

## Article

# Guided Facial Skin Color Correction

Keiichiro Shirai <sup>1,\*</sup>, Tatsuya Baba <sup>2,†,‡</sup>, Shunsuke Ono <sup>3</sup>, Masahiro Okuda <sup>4</sup>, Yusuke Tatezumi <sup>1,§</sup>  
and Paul Perrotin <sup>5,||</sup>

- <sup>1</sup> Faculty of Engineering, Shinshu University, 4 Chome-17-1, Wakasato 380-0928, Japan; ysktatesumi@gmail.com
- <sup>2</sup> Faculty of Environmental Engineering, The University of Kitakyushu, 1-1 Hibikino, Wakamatsu Ward, Kitakyushu, Fukuoka 808-0135, Japan; tatsuya.baba@oppo.com
- <sup>3</sup> Tokyo Institute of Technology, G3-52 (Room 922), 4259 Nagatsuta-cho, Midori-ku, Yokohama 226-8503, Japan; ono@c.titech.ac.jp
- <sup>4</sup> Faculty of Science and Engineering, Doshisha University, Kamigyō Ward, Kyoto 602-8580, Japan; masokuda@mail.doshisha.ac.jp
- <sup>5</sup> University Institute of Technology La Rochelle, 17000 Charente-Maritime, France; paulwanadoo@hotmail.fr
- \* Correspondence: keiichi@shinshu-u.ac.jp
- † The first two authors contribute equally to the work.
- ‡ Current address: OPPO Japan.
- § Current address: Toyota motor corporation.
- || Current address: Chair of Naval Cyber Defense, Ecole Navale, 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). This color distortion is emphasized when artificially synthesized with another background color, and the appearance becomes unnatural. In our framework, we, first, roughly extract the face region and rectify the skin color distribution in a color space. Then, 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. Our color correction process attains natural results by using a guide image, unlike conventional algorithms. 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 another application.

**Keywords:** auto correction; color correction; color grading; face recognition; guide image filtering



**Citation:** Shirai, K.; Baba, T.; Ono, S.; Okuda, M.; Tatezumi, Y.; Perrotin, P. Guided Facial Skin Color Correction. *Signals* **2021**, *2*, 540–558. <https://doi.org/10.3390/signals2030033>

Academic Editor: Felix Albu

Received: 11 May 2021

Accepted: 12 August 2021

Published: 24 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Portrait photographs sometimes get undesirable color due to a background color reflection. In particular, changes in skin color sometimes give a bad impression, so that professional cameramen arrange lighting conditions to make the skin color and brightness uniform for each photograph. Meanwhile, for an amateur, more effort is required to achieve the same quality. Therefore, there is a need for a simple and automatic method not requiring any special technique to unify 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, producing a different distribution of skin color in a color space.
- Background color is usually reflected to faces causing distorted colors.
- When each subject wears different colored clothes, color correction for the whole image region distorts the skin color, while skin color correction discolors the clothes.

To address the problems, we need a color correction scheme that affects only the face region.

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 without much effort or cost. If ones try to gather photographs taken with different cameras in different lighting conditions, 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 Section 2.

In addition to the algorithm described in this paper, we propose a hybrid *guided image filtering* (GIF) method (the red box in Figure 1), which performs color transfer on the local face region while keeping colors in other regions, making GIF suitable for the correction of facial skin color and brightness. 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. The hybrid GIF transforms a part of the image as an object based color transfer. In contrast to the fact that the original GIF [1] 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 [2] and matting [3]. 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 [4].

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

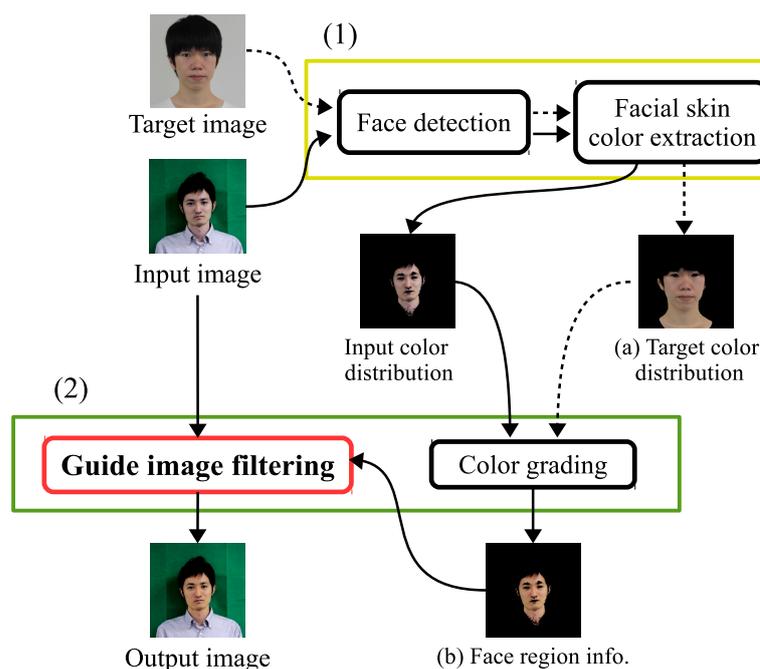


Figure 1. Flow chart of our method.

## 2. Related Work

Reference based color correction is one of the essential research issues for image editing. *Color grading* methods [5–7] are usually used for color correction among multiple images. The method matches the color attributes of the input image, e.g., color distribution and tone curve for each color, with those of the target (reference) image. These existing methods, however, are for global image correction, and not suitable for local face area correction, and also assume that subjects wear similar color clothes. *Colorization* methods [2,8] are available for this purpose. By specifying coloring information for pixels, the method changes colors of a subject. However, the method requires a lot of coloring information to obtain natural results. Another possible approach is to apply color transfer based on GIF with guide images [1,9–11], where coloring information is collectively specified as a guide image. GIF, however, requires a perfectly aligned pair of input and guide images without any position gap, which is often not the case for the situation considered.

As for facial image correction methods, a wrinkle removal method is proposed in [12], but the purpose differs from color grading of facial skin color. The method proposed in [13] 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. Figure 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 Figure 1): It detects the face area and extracts its facial skin color.
- (2) Skin color correction (green box): The facial skin color extracted is corrected using the target image (a) in the color space. Then, the color of the face region is corrected using the image (b) as the guide image in the image space.

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. *Facial skin color* is the color of the skin in the face region only. Additionally, *region* and *area* are distinguished. *Area* refers to the detected face area in Section 3.1. *Region* refers to the extracted facial skin color region in Section 3.2.

The novelty of the proposed method lies in (2), where the color correction with grading affects only a part of the image. The other steps consist of conventional methods with some modifications. Hereafter, we describe each procedure in more detail.

Incidentally, the accuracy of the conventional methods in (1) is sufficient for most of the images shown in this paper but sometimes affects the quality of the resulting images, and use of recent methods may improve the quality of the resulting images. We show that in Section 4.4 and Figure 16 as Limitations.

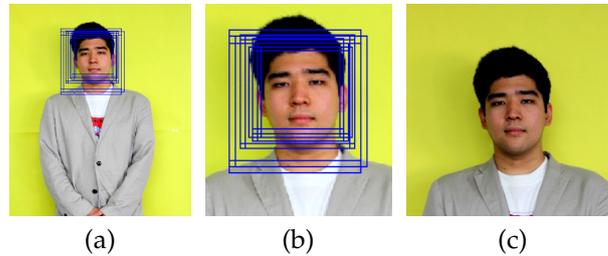
### 3.1. Face Detection

This step detects an area from head to shoulder (see Figure 2) by using the Viola–Jones algorithm [14], and the implementation in OpenCV as Haar-cascade detection for the face detection. We also examined the face detector in Dlib [15] and the face parsing methods based on deep learning, e.g., BiSeNet [16]. However, we chose the Viola–Jones method [14] because the detected data and parameters can be easily used, and the accuracy is sufficient if the face is facing closer to the front. The face area is described by  $n$  rectangle windows as shown in Figure 2b, where  $n$  is the number of candidates. As the parameters of each candidate, the barycentric coordinates  $(x, y)$  and the width and height of a detected rectangle  $(w, h)$  are obtained. We find the median of  $x, y, w, h$  as

$$\begin{aligned}\hat{x} &:= \text{median}(\{x_1, \dots, x_n\}), & \hat{w} &:= \text{median}(\{w_1, \dots, w_n\}), \\ \hat{y} &:= \text{median}(\{y_1, \dots, y_n\}), & \hat{h} &:= \text{median}(\{h_1, \dots, h_n\}),\end{aligned}\quad (1)$$

where the subscript is the index of the candidate. As the final face area, the rectangle with the median values of the barycentric coordinate  $(\hat{x}, \hat{y})$  is adopted. Finally, the size of the face area is adjusted as follows using the size of the face area  $(\hat{w}, \hat{h})$  and the prescribed scale factor  $l$ :

$$(2l \hat{w} + 1) \times (2l \hat{h} + 1). \tag{2}$$



**Figure 2.** Face detection. (a) original image; (b) detected face area (candidate rectangles); and (c) face area in the original image.

### 3.2. Skin Color Extraction

This step roughly extracts the skin color region near the face. We use a common approach using clustering in the color space and morphological labeling in the image space:

- (i) The color of each pixel is classified by the color distribution of the entire image in the HSV color space. Each pixel is assigned the label of the cluster to which it belongs.
- (ii) Some regions are generated by concatenating neighboring pixels with the same labels in the image space. Regions mainly in the detected face area  $\Omega_{\text{rect}}$  (Equation (2)) are extracted, and regarded as the facial skin region.

As for the method for clustering color distributions, since the skin color in the detected face area (Section 3.1) is known, simpler approaches, e.g., *k-means* method [17] may give satisfactory results. We use *k-means++* [18] to select stable seed clusters.

We use the above *k-means* clustering [18] in this paper. Additionally, for robustness to *value* (luminance) and *saturation* variation and also simplicity, we only use the *hue* values for clustering the 1D distribution. The aforementioned procedures (i) and (ii) are performed to detect the face region  $\Omega_S^{\text{hue}}$ . Then, at each pixel contained in the region  $\Omega_S^{\text{hue}}$ , the *saturation* and *value* components are thresholded. The condition for the skin color at pixel  $p$  is experimentally defined as follows, and the skin color region  $\Omega_S$  is obtained:

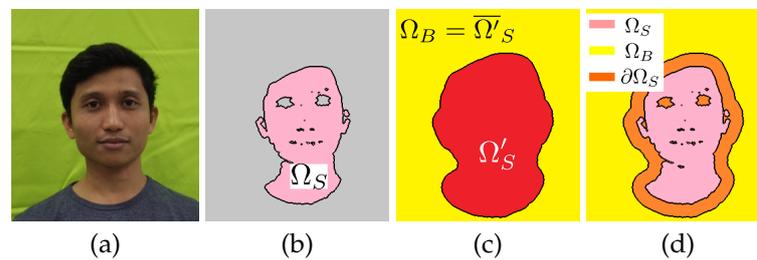
$$\Omega_S := \left\{ p \mid p \in \Omega_S^{\text{hue}}, s_p \in [\hat{s} - 0.2, \hat{s} + 0.2], v_p \in [0.15, 0.95] \right\}, \tag{3}$$

where  $\hat{s}$  is the median of the *saturation* in the face area. The *saturation* and the *value* are normalized to be in the range  $[0, 1]$  and the threshold values are set to excellently extract the facial skin color region on our dataset, which is available on the project page. The number of clusters depends on the photographic environment, especially if the skin color region needs to be perfectly extracted to the edge of the face because the edge color tends to change due to light reflection. However, the post-processing can handle this even in the case of rough extraction. We experimentally set  $k = 4$  in this paper.

For our guide image filtering, we define  $\Omega_B$  and  $\Omega_{\partial S}$  using the dilation function, which is a type of morphological operation. They are given as

$$\begin{aligned} \Omega'_S &:= \text{dilation}(\Omega_S), \\ \Omega_B &:= \overline{\Omega'_S}, \\ \Omega_{\partial S} &:= \Omega'_S \setminus \Omega_S, \end{aligned} \tag{4}$$

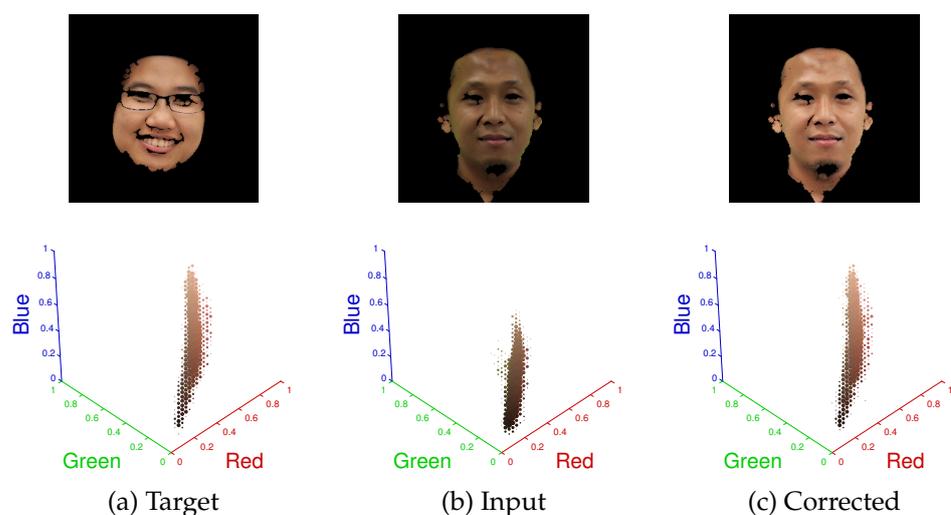
where  $\text{dilation}(\cdot)$  is the dilation function with a structuring element, which consists of a circle with a 20 pixel radius. The overline  $\bar{\cdot}$  is the complement and the operator  $\setminus$  is the set difference. An example of each region can be seen in Figure 3.



**Figure 3.** Region example. (a) input image; (b) skin color region  $\Omega_S$ , (c) dilated skin color region  $\Omega'_S$ , and (d) each region in the whole image. Pink, red, yellow, and orange indicate each region  $\Omega_S$ ,  $\Omega'_S$ ,  $\Omega_B$ , and  $\Omega_{\partial S}$ , respectively.

### 3.3. Color Grading

Figure 4 shows an example of color grading. Color grading brings the skin color in (b), the extracted region  $\Omega_S$ , close to the target facial skin color in (a). More specifically, this method transforms the shapes of the color distributions, i.e.,  $(r, g, b)$  3D coordinates, in the RGB color space. 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 [5] 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. As a simpler technique, one may use a 1D tone curve to transfer each RGB component between the two images (see Section 3.1 in [5]). However, to improve the accuracy, most of the methods require pixel correspondences between the images [6], which is unsuitable in our case that handles the faces of different persons. As a method that does not require the correspondences, we employ Pitié et al.’s method [5]. The method realizes nonlinear transformation of the color distribution by iteratively changing the width of the color distribution along a particular axis chosen in each iteration so that the shape of the marginal distribution projected onto the axis matches that of the target distribution. The method, however, yields some artifacts, e.g., blurring artifacts around the edges, and requires a correction method such as joint/cross filtering as used in [11]. As the correction method, we describe our GIF in the next section.

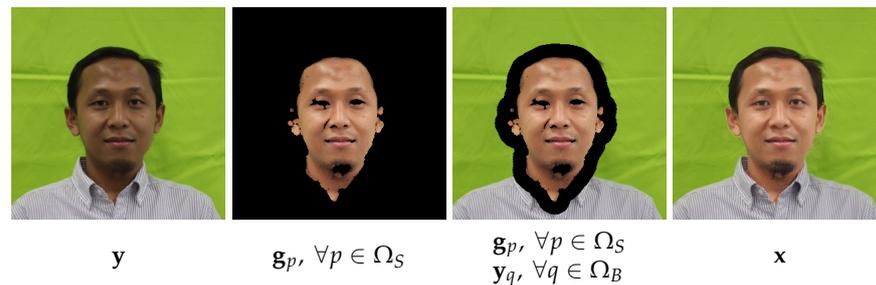


**Figure 4.** Each extracted facial skin color region and their distributions. The top row shows (a) target color image; (b) input image; and (c) color graded image. The bottom row shows their color distributions in the RGB color space.

### 3.4. Guide Image Filtering via Optimization

GIF is a sort of joint filtering (also called cross filtering) that uses a guide image to correct the input image. The energy function of GIF is based on the image matting method [3]. The data fidelity term is also known as the *local linear model*, and we focus on the data fidelity term, but the design objective is different. As a similar method, Ref. [19] has been proposed in high dynamic range imaging.

Our guide image filtering reconstructs an image  $\mathbf{x} \in \mathbb{R}^{3N}$  where the facial skin color region has a corrected color by using the input image  $\mathbf{y} \in \mathbb{R}^{3N}$  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  $\mathbf{x}_j$ , the input image be  $\mathbf{y}_j$ , and the guide image be  $\mathbf{g}_j$  which are given by the color grading in Section 3.3. Figure 5 shows each image.



**Figure 5.** Guide image filtering. From left: Input image, Guide image, Target color in  $\Omega_B$  and  $\Omega_S$ , and Color transformed image.

Using them, we formulate our guide image filtering as the following convex optimization problem:

$$\min_{\mathbf{x}, \mathbf{A}, \mathbf{b}} \underbrace{\sum_i \left( \sum_{j \in w_i} \|\mathbf{x}_j - \mathbf{A}_i \mathbf{y}_j - \mathbf{b}_i\|_2^2 + \varepsilon \|\mathbf{A}_i\|_F^2 \right)}_{\text{data fidelity}} + \underbrace{\iota_{\mathcal{C}_S}(\mathbf{x}) + \iota_{\mathcal{C}_B}(\mathbf{x})}_{\text{constraints}}. \tag{5}$$

First, the data fidelity terms, which correspond to [3], reflect textures and local contrasts of  $\mathbf{y}$  onto  $\mathbf{x}$ . As for the symbols,  $\mathbf{A}_i \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{b}_i \in \mathbb{R}^3$  are a scaling matrix and an offsetting vector to approximate  $\mathbf{y}_j$  to  $\mathbf{x}_j$  for each pixel  $i$ .  $w_i$  is a square window around pixel  $i$ .  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the  $\ell_2$  norm and Frobenius norm, respectively. Next, the constraint terms are expressed by the following indicator function and conditions:

$$\iota_{\mathcal{C}}(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{cases} \tag{6}$$

$\mathcal{C}_S$  and  $\mathcal{C}_B$  are conditions and given by

$$\begin{aligned} \mathcal{C}_S &:= \left\{ \mathbf{x} \mid \sqrt{\sum_{p \in \Omega_S} \|\mathbf{x}_p - \mathbf{g}_p\|_2^2} \leq \eta_S \right\}, \\ \mathcal{C}_B &:= \left\{ \mathbf{x} \mid \sqrt{\sum_{q \in \Omega_B} \|\mathbf{x}_q - \mathbf{y}_q\|_2^2} \leq \eta_B \right\}. \end{aligned} \tag{7}$$

The term  $\iota_{\mathcal{C}_S}(\mathbf{x})$  works on the facial skin color region  $\Omega_S$ , and brings the facial skin color close to the guide image  $\mathbf{g}$ . The term  $\iota_{\mathcal{C}_B}(\mathbf{x})$  works on the background region  $\Omega_B$ , and keeps the color of the background the same as the original image  $\mathbf{y}$ . For reducing undesirable artifacts, we design the constraint that the summation of the color differences obtained 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  $\lambda \sqrt{\sum_{p \in \Omega_S} \|\mathbf{x}_p - \mathbf{g}_p\|_2^2}$ , 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 Section 4). Such advantages of the formulation have been addressed in the literature of image restoration based on convex optimization [20–26].

Among convex optimization algorithms, we adopt a *monotone version of the fast iterative shrinkage-thresholding algorithm* (MFISTA) [27] 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:

$$\min_{\mathbf{x}} f(\mathbf{x}) + h(\mathbf{x}), \tag{8}$$

where  $f(\mathbf{x})$  is a differentiable convex function with a Lipschitz continuous gradient and  $h(\mathbf{x})$  is a proper lower semicontinuous convex function. The problem (8) is calculated by the proximity operator that is defined by

$$\text{prox}_{\kappa h}(y) := \arg \min_x \kappa h(x) + \frac{1}{2} \|y - x\|_2^2, \tag{9}$$

where  $h(\cdot)$  is a proper lower semicontinuous convex function, and  $\kappa$  is the index. For given  $\mathbf{x}^0 =: \mathbf{v}^1 \in \mathbb{R}^N$  and  $t^1 := 1$ , the iteration of the MFISTA consists of the following five steps:

$$\left\{ \begin{array}{l} \hat{\mathbf{z}}^k = \mathbf{v}^k - \frac{1}{L} \nabla f(\mathbf{v}^k), \\ \mathbf{z}^k = \text{prox}_{h/L}(\hat{\mathbf{z}}^k), \\ \mathbf{x}^k = \arg \min \{ f(\mathbf{z}) + h(\mathbf{z}) \mid \mathbf{z} \in \{\mathbf{z}^k, \mathbf{x}^{k-1}\} \}, \\ t^{k+1} = \frac{1 + \sqrt{1 + 4(t^k)^2}}{2}, \\ \mathbf{v}^{k+1} = \mathbf{x}^k + \frac{t^k}{t^{k+1}}(\mathbf{z}^k - \mathbf{x}^k) + \frac{t^k - 1}{t^{k+1}}(\mathbf{x}^k - \mathbf{x}^{k-1}), \end{array} \right. \tag{10}$$

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

$$\begin{aligned} f(\mathbf{x}) &:= \sum_i \left( \sum_{j \in w_i} \|\mathbf{x}_j - \mathbf{A}_i \mathbf{y}_j - \mathbf{b}_i\|_2^2 + \varepsilon \|\mathbf{A}_i\|_F^2 \right), \\ h(\mathbf{x}) &:= \iota_{C_S}(\mathbf{x}) + \iota_{C_B}(\mathbf{x}). \end{aligned} \tag{11}$$

To compute the gradient  $\nabla f(\mathbf{x})$ , we use a method similar to [28], which is an accelerated version of [3]. We compute each value of  $[\nabla f(\mathbf{x})]_i \in \mathbb{R}^3$  as follows:

$$\mathbf{A}_i^* = \Delta_i^{-1} \left( \frac{1}{|w_i|} \left( \sum_{j \in w_i} \mathbf{y}_j \mathbf{x}_j^\top \right) - \bar{\mathbf{y}}_i \bar{\mathbf{x}}_i^\top \right), \tag{12}$$

$$\mathbf{b}_i^* = \bar{\mathbf{x}}_i - \mathbf{A}_i^* \bar{\mathbf{y}}_i, \tag{13}$$

$$[\nabla f(\mathbf{x})]_i = |w_i| \mathbf{y}_i - \left( \left( \sum_{j \in w_i} \mathbf{A}_j^* \right) \mathbf{y}_i + \left( \sum_{j \in w_i} \mathbf{b}_j^* \right) \right), \tag{14}$$

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

The proximity operator  $\text{prox}_{h/L}$  in (10) 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

$$\mathbf{z}_i := [\text{prox}_{h/L}(\hat{\mathbf{z}})]_i = \begin{cases} \hat{\mathbf{z}}_i + \eta_S \frac{\hat{\mathbf{z}}_i - \mathbf{g}_i}{\sqrt{\sum_{p \in \Omega_S} \|\hat{\mathbf{z}}_p - \mathbf{g}_p\|_2^2}} & \text{if } \hat{\mathbf{z}} \notin \mathcal{C}_S \wedge i \in \Omega_S, \\ \hat{\mathbf{z}}_i + \eta_B \frac{\hat{\mathbf{z}}_i - \mathbf{y}_i}{\sqrt{\sum_{q \in \Omega_B} \|\hat{\mathbf{z}}_q - \mathbf{y}_q\|_2^2}} & \text{if } \hat{\mathbf{z}} \notin \mathcal{C}_B \wedge i \in \Omega_B, \\ \hat{\mathbf{z}}_i & \text{otherwise.} \end{cases} \tag{15}$$

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 (10), 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. The experiment tests our algorithm on various sets, which are available from our website [http://wwell.cs.shinshu-u.ac.jp/~keiichi/projects/auto\\_album](http://wwell.cs.shinshu-u.ac.jp/~keiichi/projects/auto_album) (accessed on 14 August 2021). The range of RGB values is normalized to  $[0, 1]$ . The prescribed scale factor  $l$  in Section 3.1 is set to 2.0. The filter window sizes used in Section 3.4 and Appendix A are  $19 \times 19$  and  $31 \times 31$ , respectively.  $\eta_S = 5|\Omega_S| \times 10^{-4}$  and  $\eta_B = 5|\Omega_B| \times 10^{-10}$  are used, where  $|\Omega_{(\cdot)}|$  indicates the number of pixels contained in the region  $\Omega_{(\cdot)}$ .  $L$  in MFISTA is set to 500.

Our filtering often flattens the gradation of the input images caused by shadows. It yields unnatural images as shown in the top of Figure 6, where the input image is input 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  $\mathbf{x}_i^C \in \mathbb{R}^3$  and an intensity component  $x_i^I \in \mathbb{R}^1$  as follows:

$$\begin{aligned} x_i^I &= \frac{x_i^R}{\|\mathbf{x}_i\|_1} x_i^R + \frac{x_i^G}{\|\mathbf{x}_i\|_1} x_i^G + \frac{x_i^B}{\|\mathbf{x}_i\|_1} x_i^B, \\ \mathbf{x}_i^C &= \frac{\mathbf{x}_i}{x_i^I}, \end{aligned} \tag{16}$$

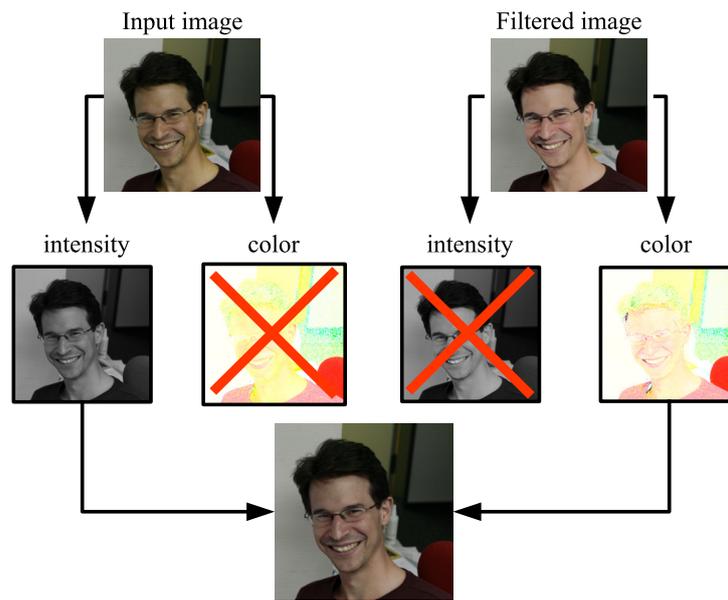
where the superscripts  $R, G$  and  $B$  indicate each color component. This decomposition procedure is the same as [10].

The input image and the filtered image are decomposed into the two components by (16), and then the intensity component of the input image  $y_i^I$  and the color component of the filtered image  $\mathbf{x}_i^C$  are combined as follows:

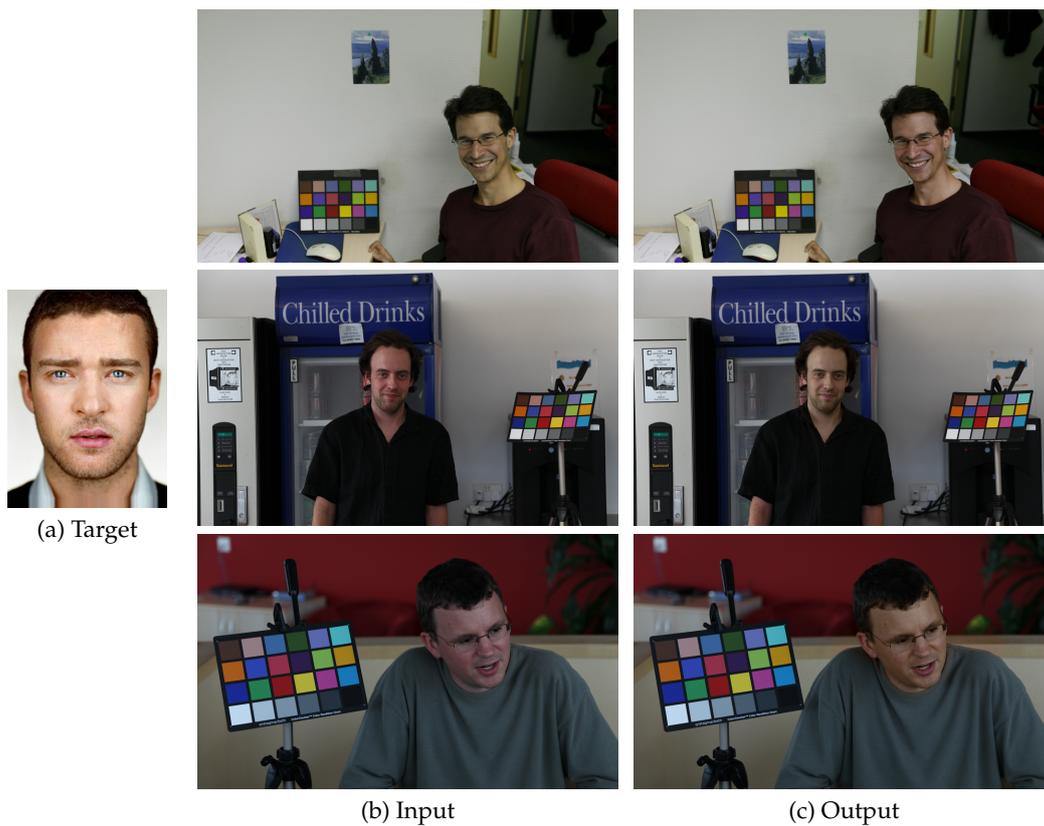
$$\mathbf{x}_i' = y_i^I \mathbf{x}_i^C. \tag{17}$$

Figure 6 shows this procedure and its effectiveness.

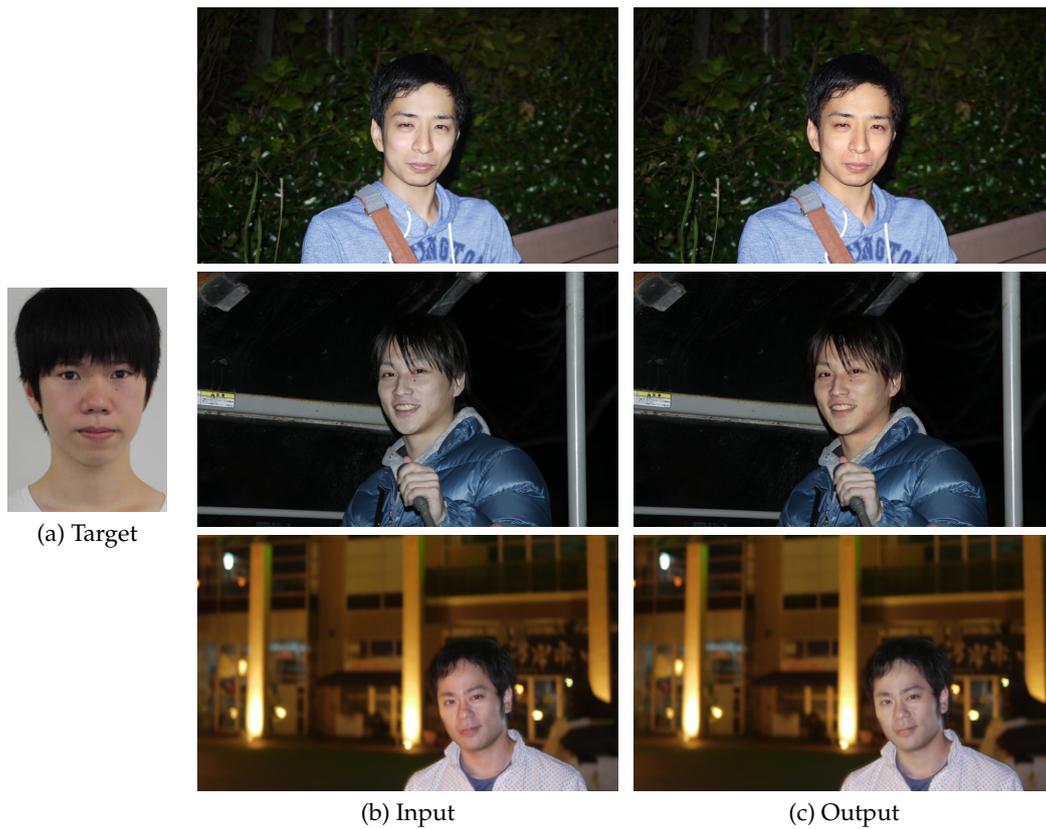
Figure 7 shows the result of our skin color correction. The area around the face in the input image has color distortion due to the lighting conditions and background color. Our result has a white balanced facial skin color similar to the target facial skin color. Sometimes, when we take a photograph in dark surroundings, the results sometimes have unnatural face colors (Figure 8b) 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 8c 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.



**Figure 6.** Simple luminance correction. Input image and filtered image are decomposed into intensity components and color components, respectively, by (16), then the intensity component of the input image and the color component of the filtered image are combined by (17).



**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 [13] and the input images are in the Gehler dataset [29].



**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.

#### 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 Figure 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 Figure 9).

##### 4.1.1. Face Area Cropping

After the face area detection in Section 3.1, we resize the cropped images so that the sizes are unified among images. The crop size is roughly adjusted according to the image size. The final image size is  $320 \times 320$  ( $h \times w$ ) in this experiment.

##### 4.1.2. Background Replacement by Alpha Blending

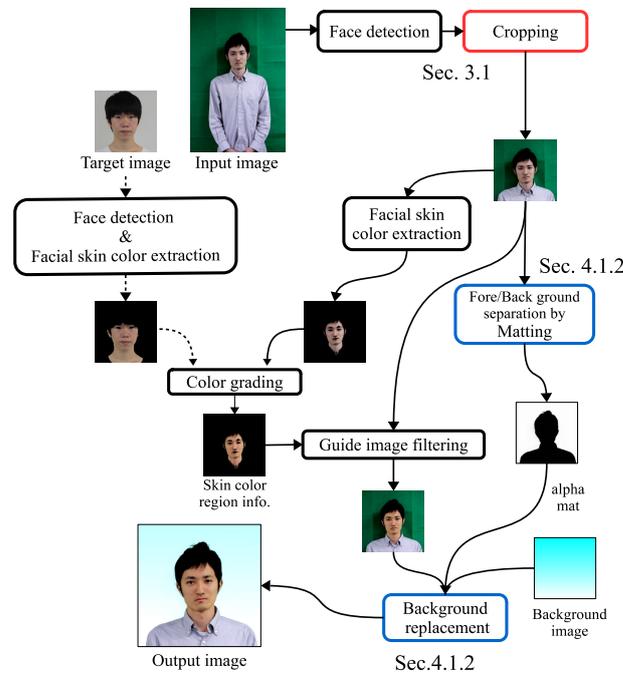
We separate the image information of the background and foreground regions and assign a value (soft label)  $\alpha_i := [0, 1]$  to them, where  $\alpha_i = 0$  and  $\alpha_i = 1$ , respectively, denote the background and the foreground. Additionally, we define foreground, background, and original image, respectively, as  $\mathbf{f}_i$ ,  $\mathbf{b}_i$ , and  $\mathbf{y}_i$  (different from that in (5)). The relationship among them is given as follows:

$$\mathbf{y}_i := \mathbf{f}_i + \mathbf{b}_i, \quad \mathbf{f}_i := (1 - \alpha_i)\mathbf{y}_i, \quad \mathbf{b}_i := \alpha_i\mathbf{y}_i. \quad (18)$$

Here,  $\alpha$  also works as the blending rate, and called *alpha-mat*. The background replacement with another background  $z$  is given by

$$y'_i := f_i + \alpha_i z_i. \tag{19}$$

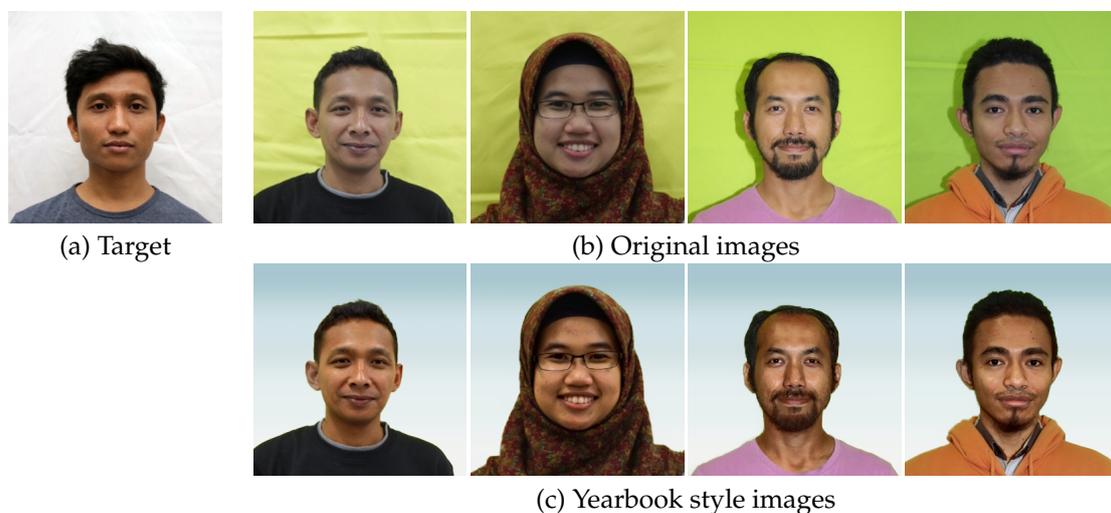
An estimation of the alpha-mats is described in Appendix A.



**Figure 9.** Flow chart of auto yearbook style photo generation. The red box indicates a pre-processing procedure, and the blue boxes indicate a post-processing procedure.

Figure 10 shows the results of our automatic yearbook style photo generation. Our algorithm generates the yearbook style photos from the original images using the target image.

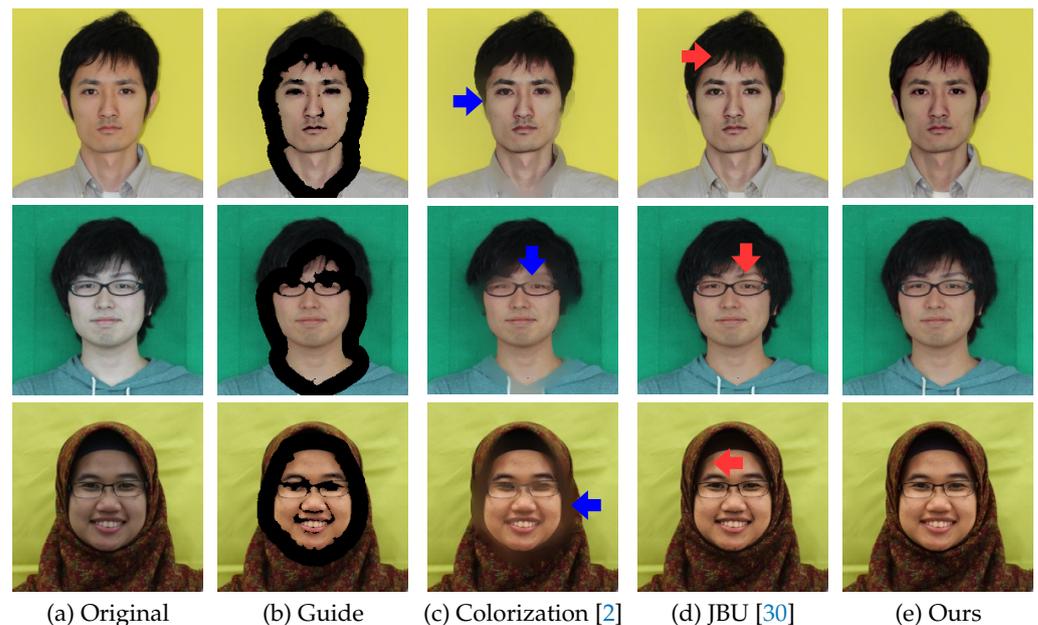
We implement the whole algorithm in MATLAB and OpenCV (C++), respectively, and the total execution time is within 11 s. on a 3.20 GHz Core i5 CPU, i.e., face detection (Section 3.1) is 5 s, facial color extraction (Section 3.2) is 1 s, color grading (Section 3.3) is 2 s, GIF (Section 3.4) is 2 s, and matting (Section 4.1.2) is 1 s, respectively.



**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.

#### 4.2. Comparison with Various Conventional Methods

Figure 11 compares our method with supplement methods [2,30]. For the methods, we process each RGB color layer because they are proposed for color components in YUV color space. Since colorization [2] just spreads colors to boundaries, the result does not have the details of the original image. The *joint bilateral upsampling* (JBU) [30] computes pixel values in hole regions by joint-filtering. We can see a contrast reduction in the top and the middle rows in Figure 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.

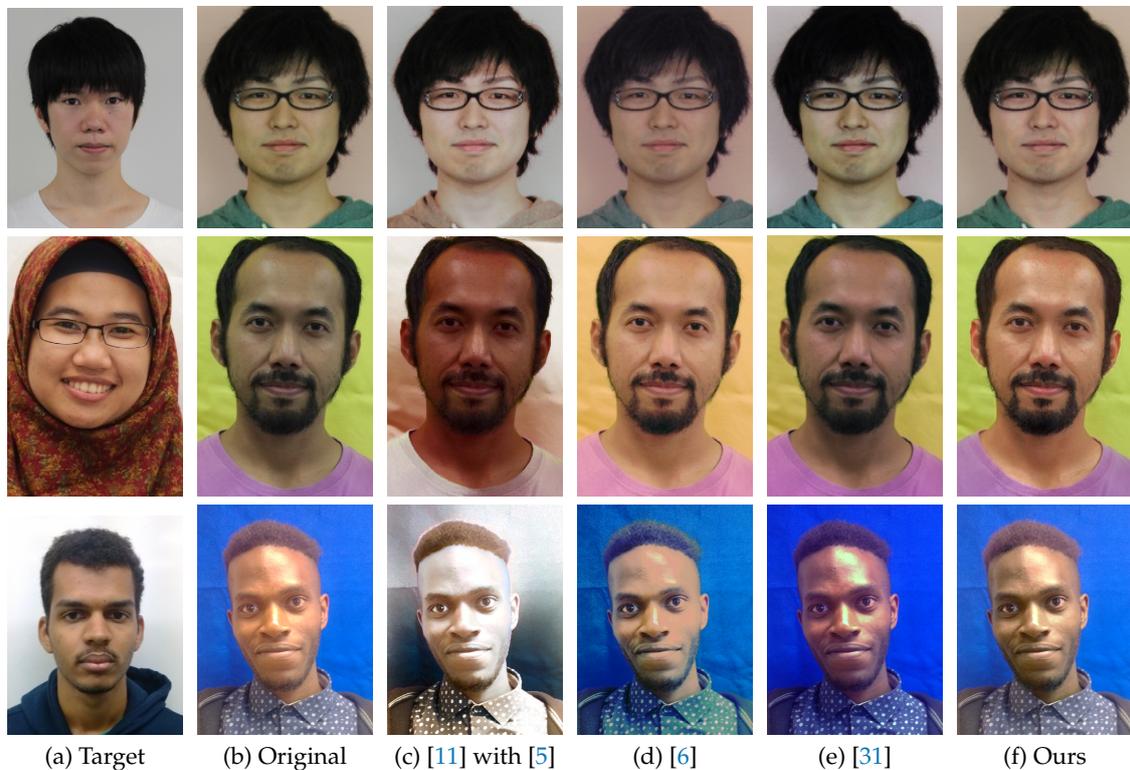


**Figure 11.** Supplement quality comparison with similar methods. (a) original; (b) guide; (c) colorization for each RGB layer [2]; (d) joint bilateral upsampling (JBU) for each RGB layer [30], and (e) our guided filtering (5). The black pixels in (b) represent a hole region, and each red arrow indicates an artifact.

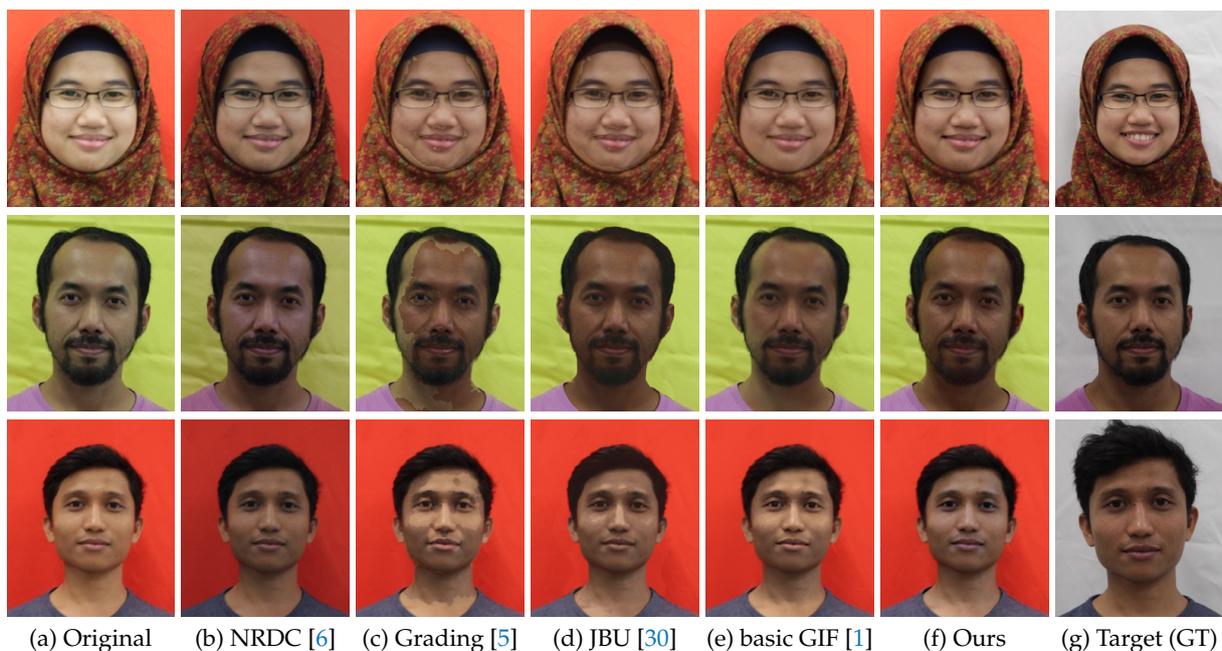
Figure 12 shows a comparison with existing color transfer methods. For the results (c), Pitié et al.'s method [5] transforms the color distribution of the input image to the target image one, then gain noise artifacts of the color transformed input image are removed by Rabin et al.'s method [11]. The gain noise artifacts can be seen in [5]. In the results of the color grading [11] with [5], 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 [6], the facial color is improved, but the regions around the clothes are discolored. In Jaesik et al.'s method [31], the improvement in facial color is small in some images. On the other hand, in our method (e), color correction is successfully done over the entire face while keeping the color except for the face, and the result looks more natural than those of the other methods.

Figure 13 shows an example in which the target image and the original image are the same person. We prepared the target image as the ground truth image in which the background color is white and the skin color is not affected by the background color. For the comparison methods, we selected the aforementioned NRDC [6] and JBU [30] (respectively, (b) and (d) in this figure) since they also gave good results in Figures 11 and 12. Additionally, (c) shows the result of color grading [5] performed in the first step by simply pasting on the original image. (e) shows the result using He et al.'s basic GIF [1] instead of our GIF, and the image (c) is used as the guide image in combination with the input image (a). In the results of NRDC (b), the skin color tends to include unnatural colors, and the

background color changes. The results of JBU (d) and basic GIF (e) are natural if the skin region can be extracted well. Otherwise, we need to manually set the processing window size for smoothing the edge where there is a color difference. Finally, one can see that our results (f) are more similar to the ground truth target images (g).



**Figure 12.** Comparison with the existing methods. (a) target; (b) original; (c) [11] with [5], (d) NRDC [6]; (e) Jaesik et al. [31], and (f) our method.



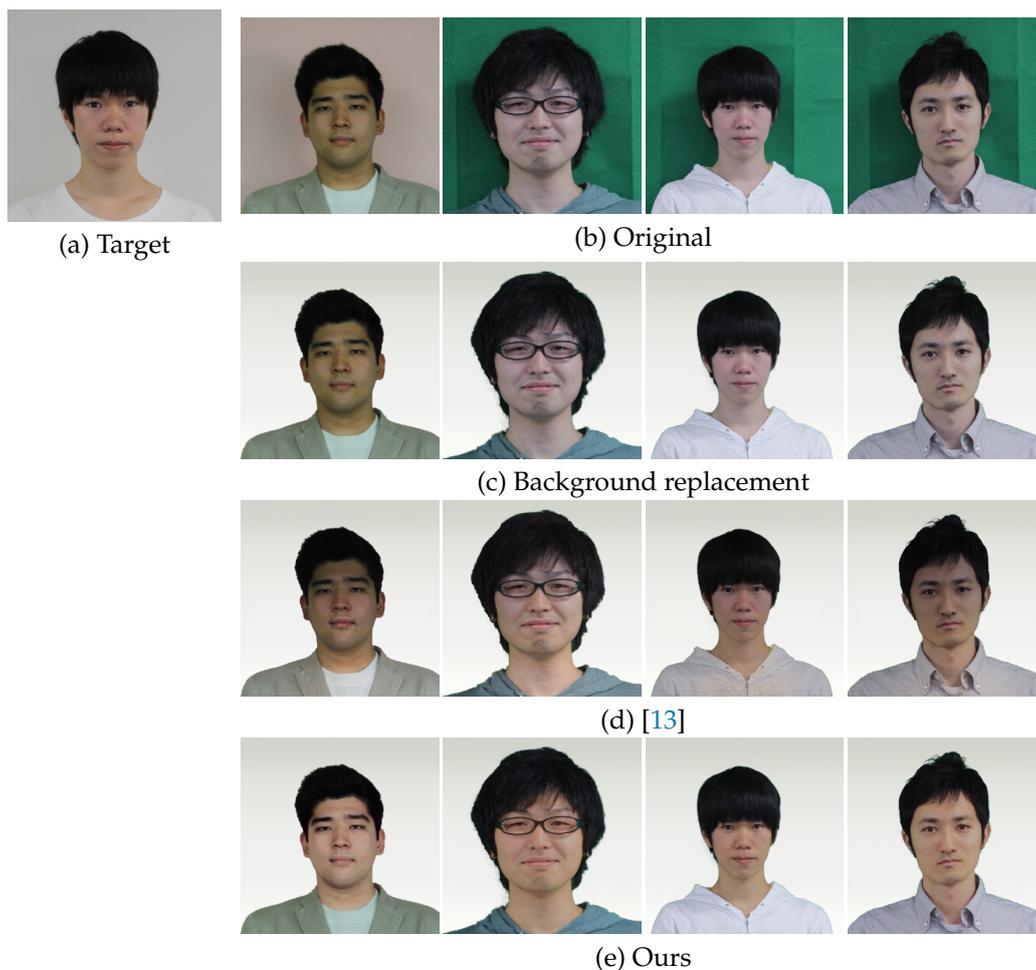
**Figure 13.** Skin color restoration. (a) original; (b) NRDC [6]; (c) color grading [5] used in our 1st step and the result simply pasted on (a); (d) JBU [30]; (e) He et al.'s basic GIF [1] instead of our GIF; (f) our method; and (g) target and ground truth.

Figure 14 shows a comparison with a background replacement result of the original image and [13]. 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 the 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.

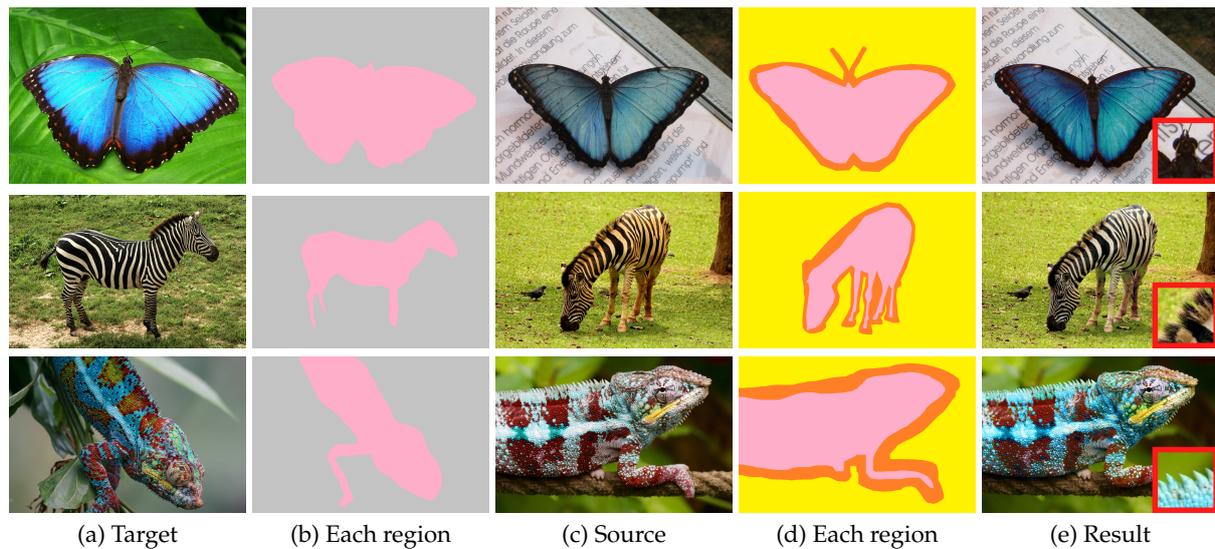
#### 4.3. Semi-Automatic Color Correction

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 Figure 15c, we make each region manually as (b) and (d), where the colors indicate the region corresponding to Figure 3, respectively. Finally, our method adjusts the object color of the source image to the target one automatically. Figure 15e shows natural scene color correction results, and the red box shows that our method corrects the colors of main objects without boundary artifacts. By implementing an automatic object detection method, this application will be an auto-application.



**Figure 14.** Comparison with the background replacement result and [13]. (a) target facial skin color image; (b) original images; (c) the background replacement result as a example; (d) the style transfer result by [13]; and (e) the yearbook style photo using our algorithm.



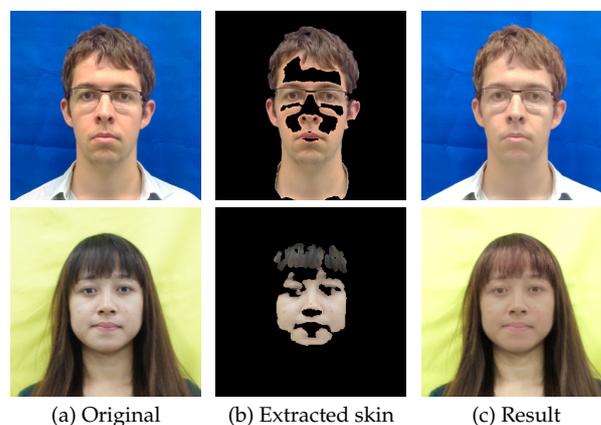
**Figure 15.** Semi-automatic color correction. (a) target; (b) each region in the target image; (c) source; (d) each region in source image, and (e) results. Colors in (b,d) indicate each region as Figure 3. The source and target images are available at <https://pixabay.com> (accessed on 3 August 2017).

#### 4.4. Limitations

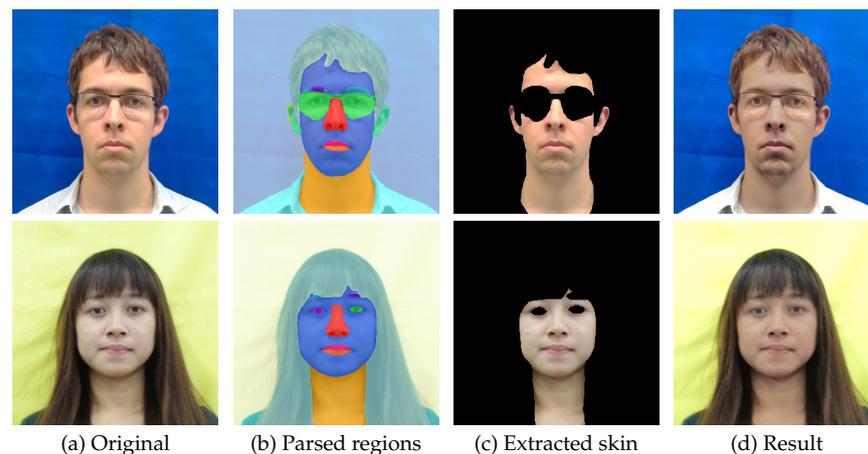
Figure 16 shows example images where our method does not work well. The face and hair are a similar color; hence, the facial skin color extraction process gets regions other than the face region (Figure 16b). As a result, our method outputs an image with a similar color between the face and hair (Figure 16c). 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.

On the other hand, we also show the good results in Figure 17 that are obtained by replacing the methods in facial skin color extraction with a face parsing method based on BiSeNet [16], and we use an implementation published on GitHub [32,33]. In the resulting images (d), the skin color and the hair color are correctly distinguished compared with the results in Figure 16c. Therefore, the facial skin color extraction process affects the quality of the final resulting images.

Incidentally, the aforementioned face parsing method was trained for the *CelebA* dataset [34], which mainly contains the area around the head, and the method does not handle the images shown in Figures 7 and 8, in which the head area is smaller than the image width. Therefore, another method to detect the face region is needed. For the experiment of the good results, we used the Viola–Jones method [14].



**Figure 16.** Bad results of our method due to the skin color extraction phase. In the original image the face and hair are a similar color. The target image is the image used in Figure 14a.



**Figure 17.** Good results of our method due to the skin color extraction phase. A face parsing method based on BiSeNet [16] is used. The target image is the image used in Figure 14a.

## 5. Conclusions

We proposed a facial skin color correction method in combination with color grading and guided image filtering. In our approach, we consider only the face region and extract this region and correct the color because, when looking at portrait photographs, we are firstly interested in the facial area, so correction focused on this area results in a good impression. The color grading step corrects the skin color using a target (guide) image in a color space, while the guided image filtering step performs the color correction in an image space. In the current method, these two steps are used in simple heuristic combination, but there is room for improving how to combine the two steps on the basis of mathematical optimization.

With regard to this, there are color segmentation methods called *soft color segmentation* [35,36], and the algorithm consists of color clustering and image filtering as well as our method. The algorithm framework is derived based on mathematical optimization and gives better resulting images than those of the heuristic combination of independent methods. On the basis of the methods, we try to revise the algorithm framework of the proposed method.

On the other hand, as for the part of facial skin detection and extraction, good results tend to be obtained by using face parsing methods based on deep learning. Therefore, we try to consider the combination with the methods and how to use them appropriately.

**Author Contributions:** Conceptualization, K.S.; methodology, K.S., T.B., S.O. and M.O.; software, K.S., T.B. and Y.T.; validation, K.S., T.B., Y.T. and P.P.; formal analysis, K.S., T.B., S.O. and M.O.; investigation, K.S. and T.B.; resources, K.S., T.B., Y.T. and P.P.; data curation, K.S. and T.B.; writing—original draft preparation, K.S. and T.B.; writing—review and editing, K.S., T.B., S.O. and M.O.; visualization, K.S. and T.B.; supervision, K.S. and M.O.; project administration, K.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The images used in the experiment are available from our website [http://wwell.cs.shinshu-u.ac.jp/~keiichi/projects/auto\\_album](http://wwell.cs.shinshu-u.ac.jp/~keiichi/projects/auto_album) (accessed on 14 August 2021).

**Acknowledgments:** The authors are grateful to David Ken Asano, Mia Rizkinia, and students Keisuke Iwata and Manabu Fujita for fruitful discussions about our method.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

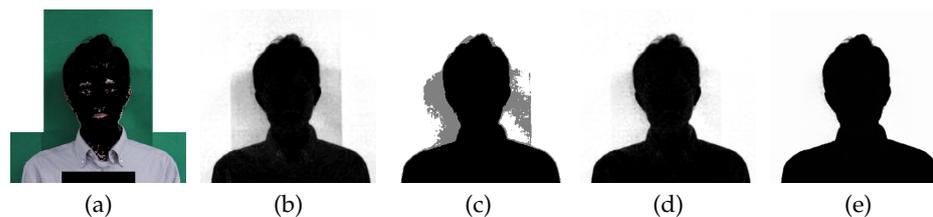
The following abbreviations are used in this manuscript:

GIF	Guided image filtering
MFIST	Monotone version of the fast iterative shrinkage-thresholding algorithm
NRDC	Non-rigid dense correspondence
JBU	Joint bilateral upsampling

## Appendix A. Fore/Background Segmentation by Matting

The alpha-mats used for alpha blending in Section 4.1.2 are generated using a *closed form matting* method [3] (we use its efficient calculation method [28]) in combination with a *region growing* method used in an earlier matting method [37]. This matting method is the base of GIF methods including our algorithm, and the calculation in Section 3.4 can partially be diverted. The labels representing the foreground and the background are obtained as real numbers (soft labels) in the range of  $[0, 1]$ . However, the method requires initial labels (referred to as a “trimap”) for fore/background regions around their boundaries. Here, the region growing method helps to spread the labels toward the boundary of the fore/background. The details are shown in Figure A1 and described as follows:

- The initial foreground  $\Omega_F$  is the combination of the skin color region  $\Omega_S$  and the hair region above the face (we roughly select a black large region). The initial background  $\Omega_B$  consists of two rectangular regions on the left and right sides of the face.
- Matting [28] is performed, giving a soft label  $\alpha_i \in [0, 1]$  for each pixel.
- Region growing in [37] is performed. The group of pixels strongly regarded as background  $\Omega_B^+ := \{i \mid \alpha_i \leq 0.2\}$  or foreground  $\Omega_F^+ := \{i \mid 0.8 \leq \alpha_i\}$  are, respectively, added to the initial regions  $\Omega_B \leftarrow \{i \in \Omega_B \cup \Omega_B^+\}$  and  $\Omega_F \leftarrow \{i \in \Omega_F \cup \Omega_F^+\}$ , for the next iteration.
- Steps (b) and (c) are repeated a few times (4 times, in our experiment). Additionally, we halve the process window size in matting to implement a coarse-to-fine approach.
- A *sigmoid function* is applied to the alpha-mat as  $\alpha_i := (1 + \exp(-10(\alpha_i - 0.5)))^{-1}$  to reduce neutral colors and enhance them to be close to 0 or 1.



**Figure A1.** Image matting with region growing. (a) Initial specification of the background (white color) and the foreground (black color). (b) is the result after the first iteration, and (c) is the result after region growing. (d) is the result after the second iteration, and (e) is the result after region growing.

## References

- He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2013**, *35*, 1397–1409. [[CrossRef](#)]
- Levin, A.; Lischinski, D.; Weiss, Y. Colorization using Optimization. *ACM Trans. Graph. (TOG)* **2004**, *23*, 689–694. [[CrossRef](#)]
- Levin, A.; Lischinski, D.; Weiss, Y. A Closed-Form Solution to Natural Image Matting. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2008**, *30*, 228–242. [[CrossRef](#)]
- Baba, T.; Perrotin, P.; Tatesumi, Y.; Shirai, K.; Okuda, M. An automatic yearbook style photo generation method using color grading and guide image filtering based facial skin color. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 1–6.
- Pitié, F.; Kokaram, A.C.; Dahyot, R. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst.* **2007**, *107*, 123–137. [[CrossRef](#)]

6. HaCohen, Y.; Shechtman, E.; Goldman, D.B.; Lischinski, D. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. (TOG)* **2011**, *30*, 70:1–70:10. [[CrossRef](#)]
7. HaCohen, Y.; Shechtman, E.; Goldman, D.B.; Lischinski, D. Optimizing color consistency in photo collections. *ACM Trans. Graph. (TOG)* **2013**, *32*, 38:1–38:10. [[CrossRef](#)]
8. Qiu, G.; Guan, J. Color by linear neighborhood embedding. In Proceedings of the IEEE International Conference on Image Processing 2005, Genova, Italy, 14 September 2005; Volume 3, pp. 1–4.
9. Petschnigg, G.; Szeliski, R.; Agrawala, M.; Cohen, M.; Hoppe, H.; Toyama, K. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. (TOG)* **2004**, *23*, 664–672. [[CrossRef](#)]
10. Eisemann, E.; Durand, F. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. (TOG)* **2004**, *23*, 673–678. [[CrossRef](#)]
11. Rabin, J.; Delon, J.; Gousseau, Y. Regularization of transportation maps for color and contrast transfer. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 1933–1936.
12. Batool, N.; Chellappa, R. Detection and inpainting of facial wrinkles using texture orientation fields and Markov random field modeling. *IEEE Trans. Image Process. (TIP)* **2014**, *23*, 3773–3788. [[CrossRef](#)]
13. Shih, Y.; Paris, S.; Barnes, C.; Freeman, W.T.; Durand, F. Style Transfer for Headshot Portraits. *ACM Trans. Graph. (TOG)* **2014**, *33*, 148:1–148:14. [[CrossRef](#)]
14. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. I-511–I-518.
15. King, D.E. Dlib-ml: A Machine Learning Toolkit. *J. Mach. Learn. Res.* **2009**, *10*, 1755–1758.
16. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *Computer Vision—ECCV 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 334–349.
17. Lloyd, S.P. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [[CrossRef](#)]
18. Arthur, D.; Vassilvitskii, S. K-means++: The Advantages of Careful Seeding. In Proceedings of the Annual ACM-SIAM Symp. Discrete Algorithms, New Orleans, Louisiana, 7–9 January 2007; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 1027–1035.
19. Shan, Q.; Jia, J.; Brown, M.S. Globally optimized linear windowed tone mapping. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 663–675. [[CrossRef](#)] [[PubMed](#)]
20. Combettes, P.L.; Pesquet, J.C. Image restoration subject to a total variation constraint. *IEEE Trans. Image Process. (TIP)* **2004**, *13*, 1213–1222. [[CrossRef](#)] [[PubMed](#)]
21. Fadili, J.M.; Peyré, G. Total variation projection with first order schemes. *IEEE Trans. Image Process. (TIP)* **2011**, *20*, 657–669. [[CrossRef](#)] [[PubMed](#)]
22. Afonso, M.V.; Bioucas-Dias, J.M.; Figueiredo, M.A.T. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Image Process. (TIP)* **2011**, *20*, 681–695. [[CrossRef](#)]
23. Teuber, T.; Steidl, G.; Chan, R.H. Minimization and parameter estimation for seminorm regularization models with  $I$ -divergence constraints. *Inverse Probl.* **2013**, *29*, 035007. [[CrossRef](#)]
24. Ono, S.; Yamada, I. Second-order total generalized variation constraint. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 4938–4942.
25. Chierchia, G.; Pustelnik, N.; Pesquet, J.C.; Pesquet-Popescu, B. Epigraphical projection and proximal tools for solving constrained convex optimization problems. *Signal Image Video Process.* **2015**, *9*, 1737–1749. [[CrossRef](#)]
26. Ono, S.; Yamada, I. Signal recovery with certain involved convex data-fidelity constraints. *IEEE Trans. Signal Process.* **2015**, *63*, 6149–6163. [[CrossRef](#)]
27. Beck, A.; Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process. (TIP)* **2009**, *18*, 2419–2434. [[CrossRef](#)]
28. He, K.; Sun, J.; Tang, X. Fast matting using large kernel matting Laplacian matrices. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2165–2172.
29. Gehler, P.V.; Rother, C.; Blake, A.; Minka, T.; Sharp, T. Bayesian color constancy revisited. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [[CrossRef](#)]
30. Kopf, J.; Cohen, M.F.; Lischinski, D.; Uyttendaele, M. Joint Bilateral Upsampling. *ACM Trans. Graph. (TOG)* **2007**, *26*, 96:1–96:5. [[CrossRef](#)]
31. Park, J.; Tai, Y.W.; Sinha, S.; Kweon, I.S. Efficient and Robust Color Consistency for Community Photo Collections. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 430–438.
32. CoinCheung. Implementatoin of BiSeNetV1 and BiSeNetV2. 2020. Available online: <https://github.com/CoinCheung/BiSeNet> (accessed on 8 July 2021).
33. zll. face-parsing.PyTorch. 2019. Available online: <https://github.com/zllrunning/face-parsing.PyTorch> (accessed on 8 July 2021).
34. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.

- 
35. Tai, Y.W.; Jia, J.; Tang, C.K. Soft color segmentation and its applications. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2007**, *29*, 1520–1537. [[CrossRef](#)]
  36. Aksoy, Y.; Aydin, T.O.; Smolić, A.; Pollefeys, M. Unmixing-based soft color segmentation for image manipulation. *ACM Trans. Graph.* **2017**, *36*, 19:1–19:19. [[CrossRef](#)]
  37. Sun, J.; Jia, J.; Tang, C.K.; Shum, H.Y. Poisson matting. *ACM Trans. Graph. (TOG)* **2004**, *23*, 315–321. [[CrossRef](#)]