



Optimization-based method removing highlights with flash/no-flash image

JONGWON CHOI,¹ SANGDOO YUN,² AND JIN YOUNG CHOI^{3,*}

¹*Dept. of Advanced Imaging, GSAIM, Chung-Ang Univ., Seoul, Republic of Korea*

²*Clova AI Research, NAVER, Republic of Korea*

³*ASRI, Dept. of Electrical and Computer Engineering, Seoul National Univ., Seoul, Republic of Korea*

**jychoi@snu.ac.kr*

Abstract: This paper proposes an optimization-based method to remove highlights from specular surfaces without any special device other than a flash, or any limited capturing environment required by existing methods. Accordingly, we formulate an optimization problem to estimate a highlight-free image from the difference image between a flash/no-flash image pair. This optimization will automatically find and recover erroneous pixels (saturated pixels and remaining highlights) in the difference image. The optimization is designed to recover erroneous pixels using the intensities of their neighboring pixels while retaining the other correct pixels to preserve the original image details. We evaluated the proposed method qualitatively and quantitatively through experimentation using datasets of synthetic and real-world scenes. In the experiments, the proposed method outperforms existing state-of-the-art methods.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

The reflected light intensity can be categorized into specular and diffuse intensities. Specular reflection causes highlights on the object's surface, which prevents humans from recognizing the photograph. Similarly, these highlights become obstacles for various computer vision algorithms such as segmentation, feature matching, and recognition. Because highlights are unavoidable in the real world, it is essential to remove them to improve the quality of practical photographs and the performance of computer vision algorithms.

Existing methods to remove highlights can be divided into multiple-image-based and single-image-based methods. Multiple image-based methods use multiple images captured by special devices or in complex capture settings such as a moving light source and various viewpoints [3,4]. The usage of special devices requires additional costs, and establishing the capture settings can be difficult in practical applications.

Single image-based methods detect the highlights and recover their diffuse color using only a single image [1,2]. Because single-image-based methods must separate two unknown variables of specular and diffuse intensities from only one-pixel value, the algorithms do not perform adequately in practice. Furthermore, single-image-based methods rely on a dichromatic assumption that the observed intensity equals the sum of the specular intensity and diffuse intensity. However, this assumption cannot be satisfied for saturated pixels that occur frequently in practice. Consequently, methods based on the dichromatic assumption may cause distorted colors, as represented by the blue boxes in Fig. 1(c and d). Furthermore, the colors with low chroma—like black or white—are prone to distortion due to the lack of color information. For example, when a pixel's original color is white, this can be recognized as a specular intensity, which suffers from color distortion, as represented by the red boxes in Fig. 1(c and d).

The method in this study uses a pair of flash and no-flash images. Because general cameras embed flash light, we can easily acquire flash and no-flash image pairs by capturing the consecutive frames where only the first image is irradiated by flash light. Due to its simple capturing condition, the flash/no-flash image pair has been widely used for computer vision algorithms [5,6], including

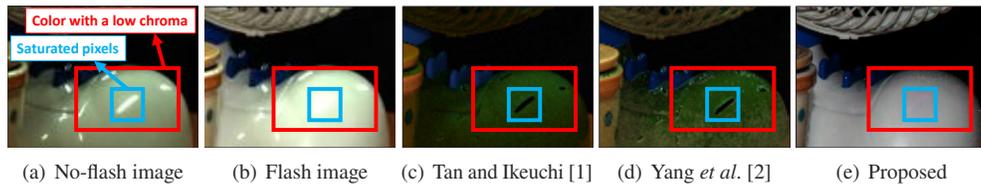


Fig. 1. Qualitative comparison of existing methods. Pixels in highlights can quickly become saturated. While existing methods [1,2] may cause color distortions in saturated pixels, the proposed method without dichromatic assumption solves problems.

image enhancement [7], shadow rejection [8], artifact removal [9,10], reflection removal [11], and foreground segmentation [12]. Agrawal *et al.* [9] used a flash and no-flash pair to remove reflections and highlights arising from the flash. However, this method cannot remove highlights arising from various light sources in the no-flash image, as discussed in Section 2.3. This study aims to remove highlights without the dichromatic assumption, even in saturated regions that occur frequently in practice.

We propose an optimization-based method that yields a highlight-free image to achieve our research purpose, as depicted in Fig. 2. This paper defines a *highlight-free image* as an image only affected by light flashes while not containing highlights caused by the light flashes; the highlight-free image differs from the diffuse image where all specular intensities are removed. For an initial image to estimate the highlight-free image, we use a *difference image* obtained by subtracting the no-flash image from the flash image. Highlights caused by complex natural lights can be removed using the difference image. However, the difference image cannot be highlight-free because the light flashes also yield highlights from specular objects and the colors are distorted on the pixels with saturated intensities.

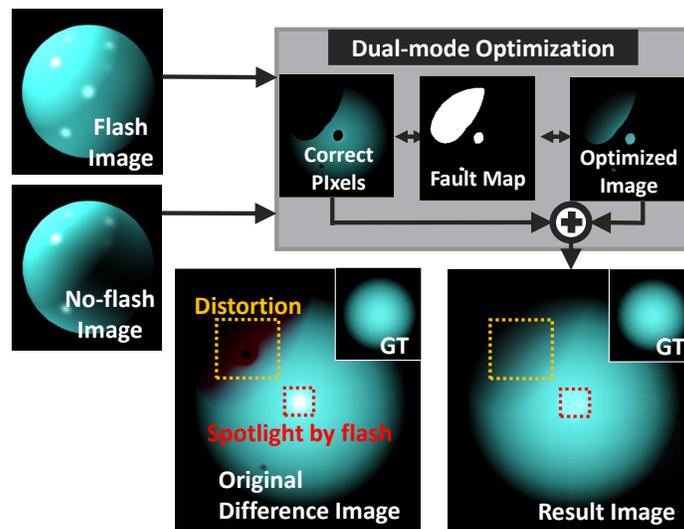


Fig. 2. Proposed highlight removal scheme. The proposed method removes the highlights with a pair of flash/no-flash images. Our method recovers distorted colors using a proposed optimization process.

In this paper, *faulty pixels* in the difference image refer to those that differ from the corresponding pixels in the differences between the difference image and the highlight-free image. We estimate

the highlight-free image from the difference image by formulating an optimization problem to recover faulty pixels from the difference image with three considerations. First, there is no exact clue to the locations of faulty pixels; we handled this issue by defining the binary fault map as an optimization variable to indicate the locations of the faulty pixels. Second, we restored the colors on the faulty pixels without the dichromatic model by establishing a cost term using the correct colors obtained from the neighbors of faulty pixels. Third, we maintained the correct colors for all pixels other than the faulty pixels by formulating an optimization constraint that enforces the correct pixels to maintain their original intensities given in the difference image.

The formulated optimization problem becomes non-convex because it includes the multiplication of a discrete variable (binary fault map) and a continuous variable (highlight-free image). We handled the non-convexity by applying an alternating optimization process by separating the variables via an auxiliary variable. We validated the performance of the proposed method using synthetic datasets and a real-scene dataset. We demonstrated the proposed method's generality by generating numerous synthetic images with various shapes, textures, and colors of illuminating lights. We verified the practical applicability of the proposed method using the results from the real-scene dataset.

2. Related works

2.1. Single-image-based highlight removal

Tan and Ikeuchi [1] proposed an algorithm to remove specularity by estimating the reflection basis functions from an input image. Shen and Cai [13] considered a smoothing constraint of the color transition along boundaries to obtain a specular-free image from a single image. Yang *et al.* [2] removed highlights using edge-preserving low-pass filters, assuming diffuse colors are similar to their neighbors. Tan *et al.* [14] separated highlights by considering the consistency of diffuse-textured characteristics among neighboring pixels. Kim *et al.* [15] estimated the region and values of highlights according to prior knowledge about dark channel values on the highlights. Suo *et al.* [16] proposed a normalized dichromatic model to estimate a specular-free image by separating identical diffuse colors. Mallick *et al.* [17] removed specularity by eroding the specular component iteratively with a partial differential equation. The recent studies using the deep learning algorithms have focused on the single-image-based highlight removals. Fu *et al.* [18] trained a multi-task deep network to detect and remove the specular highlight simultaneously. In [19], an efficient algorithm was proposed to remove the specular highlight using a pixel clustering mechanism.

2.2. Multiple-image-based highlight removal

Nayar *et al.* [3] used multiple images captured through a polarized lens by changing its polarization angle. Umeyama and Godin [20] also used multiple images with different polarization angles with polarized lenses, and they separated diffuse components using an independent component analysis method. Sato and Ikeuchi [4] separated specularity by analyzing the different pixel intensities obtained from multiple images captured with a moving light source. Lin and Shum [21] also captured multiple images with a moving light source and used linear basis functions to estimate the specular light components. Lee and Bajcsy [22] presented a specularity removal algorithm using multiple images taken from different viewpoints. Lin *et al.* [23] used multiple images of different viewpoints and filled highlights with diffuse values speculated from other images. Feng *et al.* [24] utilized a light-field camera to acquire the multiple images for the specular highlight removal. Due to the use of additional knowledge garnered from multiple views, the multiple-image-based methods outperform single-image-based methods. However, existing multiple-image-based methods are limited in practice because they require special devices and specific capture settings.

2.3. Flash/no-flash image-pair-based algorithms

Computer vision research has used various methods based on sets of multiple images such as micro-lenses [25], polarized lenses [3], multiple camera viewpoints [4,21,26], and controlled light sources [22,23,27]. This study concentrates on flash/no-flash image pairs, which can be taken without special devices or changing light source positions or camera viewpoints. Due to its simple capturing conditions, the flash/no-flash image pair has been widely used for computer vision algorithms, including image enhancement, shadow rejection, reflection removal, foreground segmentation [5–12]. Recently, the flash/no-flash image pairs are used for the albedo recovery [28,29]. In [30], the special deep-red light was used to obtain the light enhancement from the dark environments. Agrawal *et al.* [9] suggested a way of removing highlights (or reflections and other artifacts) induced from light flashes added to a no-flash image without highlights. Because the gradients of the two images differ the highlighted area, they used the difference to remove the highlight. However, if there were a highlight in the no-flash image, the gradients of the two images in the highlighted area would not change, so the highlight in the no-flash image could not be removed. Therefore, this method cannot eliminate highlights from no-flash images—purpose of our research. This study proposes an optimization approach to remove highlights from no-flash images using the characteristics of difference images between flash and no-flash images. Although Lei and Chen [31] proposed a algorithm to remove the reflection by using the flash-only image, the algorithm assumed that the target objects are ambient to be separated from the reflected layers. Thus, the reflection upon the target objects could not be removed, which differs from our approach.

3. Proposed method

3.1. Problem statement

When we suppose that there are neither saturated pixels nor highlights caused by light flashes, the highlight-free image \mathbf{F} is equivalent to the difference image \mathbf{D} . Thus, $\mathbf{F} = \mathbf{D} = \mathbf{I}_f^o - \mathbf{I}_{nf}^o$ is satisfied, where \mathbf{I}_f^o and \mathbf{I}_{nf}^o are the flash image and the no-flash image, respectively. However, the pixel intensities can become saturated due to the limited intensity ranges in practice, and additional highlights can arise from specular objects reflecting the light flash. The highlights caused by light flashes are referred to as *flash highlights* hereafter. Given the saturated pixels and the flash highlights, we can express the image pair as:

$$\begin{cases} \mathbf{I}_f^o = \mathbf{F} + \mathbf{G} - \mathbf{I}^L \\ \mathbf{I}_{nf}^o = \mathbf{G} - \mathbf{I}^{L'} \end{cases} \quad (1)$$

where \mathbf{G} is a natural image from all the lights other than a light flash and \mathbf{I}^L and $\mathbf{I}^{L'}$ are lost-intensity maps arising from the saturated pixels and flash highlights. Then, by subtracting \mathbf{I}_{nf}^o from \mathbf{I}_f^o , the difference map \mathbf{D} can be obtained as

$$\mathbf{D} = \mathbf{I}_f^o - \mathbf{I}_{nf}^o = \mathbf{F} - (\mathbf{I}^L - \mathbf{I}^{L'}). \quad (2)$$

Because the highlight-free image \mathbf{F} and the lost intensity maps (\mathbf{I}^L and $\mathbf{I}^{L'}$) are unknown, the problem becomes a blind signal separation problem in which \mathbf{F} can be determined from \mathbf{D} . We solved this problem by formulating an optimization problem and developing an alternating optimization process for handling the formulation's non-convexity.

3.2. Problem formulation

Our formulation normalizes the values in all images and intensity maps in the range of 0-1. In $(\mathbf{I}^L - \mathbf{I}^{L'})$ of Eq. (2), the elements corresponding to the correct pixels must be zero. Hence, it

is desirable to estimate only the elements for faulty pixels and determine the location of faulty pixels simultaneously during optimization. Accordingly, we introduce an additional optimization matrix \mathbf{M} masking faulty pixels, where \mathbf{M} is a binary fault map in which elements are 1 for faulty pixels and 0 for correct pixels. Then, $(\mathbf{I}^L - \mathbf{I}^{L'})$ can be replaced by the element-wise multiplication of \mathbf{M} and a lost intensity matrix \mathbf{L} . Equation (2) can be rewritten as

$$\mathbf{D} = \mathbf{F} - \mathbf{M} \circ \mathbf{L}, \quad (3)$$

where \circ is element-wise multiplication. In our formulation, $\mathbf{F} \in \mathbb{R}^{w \times h \times 3}$, $\mathbf{M} \in \mathbb{R}^{w \times h \times 1}$, and $\mathbf{L} \in \mathbb{R}^{w \times h \times 3}$ become optimization variables. According to Eq. (3), the values of \mathbf{L} can only change the values of \mathbf{F} for the faulty pixels; otherwise, the values of \mathbf{F} are equivalent to the values of \mathbf{D} for the correct pixels.

For representing the pixel-wise color vector of the matrix variables, $\mathbf{F}_{x,y}$ and $\mathbf{L}_{x,y}$ are defined as color vectors at the (x, y) -th element of \mathbf{F} and \mathbf{L} , respectively. $\mathbf{M}_{x,y}$ has a binary value (0 or 1) at the (x, y) -th element of \mathbf{M} . $\mathbf{F}_{x,y}$ at a correct pixel must be the same as $\mathbf{D}_{x,y}$ while $\mathbf{F}_{x,y}$ at a faulty pixel should differ from $\mathbf{D}_{x,y}$ due to $\mathbf{L}_{x,y}$. Hence, $\mathbf{M}_{x,y}$ depends on $\mathbf{F}_{x,y}$, which can be expressed by a function of $\mathbf{F}_{x,y}$, as

$$\mathbf{M}(\mathbf{F})_{x,y} = \begin{cases} 1 & , \text{if } \mathbf{F}_{x,y} \neq \mathbf{D}_{x,y} \\ 0 & , \text{if } \mathbf{F}_{x,y} = \mathbf{D}_{x,y}. \end{cases} \quad (4)$$

Furthermore, $\mathbf{L}_{x,y} = (\mathbf{F}_{x,y} - \mathbf{D}_{x,y})$ at the faulty pixel ($\mathbf{M}(\mathbf{F})_{x,y} = 1$) and $\mathbf{L}_{x,y} = \mathbf{0}$ at the correct pixel ($\mathbf{M}(\mathbf{F})_{x,y} = 0$). Hence, $\mathbf{L}_{x,y}$ can also be expressed by a function of $\mathbf{F}_{x,y}$, as

$$\mathbf{L}(\mathbf{F})_{x,y} = \mathbf{M}(\mathbf{F})_{x,y} (\mathbf{F}_{x,y} - \mathbf{D}_{x,y}). \quad (5)$$

By replacing \mathbf{M} and \mathbf{L} in Eq. (3) with Eqs. (4) and (5), Eq. (3) leads to

$$\mathbf{F}_{x,y} = \mathbf{D}_{x,y} + \mathbf{M}(\mathbf{F})_{x,y}^2 (\mathbf{F}_{x,y} - \mathbf{D}_{x,y}). \quad (6)$$

Because $\mathbf{M}(\mathbf{F})^2 = \mathbf{M}(\mathbf{F})$, Eq. (6) can be simplified as

$$(1 - \mathbf{M}(\mathbf{F})_{x,y}) (\mathbf{F}_{x,y} - \mathbf{D}_{x,y}) = 0, \quad (7)$$

which is an equality constraint in our formulation.

Our objective is to find \mathbf{F} that satisfies the equality constraint of Eq. (7), but the solutions are under-determined. We establish three optimization objectives to find a realistic solution. First, we regularize the lost-intensity estimation by minimizing $\|\mathbf{L}(\mathbf{F})_{x,y}\|^2$, where $\|\cdot\|$ is the l_2 norm. Through this regularization term, the over-fitting problem can be avoided and the original textures on the faulty pixels can be conserved by suppressing the redundant estimates of $\mathbf{L}(\mathbf{F})_{x,y}$. Second, for the smoothness of the estimated highlight-free image \mathbf{F} , we minimize $\|\nabla \mathbf{F}_{x,y}\|^2$ where ∇ is the gradient. This smoothness term removes the highlight by enforcing the intensity of a faulty pixel to have a value regressed from its neighboring pixels. Third, for the spatial similarity of the estimated fault map, we minimize $\|\nabla \mathbf{M}(\mathbf{F})_{x,y}\|_1$, where $\|\cdot\|_1$ is the l_1 norm. This term clusters rather than disperses the locations of faulty pixels.

Based on the equality constraint in Eq. (7), the proposed optimization problem can be formulated as

$$E(\mathbf{F}) = \sum_{x,y=1}^{w,h} \|\mathbf{L}(\mathbf{F})_{x,y}\|^2 + \lambda_1 \|\nabla \mathbf{F}_{x,y}\|^2 + \lambda_2 \|\nabla \mathbf{M}(\mathbf{F})_{x,y}\|_1 \quad (8)$$

$$s.t. \quad (1 - \mathbf{M}(\mathbf{F})_{x,y}) (\mathbf{F}_{x,y} - \mathbf{D}_{x,y}) = 0,$$

where λ_1 and λ_2 are the predefined regularization factors. We simplify the constrained problem into an unconstrained problem by including the equality constraint in the cost function as a barrier

term. Furthermore, by replacing $\|\mathbf{L}(\mathbf{F})_{x,y}\|^2$ with Eq. (4), the final objective function becomes

$$E(\mathbf{F}) = \sum_{x,y=1}^{w,h} \mathbf{M}(\mathbf{F})_{x,y} \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2 + \lambda_1 \|\nabla \mathbf{F}_{x,y}\|^2 + \lambda_2 \|\nabla \mathbf{M}(\mathbf{F})_{x,y}\|_1 + \lambda_3 (1 - \mathbf{M}(\mathbf{F})_{x,y}) \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2, \quad (9)$$

where λ_3 is a weight factor for the barrier term.

3.3. Optimization

Because the final objective function includes the multiplication of $\mathbf{M}(\mathbf{F})$ and \mathbf{F} , the problem becomes non-convex, and it is impossible to derive the solution directly. We solved this problem by applying an alternating optimization process that optimizes only one of $\mathbf{M}(\mathbf{F})$ and \mathbf{F} by fixing the other. However, because $\mathbf{M}(\mathbf{F})$ is dependent on \mathbf{F} as in Eq. (4), the direct alternating optimization converges to a premature state. We prevent this by introducing an auxiliary variable \mathbf{Z} that converges to \mathbf{F} asymptotically. We add an auxiliary penalty term that represents the equality of the original variable \mathbf{F} and the auxiliary variable \mathbf{Z} . The cost function with the auxiliary variable \mathbf{Z} can be reformulated as

$$E(\mathbf{F}, \mathbf{Z}) = \sum_{x,y=1}^{w,h} \mathbf{M}(\mathbf{Z})_{x,y} \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2 + \lambda_1 \|\nabla \mathbf{F}_{x,y}\|^2 + \lambda_2 \|\nabla \mathbf{M}(\mathbf{Z})_{x,y}\|_1 + \lambda_3 (1 - \mathbf{M}(\mathbf{Z})_{x,y}) \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2 + \frac{1}{2\theta} \|\mathbf{F}_{x,y} - \mathbf{Z}_{x,y}\|^2, \quad (10)$$

where θ is the inverse of weight for the auxiliary penalty term. As the optimization iterates, we reduce θ asymptotically to almost zero. Then, the weight of the additional penalty term becomes infinity, so \mathbf{Z} converges to \mathbf{F} and Eq. (10) becomes equivalent to Eq. (9). We control the convergence speed and optimization accuracy by tuning the initial value (θ_0) of θ and the decreasing rate ($\gamma < 1$) of θ .

Prior to the alternating optimization process, \mathbf{M} is roughly initialized, and the initial \mathbf{F} is optimized using the initial \mathbf{M} . Then, \mathbf{Z} is optimized by fixing the initial \mathbf{F} , and the next \mathbf{F} is optimized after fixing \mathbf{Z} . The alternating process is repeated until a termination condition is satisfied.

3.3.1. Initialization

A rough fault map \mathbf{M}^o is initialized by integrating two rough binary maps: an initial saturated pixel map $\mathbf{M}^{o,1}$ and an initial flash highlight map $\mathbf{M}^{o,2}$. $\mathbf{M}^{o,1}$ is the pixels where the intensities are saturated in the given flash image, and $\mathbf{M}^{o,2}$ designates the pixels located at the flash highlight. $\mathbf{M}^{o,1}$ and $\mathbf{M}^{o,2}$ are binary maps that do not have to be accurate because the fault map is also optimized as an unknown variable. \mathbf{M}^o is integrated by an OR-operation between $\mathbf{M}^{o,1}$ and $\mathbf{M}^{o,2}$.

$\mathbf{M}^{o,1}$ is found by detecting the fully saturated pixels as

$$\mathbf{M}_{x,y}^{o,1} = \begin{cases} 1 & , \text{ if } \max_{k=r,g,b} \mathbf{I}_f^o(x, y, k) = 1 \\ 0 & , \text{ otherwise.} \end{cases} \quad (11)$$

$\mathbf{M}^{o,2}$ is roughly obtained via heuristics on the flash highlights, derived from an observation that the intensity values around a highlight form an umbrella shape while those in a highlight-free area form a relatively flat shape. As a metric to classify the shape of intensity values in a patch,

we introduce regularized brightness denoted by $B_{(x,y)}$ for the patch centered at (x, y) . If $V_{(x,y)}$ is the brightness at (x, y) , $B_{(x,y)}$ is defined by

$$B_{(x,y)} = \frac{1}{(2s+1)^2} \sum_{i,j=-s}^{i,j=s} \frac{V_{(x+i,y+j)}}{V_{(x,y)}}, \quad (12)$$

where s is the size of the patch and $V_{(x,y)}$ is obtained by averaging the RGB values at (x, y) . A highlight patch has a maximum intensity around its center, so most normalized intensity values are less than one due to the umbrella shape. Furthermore, a highlight-free patch has similar values over the entire patch, so most normalized intensity values are almost one. Hence, $B_{(x,y)}$ of a highlight-free patch is greater than that of a highlight patch. The flash highlights only appear in the flash image, while the no-flash image has no highlight at the same position. Therefore, if $B_{(x,y)}$ in the no-flash image is greater than in the flash image, the pixel (x, y) in the flash image is determined as the highlighted pixel. We prevent an erroneous decision in an umbrella-shaped area other than highlights by adding a condition that the brightness at (x, y) should be greater than a predefined threshold. Thus, the value of $\mathbf{M}_{x,y}^{o,2}$ is set to 1 when the following two conditions are satisfied:

$$B_{(x,y)}^{nf} - B_{(x,y)}^f > \epsilon_1, \quad \mathbf{V}_{(x,y)}^f > \epsilon_2, \quad 0 \leq \epsilon_1 \leq 1, 0 \leq \epsilon_2 \leq 1 \quad (13)$$

where $(\cdot)^{nf}$ and $(\cdot)^f$ are the metrics from the no-flash image and the flash image, respectively, and ϵ_1 , and ϵ_2 are predefined parameters.

With the initial fault map $\mathbf{M}_{x,y}^o$, the initial \mathbf{F} can be obtained by minimizing the simplified objective function

$$E(\mathbf{F}^o) = \sum_{x,y=1}^{w,h} \mathbf{M}_{x,y}^o \|\mathbf{F}_{x,y}^o - \mathbf{D}_{x,y}\|^2 + \lambda_1 \|\nabla \mathbf{F}_{x,y}^o\|^2 + \lambda_3 \{1 - \mathbf{M}_{x,y}^o\} \|\mathbf{F}_{x,y}^o - \mathbf{D}_{x,y}\|^2, \quad (14)$$

where \mathbf{F}^o is the initial \mathbf{F} . Because Eq. (14) is a quadratic form in terms of $\mathbf{F}_{x,y}^o$, we can easily find $\mathbf{F}_{x,y}^o$ by solving the least squares problem.

3.3.2. \mathbf{Z} optimization process

When we optimize \mathbf{Z} , \mathbf{F} is fixed. The first iteration uses the initial \mathbf{F}^o . The fixed \mathbf{F} is denoted by \mathbf{F}^* ; then, the objective function is given by

$$E_{\mathbf{F}=\mathbf{F}^*}(\mathbf{Z}) = \sum_{x,y=1}^{w,h} \mathbf{M}(\mathbf{Z})_{x,y} \|\mathbf{F}_{x,y}^* - \mathbf{D}_{x,y}\|^2 + \lambda_2 \|\nabla \mathbf{M}(\mathbf{Z})_{x,y}\|_1 + \lambda_3 \{1 - \mathbf{M}(\mathbf{Z})_{x,y}\} \|\mathbf{F}_{x,y}^* - \mathbf{D}_{x,y}\|^2 + \frac{1}{2\theta} \|\mathbf{F}_{x,y}^* - \mathbf{Z}_{x,y}\|^2. \quad (15)$$

Because $\mathbf{M}(\mathbf{Z})$ is a discrete function, the optimal \mathbf{Z} is hard to find with conventional gradient-based methods. We solve the problem by applying a hypothesis testing approach. First, we suppose two solution hypotheses: $\mathbf{Z}_{x,y} = \mathbf{D}_{x,y}$ and $\mathbf{Z}_{x,y} \neq \mathbf{D}_{x,y}$. For every pixel, we evaluate the cost values with respect to the two hypotheses. Then, the solution can be determined by the hypothesis with a lower cost value.

If we set the hypothesis $\mathbf{Z}_{x,y} = \mathbf{D}_{x,y}$, $\mathbf{M}(\mathbf{Z})_{x,y}$ equals 0 from Eq. (4). For this case, the cost value of each pixel can be evaluated by

$$E_{x,y}^{(\mathbf{Z}=\mathbf{D})} = \lambda_2 \|\nabla \mathbf{M}(\mathbf{D})_{x,y}\|_1 + \left(\lambda_3 + \frac{1}{2\theta} \right) \|\mathbf{F}_{x,y}^* - \mathbf{D}_{x,y}\|^2, \quad (16)$$

where the element of $\nabla \mathbf{M}(\mathbf{D})_{x,y}$ is $-1, 0$, or 1 .

However, if we set a hypothesis that $\mathbf{Z}_{x,y} = \mathbf{F}_{x,y}^* \neq \mathbf{D}_{x,y}$, $\mathbf{M}(\mathbf{Z})_{x,y}$ becomes 1 from Eq. (4). Then the cost value equals

$$E_{x,y}^{(\mathbf{Z} \neq \mathbf{D})} = \lambda_2 \|\nabla \mathbf{M}(\mathbf{F}^*)_{x,y}\|_1 + \|\mathbf{F}_{x,y}^* - \mathbf{D}_{x,y}\|^2, \quad (17)$$

where the element $\nabla \mathbf{M}(\mathbf{F}^*)_{x,y}$ is $-1, 0$, or 1 . By comparing the cost values $E_{x,y}^{(\mathbf{Z}=\mathbf{D})}$ and $E_{x,y}^{(\mathbf{Z} \neq \mathbf{D})}$ in Eqs. (16) and (17), the optimal $\mathbf{Z}_{x,y}$ and $\mathbf{M}(\mathbf{Z})_{x,y}$ can be determined by

$$\{\mathbf{Z}_{x,y}, \mathbf{M}(\mathbf{Z})_{x,y}\} = \begin{cases} \{\mathbf{D}_{x,y}, 0\} & , \text{ if } E_{x,y}^{(\mathbf{Z}=\mathbf{D})} \leq E_{x,y}^{(\mathbf{Z} \neq \mathbf{D})} \\ \{\mathbf{F}_{x,y}^*, 1\} & , \text{ otherwise.} \end{cases} \quad (18)$$

Because the pixel-wise gradient of $\mathbf{M}(\mathbf{Z})$ should be newly estimated when $\mathbf{M}(\mathbf{Z})$ changes, the \mathbf{Z} optimization is iterated until $\mathbf{M}(\mathbf{Z})$ becomes fixed. We limit the number of iterations to 100 to prevent the infinite loop caused by unexpected oscillation of the optimized $\mathbf{M}(\mathbf{Z})$. Because the \mathbf{Z} optimization process requires pixel-wise binary comparisons, its computational complexity is $O(N)$, where N is the number of pixels.

3.3.3. F optimization process

The other variable \mathbf{Z} is fixed by \mathbf{Z}^* obtained in the previous iteration to find the optimal \mathbf{F} . Thus, the cost function for \mathbf{F} is given by

$$E_{\mathbf{Z}=\mathbf{Z}^*}(\mathbf{F}) = \sum_{x,y=1}^{w,h} \mathbf{M}(\mathbf{Z}^*)_{x,y} \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2 + \lambda_1 \|\nabla \mathbf{F}_{x,y}\|^2 \\ + \lambda_3 \{1 - \mathbf{M}(\mathbf{Z}^*)_{x,y}\} \|\mathbf{F}_{x,y} - \mathbf{D}_{x,y}\|^2 + \frac{1}{2\theta} \|\mathbf{F}_{x,y} - \mathbf{Z}_{x,y}^*\|^2. \quad (19)$$

Because Eq. (19) is a quadratic form of \mathbf{F} , the optimal \mathbf{F} can be calculated using the least squares solution with a pseudo-inverse. When the number of pixels is N , the computational complexity of the \mathbf{F} optimization process is $O(N^3)$, caused by the pseudo-inverse operation.

3.3.4. Termination condition

The two optimization processes are iterated until \mathbf{F} and \mathbf{Z} become almost equivalent. Thus, the entire optimization terminates when the following condition is satisfied.

$$\sum_{x,y=1}^{w,h} \|\mathbf{F}_{x,y} - \mathbf{Z}_{x,y}\|^2 < \epsilon, \quad (20)$$

where the left term is the sum of the final penalty terms in Eq. (10) and ϵ is a predefined threshold for the termination.

4. Experimental result

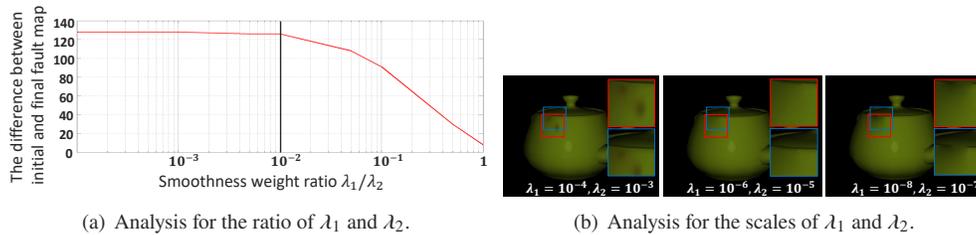
The proposed method was evaluated using two types of datasets: synthetic and real-world scenes. The three synthetic datasets include various shapes, textures, and colored lights. The synthetic dataset with various shapes comprises 10 different synthetic 3D models (*arc, ball, box, capsule, cone, spin, teapot, torus, torus2, and tube*) with various types and positions of virtual lights. The synthetic dataset with various textures has 10 synthetic 3D models (*teapot*) with different textures on their specular surfaces. The synthetic dataset with various colored lights and comprises the same 10 synthetic 3D models (*teapot*) illuminated by different colored lights. A virtual light near a virtual camera was used as a flash to obtain all the synthetic datasets. The ground truth of the highlight-free image was obtained by removing the specularly from the surface of the 3D

models. The real-scene dataset comprises four image pairs (“*objects*”, “*television*”, “*reflection*”, and “*desk*”) taken by a DSLR camera (Canon EOS 400D) with fixed camera settings for each pair. An embedded flash was used for the flash image to demonstrate the proposed method’s practical applicability.

The proposed method was compared with state-of-the-art methods based on the dichromatic model (Tan and Ikeuchi [1] and Yang *et al.* [2]), the annoying effect removal method using the flash/no-flash image pair (Agrawal *et al.* [9]), and the reflection removal using a pure flash image (Lei and Chen [31]). All datasets and codes of the proposed method are uploaded online [32]. The algorithm was executed using an unoptimized MATLAB code without any GPU resources. For the computational environment of an Intel i5-4670 CPU and 8GB memory, the algorithm took 312 seconds to estimate the results of the “*teapot*” synthetic image where the size was 640×480 pixels. Lei and Chen [31] requires the GPU to run the deep learning model, so we additionally utilized NVIDIA GTX 1080 Ti. Although Lei and Chen [31] can handle the synthetic datasets of the relatively small resolution, it fails to process the real-world datasets comprising the high resolution due to the out-of-memory issue.

4.1. Parameters

The parameter analysis experiments were performed using *teapot* of the synthetic dataset with various shapes to determine the values of λ_1 and λ_2 in the cost function Eq. (9). First, we measured the number of different pixels between the initial and final fault maps by varying the ratio λ_1/λ_2 , as depicted in Fig. 3(a). When λ_2 is between 10^{-7} and 10^{-3} , the numbers of different pixels are similar even though the values of λ_1 and λ_2 are different with the preserved ratio of λ_1/λ_2 . If λ_1/λ_2 is less than 0.1, the proposed optimization changes a sufficient number of pixels in the fault map. In the plausible range of ratios ($\lambda_1/\lambda_2 \leq 0.1$), the largest ratio ($\lambda_1/\lambda_2 = 0.1$) results in the best balancing between the two smoothness terms. Thus, we set λ_1/λ_2 to 0.1.



(a) Analysis for the ratio of λ_1 and λ_2 .

(b) Analysis for the scales of λ_1 and λ_2 .

Fig. 3. Analysis of weights. The weights of the smoothness terms were analyzed using *teapot* by changing their ratios and scales.

Second, even though the number of changed pixels in the fault map was similar when $\lambda_1/\lambda_2 = 0.1$, the performance varies depending on the values of λ_1 and λ_2 . Thus, we determined the appropriate values of λ_1 and λ_2 by performing optimization with various values of λ_1 and λ_2 fixing the ratio λ_1/λ_2 at 0.1. When $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-3}$, the color distortion remains even after the proposed optimization, as illustrated in the first image (red box) in Fig. 3(b). Furthermore, when $\lambda_1 = 10^{-8}$ and $\lambda_2 = 10^{-7}$, the edges of the faulty pixels are blurred by the smoothness terms, as depicted by the last image (blue box) in Fig. 3(b). Therefore, we recovered the color of the faulty pixels while preserving the edges by selecting 10^{-6} and 10^{-5} as the values of λ_1 and λ_2 , respectively. λ_3 should be much larger than the other weights because it is the weight of the barrier term that should be enforced as zero. Thus, we set λ_3 to 100.

The termination threshold ϵ was set to 0.01. The size of the normalized patch (s) in Eq. (12) was 30, and the thresholds for the initial fault maps ϵ_1 and ϵ_2 were set to 0.02 and 0.5, respectively. In our experiments, all the above parameters were fixed. Although the parameters were chosen

through experiments using one of the synthetic scenes, the parameters produced consistent, satisfactory performance for all synthetic and real-world scenes.

4.2. Qualitative evaluation

We evaluated the performance qualitatively by comparing the proposed algorithm with three state-of-the-art algorithms: Yang *et al.* [2], Agrawal *et al.* [9], and Lei and Chen [31]. Representative cases are illustrated using the final resultant map \mathbf{F} together with the ground-truth images where the specularity of the target object was removed for the ambient surface. For comparison, we also provided the resultant maps by Yang *et al.* [2], Agrawal *et al.* [9], and Lei and Chen [31].

4.2.1. Synthetic dataset

Synthetic dataset with various shapes. Fig. 4(a and b) illustrates a qualitative comparison between the results from the synthetic dataset with various shapes and the ground truth. In *teapot* (Fig. 4(a)), the colors on the highlights were distorted by existing methods while being recovered by the proposed method. Even with the complex shape of *torus2* (Fig. 4(b)), the original colors of the highlights were well-estimated, and the result was similar to the ground truth after using the proposed optimization.

Synthetic dataset with various textures. We validated the proposed method with complex surfaces by experimenting with 10 synthetic *teapot* images covered with different textures. As depicted in Fig. 4(c and d), the proposed method outperforms existing methods on textured surfaces. The proposed algorithm preserves the intensities and textures on the correct pixels. Furthermore, by locating the first regularizing term in Eq. (8), the proposed algorithm can reduce the blurriness of the textures on the faulty pixels. Although Lei and Chen [31] show the impressive results for the textured objects, the reflection and highlights upon the objects' surfaces remained while they were removed using our proposed algorithm. We can discover the difference because Lei and Chen [31] assumes the surfaces is non-specular.

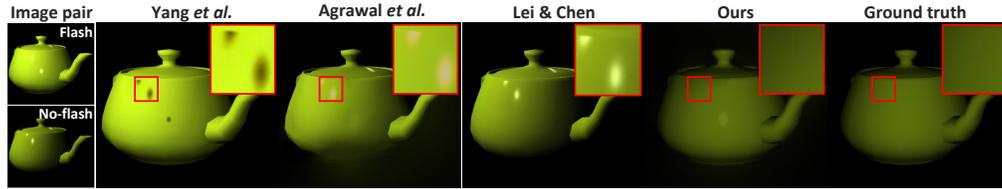
Synthetic dataset with various colored lights. We demonstrated the method's robustness on the colors of illuminating lights by performing experiments on 10 synthetic *teapot* images illuminated with lights of different colors. As depicted in Fig. 4(e and f), the proposed algorithm can remove highlights caused by natural lights even with colored lights. Because all the effects of natural lights are removed from the difference image, the proposed algorithm can ignore the various colors of the illuminating lights.

4.2.2. Real scene dataset

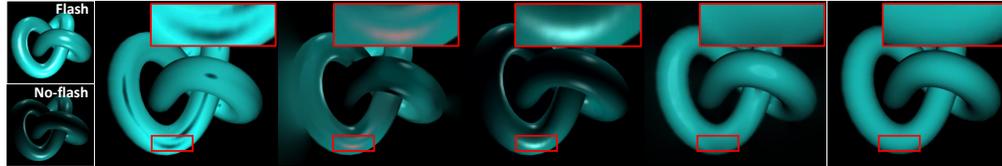
The flash is much dimmer than the other natural lights. Because the proposed method determines the highlight-free image based on the difference image, the brightness of the resultant image can be much lower than that of the flash image. We solved this problem through post-processing by increasing the brightness values (L in the Lab space) of every pixel in the resultant image to set the average brightness of the resulting image equivalent to that of the flash image. We validated the applicability of the proposed method by testing it with four real-scene pairs, as depicted in Fig. 5: "*objects*", "*television*", "*reflection*", and "*desk*".

The *objects* pairs include various objects with complex light sources. Several highlights appear on the objects, especially the surface of the fan and the ceramic lid of the cup. In the resultant images from the methods based on the dichromatic model [2], the white color on the surface of the fan is distorted green. Furthermore, the method using a flash/no-flash pair [9] blurs the boundary of objects and cannot remove the highlights from the surface of the fan.

In contrast, the proposed algorithm removes all highlights successfully with the preserved details on the non-highlighted regions. Moreover, *television* captures a television screen and a whiteboard, which are both highly specular. Fluorescent lights are reflected directly onto the specular surface of the television, which creates various highlights. This image pair is challenging,



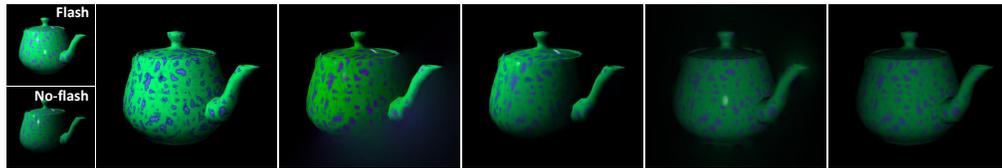
(a) *Teapot* in the synthetic dataset with various shapes



(b) *Torus2* in the synthetic dataset with various shapes



(c) *Woody-texture* in the synthetic dataset with various textures



(d) *Dent-texture* in the synthetic dataset with various textures



(e) *Red-light* in the synthetic dataset with colored lights



(f) *Emerald-light* in the synthetic dataset with colored lights

Fig. 4. Qualitative results for synthetic datasets. The images in the fifth column are the results of the proposed method, and those in the final column are the ground-truth images of the highlight-free images.

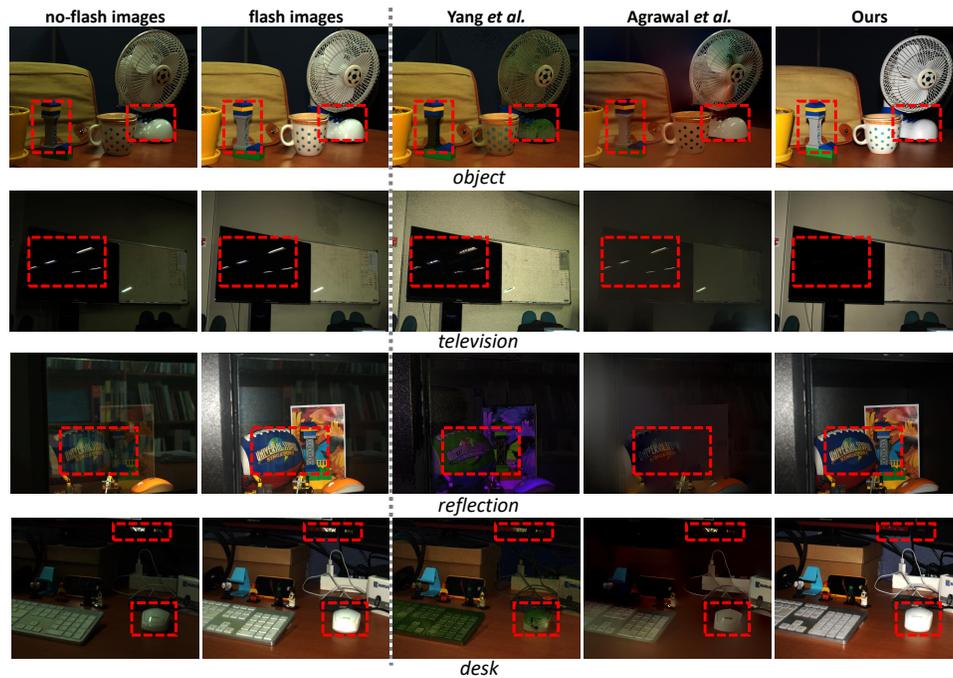


Fig. 5. Qualitative results for real-world scenes. The images in first and second columns are the no-flash/flash image pairs for each dataset. The images in the remaining columns are the results of Yang *et al.* [2], Agrawal *et al.* [9], and our method in order.

but the proposed method successfully removes the highlights caused by the fluorescent lights from the television screen. In contrast, the compared algorithms fail to remove the highlights.

The *reflection* pairs consist of several objects behind transparent glass reflecting an annoying background. Although the objects have little specularicity, existing algorithms create color distortion when removing the specular component from the input image. In contrast to existing algorithms, the proposed method preserves the objects' original colors. The proposed optimization removes the background reflected on the glass because it obtains a highlight-free image from the difference image that ignores the effects of natural lights causing the background reflection on the glass.

The *desk* pairs include various objects such as a monitor and keyboard on a desk. With existing algorithms, the colors of the keyboard and mouse are biased, there are highlights on the mouse, and the bottom-right boundary of the monitor is still not removed. In contrast, the proposed method successfully estimates the original color of the highlighted pixels while maintaining the textures on the mouse.

The comparison using real-world scenes demonstrates that the proposed method can be easily used in practice. Furthermore, the previous state-of-the-art algorithms suffer from practical limitations, such as remaining highlights caused by the complex natural lights and the color distortion on saturated or low-chromatic pixels. Especially, Lei and Chen [31] fail to handle the high-resolution real-world scenes due to the lack of VRAM. This failure verifies the real-world applicability of the proposed algorithm where we need only the CPU-based optimization.

4.3. Quantitative evaluation

We evaluated the quantitative performance of the proposed method using hue distance and lab distance. The hue distance is the Euclidean distance between a result and a ground truth based on a and b of the lab color space. The highlight-free image obtained by our method includes

the lightness only from the light flash, but existing methods include the lightness from all lights. Therefore, we compared the highlight removal performance without using the lightness channel for a fair comparison. We used the lab distance to compensate for the metric without the lightness channel. The lab distance represents the perceptual similarity between the results and the ground truth.

4.3.1. Synthetic dataset

As represented by the hue distances in Table 1, the proposed method exhibited much less color distortion than the other methods for all the synthetic datasets with various shapes. From the lab distances, we validated that the results from the proposed algorithm were most similar to the ground truth. Although the proposed algorithm shows the impressive performance for a part of images containing textures surfaces (Table 2), Lei and Chen [31] shows the state-of-the-art performance on average. However, as shown in the qualitative results, despite its high quantitative performance, the highlights and reflections on the objects remain through Lei and Chen [31]. This issue happens because the complex textures can be blurred by the proposed algorithm which results in the dramatic increment of the error even while highlights and reflections were successfully removed. The limitations would be analyzed in Section 4.4. The proposed method outperformed the other methods for the images illuminating lights of various colors (Table 3). Interestingly, the distortion error of the proposed method was independent of the color of the illuminating lights because its optimization removes the natural light by subtracting the no-flash image from the flash image.

Table 1. Quantitative comparison with various shapes

		<i>arc</i>	<i>ball</i>	<i>box</i>	<i>capsule</i>	<i>cone</i>	<i>spin</i>	<i>teapot</i>	<i>torus</i>	<i>torus2</i>	<i>tube</i>	Avg.
Hue Distance	Tan [1]	8.70	98.82	299.45	262.32	487.00	230.40	231.38	256.76	14.07	619.85	250.87
	Yang [2]	5.53	28.98	38.04	107.40	3.80	91.44	84.63	75.72	7.49	14.87	45.79
	Agrawal [9]	26.88	25.70	30.80	92.54	2.89	81.11	56.15	50.29	36.75	11.33	41.44
	Lei&Chen [31]	28.02	39.71	21.64	62.08	59.78	24.63	19.60	19.06	43.79	24.04	34.24
	Ours	0.41	2.31	15.78	0.85	0.10	4.01	1.19	1.28	1.80	6.04	3.38
Lab Distance	Tan [1]	26.52	179.50	237.40	177.29	457.95	320.45	426.42	374.43	83.09	173.59	245.66
	Yang [2]	85.23	812.94	978.89	551.60	164.97	676.60	802.32	722.38	140.48	347.56	528.30
	Agrawal [9]	125.73	909.51	1098.60	466.30	145.99	627.95	399.42	472.19	256.94	423.40	492.60
	Lei&Chen [31]	84.07	134.58	117.40	113.24	169.17	144.84	83.61	75.69	205.57	94.60	122.28
	Ours	3.10	7.76	31.44	0.52	0.22	14.85	0.93	1.10	7.20	10.12	7.72

Table 2. Quantitative comparison with various textures

	Texture	Woody	Dent	Dent2	Marble	Checker	Cellular	Perlin	Swirl	Tiles	Smoky	Avg.
Hue Distance	Tan [1]	333.18	37.10	198.21	340.85	251.11	86.79	251.04	319.84	172.32	280.95	227.14
	Yang [2]	63.55	63.09	173.70	6.64	146.63	6.21	3.96	33.22	262.50	71.97	83.15
	Agrawal [9]	92.68	317.72	191.51	22.25	107.07	37.99	8.71	32.76	203.30	131.52	114.55
	Lei&Chen [31]	16.37	31.22	35.78	1.46	35.17	11.65	6.73	23.71	30.83	27.55	22.05
	Ours	1.58	6.87	130.78	96.04	451.20	15.80	0.61	20.44	44.09	2.02	76.94
Lab Distance	Tan [1]	282.26	632.39	783.45	413.82	733.72	655.12	191.62	219.06	585.84	287.40	478.47
	Yang [2]	555.22	581.67	814.66	361.20	738.15	539.45	261.01	525.89	687.87	395.09	546.02
	Agrawal [9]	449.60	437.65	309.11	283.36	351.06	317.57	230.09	320.84	227.79	206.90	313.40
	Lei&Chen [31]	71.30	77.63	92.56	5.24	92.56	97.68	68.41	80.69	58.72	52.54	68.78
	Ours	13.91	6.11	26.66	177.02	1564.62	79.00	3.05	97.87	112.07	1.71	208.20

Table 3. Quantitative comparison with various light colors

	Color of Light	Red	Green	Blue	Yellow	Purple	Emerald	Mixed1	Mixed2	Mixed3	Mixed4	Avg.
Hue Distance	Tan [1]	625.92	1215.53	64.10	131.79	191.65	382.77	303.36	39.62	154.74	426.32	353.58
	Yang [2]	586.48	1133.46	18.83	249.51	158.98	328.99	254.33	80.17	291.22	802.32	390.43
	Agrawal [9]	629.35	622.46	341.72	120.54	347.39	175.68	196.49	97.75	122.09	399.42	305.29
	Lei&Chen [31]	66.77	47.08	26.08	14.96	43.17	26.90	27.25	15.98	13.56	15.78	29.75
	Ours	2.01	1.91	1.95	2.10	1.98	1.80	2.01	1.96	1.76	1.58	1.91
Lab Distance	Tan [1]	55.87	86.49	6.84	9.28	9.95	20.00	16.91	1.81	9.78	34.61	25.15
	Yang [2]	34.85	50.13	0.96	21.52	8.69	12.99	10.21	5.10	25.02	82.69	25.22
	Agrawal [9]	22.77	27.94	18.99	8.27	13.08	11.48	11.54	7.30	9.11	37.97	16.85
	Lei&Chen [31]	93.35	80.98	79.74	54.20	84.29	59.62	68.32	60.92	51.67	64.16	69.72
	Ours	0.38	0.26	0.46	0.39	0.46	0.38	0.44	0.43	0.24	0.37	0.38

4.3.2. Real-scene dataset

We validated the proposed method quantitatively in the real-scene dataset by creating ground-truth images for the real-world scenes by removing all the highlights in the flash images. The removed regions were filled in with the colors of the nearby pixels. Although the contours of the removed highlight regions remained, the hue distance we used was barely affected by the contours because the brightness values were ignored in the measure.

Table 4 presents the quantitative comparison in which the proposed method outperformed existing algorithms on average. Because the a and b values of the lab space became diverse for colors with low chroma, the hue distance on the white or black colors could be enlarged regardless of the qualitative results. Thus, for the *object* scene, the hue distance of the proposed method was worse than that of Yang *et al.* [2], regardless of the quality of the resulting images. In contrast to the hue distance, the lab distance accurately represented the quantitative quality of the result from the proposed algorithm.

Table 4. Quantitative comparison with real-world scenes

	Scene	<i>object</i>	<i>television</i>	<i>reflection</i>	<i>desk</i>	Avg.
Hue Dist.	Yang [2]	80.32	60.73	1247.23	114.63	375.83
	Agrawal [9]	347.18	34.84	230.42	343.77	239.05
	Ours	180.75	32.58	25.61	97.52	84.11
	Yang [2]	371.77	505.71	1738.82	444.15	765.11
Lab Dist.	Agrawal [9]	712.33	419.12	819.72	502.93	613.52
	Ours	614.04	110.64	308.26	316.72	337.42

4.4. Limitations

Although the proposed algorithm can successfully remove highlights and reflections, it still has limitations. First, when the colors of the neighboring correct pixels are not similar to the original colors of the faulty pixels, our method exhibits reduced performance because the smoothness cost term $\|\nabla \mathbf{F}_{x,y}\|^2$ assumes similarity among the neighboring pixels. Consequently, our method had a relatively lower performance for the images with textures than for those with various shapes or colored lights, as presented in Tables 1, 2, and 3. Furthermore, when the region of the faulty pixels extends beyond the object boundary, the neighboring pixels are located near the other objects, which makes the boundary blurry, as illustrated in the top-left region in Fig. 2. Second, the flash and no-flash image pairs must be precisely aligned to be used for the proposed algorithm.

For this precise alignment, the camera should be fixed by a tripod; thus, our method cannot be applied to a scene containing moving objects. Finally, the difference image used in our method has inherent limitations. Because the difference image is based on the effect of the light flash, the resulting image can vary according to the location and the intensity of the light flash. Shadows may appear by the light flash, and the objects far from the flash can be represented with darker intensities in the difference image than in a natural light image.

5. Conclusion

In natural scenes, highlights on objects' surfaces are inevitable, degrading the quality of photographs and the performance of computer vision algorithms. We removed the highlights using a proposed optimization-based method based on a flash/no-flash image pair. The contributions of this study are summarized as follows. First, the efficiency increases when using only a pair of flash/no-flash images taken without any special device nor any limited capturing setting, otherwise necessary for existing methods. Second, our method does not require the dichromatic assumption. Instead, we formulated an optimization problem to recover the original color even from saturated or low-chromatic pixels, outperforming existing methods based on the dichromatic model. Third, our optimization formulation is novel because the locations of faulty pixels and the highlight-free image map are estimated by the alternating optimization process with our cost function, outperforming existing methods. However, our approach also has limitations, such as the blurriness of texture details and the need for camera alignment. Future studies that tackle the limitations of our method can consider adaptive weights based on the color similarity of the neighboring pixels to preserve textured details. Furthermore, the optical flow algorithm can be used for pre-processing to align the mismatched pixels caused by incorrect camera alignment. Finally, with the use of additional controllable light sources, our method can be extended to obtain an image with user-designed colors and lightness after removing the unwanted colors and lightness from uncontrollable natural light sources.

Funding. Korea Creative Content Agency (R2021040044); Chung-Ang University.

Acknowledgments. This work was supported by Culture Technology R&D Program 2021 through the Korea Creative Content Agency funded by Ministry of Culture, Sports and Tourism (A Specialist Training of Content R&D based on Virtual Production, R2021040044) and the Chung-Ang University Research Grants in 2020.

Disclosures. Sangdoo Yun: NAVER Corporation (E)

Data availability. Data underlying the results presented in this paper are available in dataset 1, Ref. [32].

References

1. R. Tan and K. Ikeuchi, "Separating reflection components of textured surfaces using a single image," *IEEE Trans. Pattern Anal. and Mach. Intell.* **27**(2), 178–193 (2005).
2. Q. Yang, J. Tang, and N. Ahuja, "Efficient and robust specular highlight removal," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(6), 1304–1311 (2015).
3. S. Nayar, X. Fang, and T. Boult, "Separation of reflection components using color and polarization," *Int. J. Comput. Vis.* **21**(3), 163–186 (1997).
4. Y. Sato and K. Ikeuchi, "Temporal-color space analysis of reflection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (1993).
5. J. DiCarlo, F. Xiao, and B. Wandell, "Illuminating illumination," in *Color and Imaging Conference*, (2001).
6. G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.* **23**(3), 664–672 (2004).
7. E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Trans. Graph.* **23**(3), 673–678 (2004).
8. M. Drew, C. Lu, and G. Finlayson, "Removing shadows using flash/noflash image," in *IEEE International Conference on Multimedia and Expo*, (2006).
9. A. Agrawal, R. Raskar, S. Nayar, and Y. Li, "Removing photography artifacts using gradient projection and flash-exposure sampling," *ACM Trans. Graph.* **24**(3), 828–835 (2005).
10. D. Crispell, D. Lanman, P. Sibley, Y. Zhao, and G. Taubin, "Beyond silhouettes surface reconstruction using multi-flash photography," in *International Symposium on 3D Data Processing, Visualization, and Transmission*, (2006).

11. R. Feris, R. Raskar, K. Tan, and M. Turk, "Specular reflection reduction with multi-flash imaging," in *Brazilian Symposium on Computer Graphics and Image Processing*, (2004).
12. J. Sun, J. Sun, S. Kang, Z. Xu, X. Tang, and H. Shum, "Flash cut: foreground extraction with flash and no-flash image pairs," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2007).
13. Q. C. H. Shen, "Simple and efficient method for specular removal in an image," *Appl. Opt.* **48**(14), 2711–2719 (2009).
14. P. Tan, L. Quan, and S. Lin, "Separation of highlight reflections on textured surfaces," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2006).
15. H. Kim, H. Jin, S. Hadap, and I. Kweon, "Specular reflection separation using dark channel prior," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2013).
16. J. Suo, D. An, X. Ji, H. Wang, and Q. Dai, "Fast and high quality highlight removal from a single image," *IEEE Trans. Image Processing* **25**(11), 5441–5454 (2016).
17. S. P. Mallick, T. Zickler, P. N. Belhumeur, and D. J. Kriegman, "Specularity removal in images and videos: A pde approach," in *European Conference on Computer Vision (ECCV)*, (2006).
18. G. Fu, Q. Zhang, L. Zhu, P. Li, and C. Xiao, "A multi-task network for joint specular highlight detection and removal," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021).
19. A. Souza, M. Macedo, V. Nascimento, and B. Oliveira, "Real-time high-quality specular highlight removal using efficient pixel clustering," in *Brazilian Symposium on Computer Graphics and Image Processing*, (2018).
20. S. Umeyama and G. Godin, "Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(5), 639–647 (2004).
21. S. Lin and H. Shum, "Separation of diffuse and specular reflection in color images," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2001).
22. S. Lee and R. Bajcsy, "Detection of specularity using color and multiple views," in *European Conference on Computer Vision (ECCV)*, (1992).
23. S. Lin, Y. Li, S. Kang, X. Tong, and H. Shum, "Diffuse-specular separation and depth recovery from image sequences," in *European Conference on Computer Vision (ECCV)*, (2002).
24. W. Feng, X. Li, X. Sun, J. Gao, T. Qu, S. Wang, and D. Zhao, "Specular highlight removal based on an iterative light field method," *Appl. Opt.* **60**(13), 4039–4046 (2021).
25. A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. Graph.* **26**(3), 69 (2007).
26. M. Havlena and K. Schindler, "Vocmatch: efficient multiview correspondence for structure from motion," in *European Conference on Computer Vision (ECCV)*, (2014).
27. F. Lu, Y. Matsushita, I. Sato, T. Okabe, and Y. Sato, "From intensity profile to surface normal: photometric stereo for unknown light sources and isotropic reflectances," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(10), 1999–2012 (2015).
28. X. Cao, M. Waechter, B. Shi, Y. Gao, B. Zheng, and Y. Matsushita, "Stereoscopic flash and no-flash photography for shape and albedo recovery," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020).
29. X. Cao, M. Waechter, B. Shi, Y. Gao, B. Zheng, F. Okura, and Y. Matsushita, "Shape and albedo recovery by your phone using stereoscopic flash and no-flash photography," *Int. J. Comput. Vis.* (2022).
30. J. Xiong, J. Wang, W. Heidrich, and S. Nayar, "Seeing in extra darkness using a deep-red flash," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021).
31. C. Lei and Q. Chen, "Robust reflection removal with reflection-free flash-only cues," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021).
32. J. Choi, "Highlight Removal Website," <https://sites.google.com/vilab.cau.ac.kr/highlightremoval/> (2022). [Online; accessed 13-May-2022].