) are a scaling matrix and an offsetting vector to approximate \( y_j \) to \( x_j \) for each pixel \( i \), \( w_i \) is a square window around pixel \( i \), and \( C_S \) and \( C_B \) are given by

\[
C_S := \left\{ x \mid \sqrt{\sum_{p \in \Omega_S} \|x_p - g_p\|^2} \leq \eta_S \right\},
\]

\[
C_B := \left\{ x \mid \sqrt{\sum_{q \in \Omega_B} \|x_q - y_q\|^2} \leq \eta_B \right\}.
\]

Terms in the top row correspond to [17], where \( \| \cdot \|_2 \) and \( \| \cdot \|_F \) denote the \( \ell_2 \) norm and Frobenius norm respectively, and we use them as a data fidelity that reflects textures and local contrasts of \( y \) onto \( x \). The second and third terms are constraints expressed by the following indicator function:

\[
t_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise}. \end{cases}
\]

The second term brings the facial skin color close to that of the guide image \( g \) in the facial skin color region \( \Omega_S \). The third term keeps the color of the background the same as the original image \( y \) in \( \Omega_B \). For reducing undesirable artifacts arising from the guide image, we add a constraint that the color difference at each pixel does not exceed \( \eta \).

Note that we purposefully represent the second and third terms as constraint formulation over unconstrained one, e.g., the second term can be replaced with a regularizer

\[
\lambda \sqrt{\sum_{p \in \Omega_S} \|x_p - g_p\|^2}
\]

where \( \lambda \) is a balancing parameter. This is because \( \eta \) can be controlled more intuitively than \( \lambda \), and it allows us to adaptively change \( \eta \) depending on the area of \( \Omega_s \) (see the next Sec. 4). Such advantages of the formulation have been addressed in the literature of image restoration based on convex optimization [1,6,7,9,19,20,28].

Among convex optimization algorithms, we adopt a monotone version of the fast iterative shrinkage-thresholding algorithm (MFISTA) [5] to solve (5) because it is a first-order method that achieves a sublinear global rate of convergence, namely, computationally efficient and very fast. The algorithm solves an optimization problem of the form:

\[
\min_x f(x) + h(x),
\]

where \( f(x) \) is a differentiable convex function with a Lipschitz continuous gradient and \( h(x) \) is a proper lower semicontinuous convex function. The problem (8) is calculated

Figure 5: Guide image filtering. From left: Input image, Guide image, Target color in \( \Omega_B \) and \( \Omega_S \), and Color transformed image.
by the proximity operator 3. For given \( x^0 := \mathbf{v}^1 \in \mathbb{R}^N \) and \( t^1 := 1 \), the iteration of the MFISTA consists of the following five steps:

\[
\begin{align*}
\hat{z}^k &= \mathbf{v}^k - \frac{1}{L} \nabla f(\mathbf{v}^k), \\
\mathbf{z}^k &= \text{prox}_{h/L}(\hat{z}^k), \\
\mathbf{x}^k &= \arg \min \{ f(\mathbf{z}) + h(\mathbf{z}) | \mathbf{z} \in \{ \mathbf{z}^k, \mathbf{x}^{k-1} \} \}, \\
t^{k+1} &= 1 + \sqrt{1 + 4(t^k)^2}, \\
\mathbf{v}^{k+1} &= \mathbf{x}^k + \frac{t^k}{t^{k+1}} (\mathbf{z}^k - \mathbf{x}^k) + \frac{t^k - 1}{t^{k+1}} (\mathbf{x}^k - \mathbf{x}^{k-1}),
\end{align*}
\]

where “prox” is the proximity operator and \( 1/L \) is the step size.

In order to apply the MFISTA to our problem (5), we set

\[
f(\mathbf{x}) := \sum_i \left( \sum_{j \in w_i} ||x_j - A_i y_j - b_i||^2_2 + \varepsilon ||A_i||^2_F \right),
\]

\[
h(\mathbf{x}) := \iota_{c_S}(\mathbf{x}) + \iota_{\Omega}(\mathbf{x}).
\]

To compute the gradient \( \nabla f(\mathbf{x}) \), we use a method similar to [13], which is an accelerated version of [17]. We compute each value of \( |\nabla f(\mathbf{x})|_i \in \mathbb{R}^3 \) as follows:

\[
A_i^* = \Delta_i^{-1} \left( \frac{1}{|w_i|} \left( \sum_{j \in w_i} y_j x_j^T - \bar{y}_i \bar{x}_i^T \right) - \bar{y}_i \bar{x}_i^T \right),
\]

\[
b_i^* = x_i - A_i^* y_i,
\]

\[
|\nabla f(\mathbf{x})|_i = |w_i| y_i - \left( \sum_{j \in w_i} A_i^* y_i + \sum_{j \in w_i} b_j^* \right),
\]

where \( \bar{x}_i = [\bar{x}_i^R \bar{x}_i^G \bar{x}_i^B]^T \) and \( \bar{y}_i = [\bar{y}_i^R \bar{y}_i^G \bar{y}_i^B]^T \) are mean value vectors of each color in a square window \( w_i \), \( |w_i| \) is the number of pixels in \( w_i \), \( \Delta = \Sigma_i + \frac{1}{|w_i|} U \in \mathbb{R}^{3 \times 3} \), \( \Sigma_i = \frac{1}{|w_i|} \left( \sum_{j \in w_i} y_j y_j^T \right) - \bar{y}_i \bar{y}_i^T \) is a covariance matrix, and \( U \) is an identity matrix.

The proximity operator \( \text{prox}_{h/L} \) in (9) consists of two functions which are the second and third terms of (5) which handle different regions, \( \Omega_S \) and \( \Omega_B \), respectively that satisfies, \( i.e., \), \( \Omega_S \cap \Omega_B = \{0\} \). Therefore, \( \text{prox}_{h/L} \) can be calculated using

\[
\mathbf{z}_i := \left[ \text{prox}_{h/L}(\hat{z}) \right]_i = \begin{cases} \hat{z}_i - g_i, & \text{if } \hat{z} \notin \mathcal{C}_S \land i \in \Omega_S, \\ \hat{z}_i - y_i, & \text{if } \hat{z} \notin \mathcal{C}_B \land i \in \Omega_B, \\ \hat{z}_i, & \text{otherwise.} \end{cases}
\]

This process corresponds to a \( \ell_2 \)-ball projection with a region constraint.

Finally, we update \( \mathbf{x}, \mathbf{z}, \mathbf{v}, t \) using the procedure in (9), then the solution \( \mathbf{x} \) becomes our GIF result.

4 Results and discussion

In this section, we show the results obtained through the proposed process 4. The range of RGB values is normalized to be in the range \([0, 1]\). The prescribed scale factor \( l \) in Sec. 3.1 is set to 2.0. The filter window sizes used in Sec. 3.4 and Sec. 5 are 19 × 19 and 31 × 31, respectively. \( \eta_S = 5|\Omega_S| \times 10^{-4} \) and \( \eta_B = 5|\Omega_B| \times 10^{-10} \) are used, where \( |\Omega_S| \) indicates the number of pixels contained in the region \( \Omega_S \).

Our filtering often flattens the gradation of the input images caused by shadows. It yields unnatural images as shown in the top of Fig. 6, where the input image is input

\[\text{Figure 6: Simple luminance correction. Input image and filtered image are decomposed into intensity components and color components, respectively by (15), then the intensity component of the input image and the color component of the filtered image are combined by (16).}\]
Figure 7: Our facial skin color correction. (a) Target facial skin color image, (b) Input image, and (c) Output image using our facial skin color correction. The target image is in Shin et al.'s dataset [26] and the input images are in the Gehler dataset [10].

Figure 8: Our facial skin color correction on the flash images. (a) Target facial skin color image, (b) Input flash images, (c) Output images using our facial skin color correction.
to our algorithm and the filtered image is a result of our guide image filtering. For the luminance correction, each pixel color value of the image is decomposed into a color component \( x_i^C \in \mathbb{R}^3 \) and an intensity component \( x_i^I \in \mathbb{R}^1 \) as follows

\[
x_i^I = \frac{x_i^R}{\|x_i\|_1} + \frac{x_i^G}{\|x_i\|_1} + \frac{x_i^B}{\|x_i\|_1},
\]

\[
x_i^C = \frac{x_i}{x_i^I},
\]

where the superscripts \( R, G, \) and \( B \) indicate each color component. This decomposition procedure is same as [8].

The input image and the filtered image are decomposed into the two components by (15), and then the intensity component of the input image \( y_i^I \) and the color component of the filtered image \( x_i^C \) are combined as follows

\[
x_i' = y_i^I x_i^C.
\]

Figure 6 shows this procedure and its effectiveness.

Figure 7 (hereafter indexes are swapped due to figure positions) shows the result of our skin color correction. The area around the face in the input image has color distortion due to the lighting conditions and background color. Our result has a white balanced facial skin color similar to the target facial skin color. Sometimes when we take a photograph in dark surroundings, the results sometimes have unnatural face colors (Fig. 8(b)) due to the camera flash. Hence, we apply our proposed method to flash images in dark surroundings to reduce undesirable effects of artificial lights. Figure 8(c) shows the flash image editing results using our method. One can see that unnatural colors of the original image are corrected to natural colors by our method.

4.1 Automatic yearbook style photo generation

This section presents automatic yearbook style photo generation using our guided facial skin correction method and pre/post-processing procedure. It takes a long time to manually process a large number of images. Our algorithm generates a yearbook style photo in a short amount of time.

We first crop a photo with our face detection procedure as a pre-processing procedure (the red box in Fig. 9), and then correct the facial skin color. Finally, the noisy background is replaced to clear the background as a post-processing procedure (the blue boxes in Fig. 9).

4.1.1 Face area cropping

In Sec. 3.1, we detect the face area of a photo. Then we unify the sizes of the cropped images by expansion and reduction. The crop size is roughly adjusted according to the image size. The image size after cropping and resizing is \( 320 \times 320 \) (\( h \times w \)) in this experiment.

4.1.2 Background replacement by alpha blending

We separate the image information of the foreground and background regions and assign a value \( \alpha_i := [0, 1] \) to them. Using the values as labels, the relationship between the foreground \( f_i \), background \( b_i \) (different from that in (5)), and original image \( y_i \) at each pixel is given as follows:

\[
y_i := f_i + b_i, \quad f_i := (1 - \alpha_i)y_i, \quad b_i := \alpha_i y_i.
\]

The label \( \alpha \) at each pixel is the blending rate, and called alpha-mat. Replacement of the background with another background \( z \) is performed by

\[
y_i' := f_i + \alpha_i z_i.
\]

An estimation of the alpha-mats is described in Appendix A.

Figure 10 shows the results of our automatic yearbook style photo generation. The red box indicates a pre-processing procedure and the blue boxes indicate a post-processing procedure.
Figure 10: Results of automatic yearbook style photo generation using our method. (a) Target facial skin color image, (b) Original images which have a background color similar to the facial skin color, (c) Yearbook style images using our algorithm.

Figure 11: Supplement quality comparison with similar methods. (a) Original, (b) Guide, (c) Colorization for each RGB layer [16], (d) Joint Bilateral Upsampling (JBU) for each RGB layer [15], and (e) Our guided filtering (5). The black pixels in (b) represents a hole region and each red arrow indicates an artifact.
4.2 Comparison with various conventional method

Figure 11 compares our method with supplement methods [15, 16]. For the methods, we process each RGB color layer because they are proposed for color components in YUV color space. Since colorization [16] just spreads colors to boundaries, the result does not have the details of the original image. Joint Bilateral Upsampling [15] computes pixel values in hole regions by joint-filtering. We can see a contrast reduction in the top and the middle rows in Fig. 11. The blue arrows and the red arrows indicate artifacts of each method. Meanwhile our guide image filtering outputs sharp details of the original image while keeping the guide image color.

Figure 12 shows a comparison with existing color transfer methods. For the results (c), Pitié et al.’s method [23] transforms the color distribution of the input image to the target image one, then gain noise artifacts of the color transformed input image are removed by Rabin et al.’s method [24]. The gain noise artifacts can be seen in [23]. In the results of the color grading [24] with [23], the face and the background of the resultant image have a similar color since this method is a uniform color transfer method. In addition, this method takes a long time because it needs iterative bilateral filtering. In the non-rigid dense correspondence (NRDC) method [11] and Jaesik et al. [21], the improvement in facial color is small, and the regions around the clothes are discolored. However, in our method (e), color correction is successfully done over the entire face, and look more natural. From the above results, the proposed process gives satisfactory results despite the automatic processing.

Figure 13 shows a comparison with a background replacement result of the original image and [26]. The background replacement result has a distorted facial skin color due to the background color. Shin et al.’s method transfers a style from an input image to a target image. Although the facial skin color of the result is almost the same as the target image one, this method cannot correct skin color of a person wearing glasses to the skin color of the target image. Our algorithm generates an image which has the target facial skin color even if a person wears glasses.

4.3 Semi-automatic color correction

Our method can correct other images, such as animal photographs. Our color correction method requires some regions, which are $\Omega_S$ as a foreground and $\Omega_B$ as a background. Many methods have been proposed to detect face regions, but accurate object detection in natural scenes is still a challenging problem.

For color correction in natural scenes such as Fig. 14(c), we make each region manually as (b) and (d), where the colors indicate the region corresponding to Fig. 3, respectively. Finally, our method adjusts the object color of the source image to the target one automatically. Figure 14(e) shows natural scene color correction results and the red box shows that our method corrects the colors of main object without boundary artifacts. By implementing an automatic object detection method, this application will be an auto-application.

4.4 Limitation

Figure 15 shows example images where our method does not work well. The face and hair are a similar color, hence, the facial skin color extraction process gets regions other than the face region (Fig. 15(b)). As a result, our method outputs an image with a similar color between the face and hair (Fig. 15(c)). Actually, the facial skin color extraction process seldom gets other regions even if the subject does not have white skin, so the guide image filtering results in the same color between the face and the other extracted regions.

5 Conclusions

This paper presents a guided facial skin color correction method. The color grading method used in the procedure can be thought of as a correction method using a target (guide) image in a color space. On the other hand, our GIF can be thought of as a correction method using a guide image in an image space. In future work, we will consider combining both methods to extend the range of constraint expressions to color correction so as to obtain more natural results.

Appendix A: Fore/Background segmentation by matting

To obtain the alpha-mats that give natural blending results, we employ a closed form matting method [13] because the algorithm is also used in Sec. 3.4 and can be diverted. The method is based on continuous optimization in which labels are obtained as real numbers (soft labels) in the range of $[0, 1]$. However this method requires semantics (user assisted information) for fore/background regions around their boundaries. In order to avoid user assistance, we also employ a region growing method used in an earlier matting method [27] and perform matting iteratively. The details are shown in Fig. 16 and described as follows:

(a) As the initial foreground $\Omega_F$, in addition to the skin color region $\Omega_S$, the hair region above the face (a black large region is roughly selected), and the clothes region below the face are used. As the initial background $\Omega_B$, two rectangular regions on the left and right sides of the face are used.

(b) Matting [13] with the preconditioning conjugate gradient method is performed. A soft label $\alpha_i \in [0, 1]$ is obtained at each pixel.
Figure 12: Comparison with the existing methods. (a) Target, (b) Original, (c) [24] with [23], (d) NRDC [11], (e) Jaesik et al. [21], and (f) Our method.

Figure 13: Comparison with the background replacement result and [26]. (a) Target facial skin color image, (b) Original images, (c) The background replacement result as a example, (d) The style transfer result by [26], and (e) The yearbook style photo using our algorithm.
Figure 14: Semi-automatic color correction. (a) Target, (b) Each region in target image, (c) Source, (d) Each region in source image, and (e) Result. Colors in (b) and (d) indicate each region as Fig. 3. The source and target images are available at https://pixabay.com.

Figure 15: Bad results of our method due to the skin color extraction phase. In the original image the face and hair are a similar color.

(c) The pixels that are strongly regarded as foreground or background are added to the initial region, which is generated in (a), for the next iteration: $\Omega_B := \{ i \mid \alpha_i \leq 0.2 \}$, $\Omega_F := \{ i \mid 0.8 \leq \alpha_i \}$ and then $\{ \alpha_i := 0 \mid i \in \Omega_B \cup \Omega_B^+ \}$, $\{ \alpha_i := 1 \mid i \in \Omega_F \cup \Omega_F^+ \}$.

(d) Steps (b) and (c) are repeated a few times (4 times, in our experiment). The radius of the window size is halved to implement a coarse-to-fine approach.

(e) To reduce neutral colors and enhance them to be close to 0 or 1, a sigmoid function is applied to the alpha-mat:

$$\alpha_i := \frac{1}{1 + \exp(-10(\alpha_i - 0.5))}.$$ 

Figure 16: Image matting with region growing.

References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Image Process. (TIP)*, 20(3):681–695, Mar. 2011.

[2] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proc. Annual ACM-SIAM Symp. Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] Tatsuya Baba, Paul Perrotin, Yusuke Tatesumi, Keiichiro Shirai, and Masahiro Okuda. An automatic yearbook style photo generation method using color grading and guide image filtering based facial skin color. In *Proc. IAPR Asian Conf. Pattern Recognit. (ACPR)*, pages 1–6, 2015.

[4] Narze Batool and Rama Chellappa. Detection and inpainting of facial wrinkles using texture orientation
fields and Markov random field modeling. IEEE Trans. Image Process. (TIP), 23(9):3773–3788, 2014.

[5] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Process. (TIP), 18(11):2419–2434, 2009.

[6] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu. Epigraphical projection and proximal tools for solving constrained convex optimization problems. Signal, Image Video Process., 9(8):1737–1749, Nov. 2015.

[7] P. L. Combettes and J. C. Pesquet. Image restoration subject to a total variation constraint. IEEE Trans. Image Process. (TIP), 13(9):1213–1222, Sept. 2004.

[8] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. ACM Trans. Graph. (TOG), 23(3):673–678, 2004.

[9] J. M. Fadili and G. Peyré. Total variation projection with first order schemes. IEEE Trans. Image Process. (TIP), 20(3):657–669, Mar. 2011.

[10] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pages 1–8, 2008.

[11] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. ACM Trans. Graph. (TOG), 30(4):70:1–70:10, 2011.

[12] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. Optimizing color consistency in photo collections. ACM Trans. Graph. (TOG), 32(4):38:1–38:10, 2013.

[13] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting Laplacian matrices. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pages 2165–2172, 2010.

[14] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 35(6):1397–1409, 2013.

[15] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. ACM Trans. Graph. (TOG), 26(3):96:1–96:5, 2007.

[16] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. ACM Trans. Graph. (TOG), 23(3):689–694, 2004.

[17] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 30(2):228–242, 2008.