White Balance under Mixed Illumination using Flash Photography

Zhuo Hui, Aswin C. Sankaranarayanan
Carnegie Mellon University

Kalyan Sunkavalli, Sunil Hadap
Adobe Research

Abstract

Real-world illumination is often a complex spatially-varying combination of multiple illuminants. In this work, we present a technique to white-balance images captured in such illumination by leveraging flash photography. Even though this problem is severely ill-posed, we show that using two images – captured with and without flash lighting – leads to a closed form solution for spatially-varying mixed illumination. Our solution is completely automatic and makes no assumptions about the number or nature of the illuminants. We also propose an extension of our scheme to handle practical challenges such as shadows, specularities, as well as the camera and scene motion. We evaluate our technique on datasets captured in both the laboratory and the real-world, and show that it significantly outperforms a number of previous white balance algorithms.

1. Introduction

White balancing aims to remove the effect of the color of the light sources illuminating a scene. Specifically, the goal is to render a photograph such that the observed color of scene points is influenced only by their reflectance and not by the spectral profile on the scene illumination. As is to be expected, white balancing is an integral part of most digital camera pipelines as well as image processing tools. While there are numerous techniques for performing white balancing, the vast majority of them assume that the captured scene is illuminated by a single light source [9–11,15].

Many real-world scenes are interesting precisely due to complex spatially-varying illumination with multiple light sources where the assumption of a single dominant light source is completely violated. In the context of white balancing, this is commonly referred to as the "mixed illumination" scenario [4, 16, 19, 20, 27]. In the absence of additional assumptions or constraints, white balancing under mixed illumination is highly ill-posed and intractable since it is entirely possible that each scene point is illuminated by its own unique configuration of the light sources. As a consequence, the vast majority of prior techniques that perform white balancing method for mixed illumination rely either on user guidance [3, 4, 21], or require knowledge of the number of light sources and their colors [19], or make simplifying assumptions that individual regions in the scene are illuminated by a single source [27].

In this paper, we propose an *automatic* technique for white balancing by making use of flash photography, i.e., capturing two photographs with and without the use of the camera flash. Our technique relies on two assumptions: first, the scene is predominantly Lambertian; and second, image intensities are preserved after white balancing (with a specific notion of preservation that we detail later). Under these assumptions, and given the flash/no-flash image pair, we show that a per-pixel white balancing kernel can be estimated in closed-form, without making any assumptions about the number or the nature of the light sources in the scene. As a consequence, the proposed method obviates the need for intensive user input as well as prior knowledge of the light sources. The interplay of these ideas leads to a robust, per-pixel, white balancing technique that provides state-of-the-art results on challenging real-life scenes. An example of the proposed technique is shown in Figure 1.

Contributions. We propose an automatic framework for white balancing, using flash photography, for arbitrarily complex lighting without requiring any user inputs or the restrictive assumptions on the scene illumination, all of which are endemic to most state-of-the-art techniques. We make the following contributions.

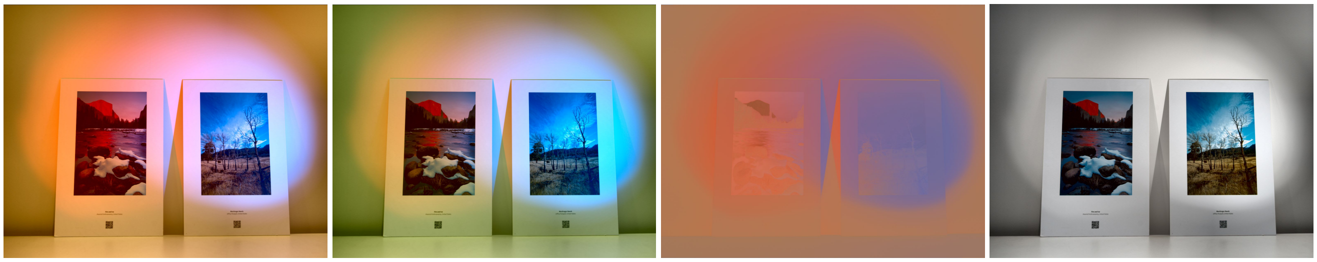
(i) *White balance kernel estimation.* We leverage the flash photograph to derive a closed-form solution for the per-pixel white balancing kernel.

(ii) *Model mismatch.* We propose techniques to handle shadows, specularities, and scene/camera motion.

(iii) *Validation.* We showcase the accuracy of the proposed white balancing technique on a wide range of scenes.

2. Prior work

Automatic approaches. Most state-of-the-art methods for automatic white balancing are based on the single light assumption [10, 12, 14, 15]. This naturally leads to techniques



(a) Input (no-flash) image (b) Gijsenij et al. [15] (c) Our white balance kernel (d) Our result

Figure 1: The input (no-flash) image (a) has complex mixed illumination that confounds white balance algorithms that assume a single illuminant [15]. This leads to a result (b), where the color casts have not been eliminated. Our technique leverages an additional flash photograph to estimate spatially-varying white balance kernels that capture the scene lighting (c), and produces a result (d), where the lighting color has been eliminated and the scene is rendered faithfully.

[2, 7, 16, 27] that deal with mixture of lights by assuming a single dominant light in each local region. However, the single/dominant light source model, even for local regions, is inherently limited since it is inapplicable to scenarios where multiple light sources mix gracefully. Hsu et al. [19] propose an efficient method for white balancing in the presence of two local lights. However, this method requires precise knowledge of the two light colors. Further, the two source assumption is often restrictive and hence, not desirable.

User-guided approaches. A flexible approach to white balancing is to obtain additional inputs from the users in the form of scribbles that mark regions with the same color or even regions that are known to be entirely white. User-guided approaches have been widely used in many image processing tasks including image editing [21], intrinsic images [3, 29], and color correction [4, 5]. In the context of white balancing, Boyadzhiev et al. [4] utilize user scribbles to indicate color attributes of the scene such as white surfaces and constant lighting regions. This enables an interpolation framework that propagates the information from user specified pixels to the under-determined regions. However, user guided approaches are not preferable for many reasons including the time and expertise required for a user to provide meaningful input.

Flash photography. Flash photography refers to techniques where two successive photographs are taken, one purely under scene illumination and another with additional flash illumination. Computational imaging using flash/no-flash image pairs has enabled many novel capabilities including denoising [8, 25], image segmentation [30], deblurring [31], artifact removal [1, 8, 28], saliency detection [18], and non-photorealistic rendering [26]. Flash photography has also been used for white balancing [24, 25]. Prior work in this area has relied critically on the work of DiCarlo et al. [6] where the color of spatially-varying illumination is estimated using flash/no-flash image pairs. Using this technique to estimate illumination color, Petschnigg et at. [25] propose an approach for white balancing that leverages the difference between the flash/no-flash pairs to construct a white balancing kernel. Lu and Drew [24] extend the color estimation technique in [6] using a simplified image formation model and use this result to white balance images. However, while all three methods above assume the illumination to be spatially-varying, they also assume that each scene patch is illuminated by a single dominant illuminant which is often an inaccurate model for scene lighting. In contrast, our proposed technique enables a tractable white balancing solution for arbitrarily complex lighting while allowing spatial mixing of these lights.

3. White balance under mixed illumination

We now describe our proposed per-pixel white balancing technique for mixed lighting using flash photography.

Setup and image formation. We assume that the scene is illuminated with N point light sources. Each source is associated with an intensity-scaled direction $\mathbf{s} \in \mathbb{R}^3$, and a unit-norm color vector, $\mathbf{l}_k \in \mathbb{R}^3$, where $\mathbf{l}$ is comprised of the per-channel intensities, $[l^r, l^g, l^b]^\top$. The N light sources are defined by the set of directions $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N\}$ and colors $\{\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_N\}$. Under this setup, and assuming that the scene is Lambertian, the intensity value observed in color channel c at pixel $\mathbf{p}$ in a no-flash image, I_{nf}, i.e., imaged in the absence of flash lighting, is given as:

$$I^c_{nf}(\mathbf{p}) = R^c(\mathbf{p}) \sum_i \max(\mathbf{n}(\mathbf{p})^\top \mathbf{s}_i, 0)\, l^c_i, \tag{1}$$

where $R(\mathbf{p})$ and $\mathbf{n}(\mathbf{p})$ denote the albedo and surface normal at pixel $\mathbf{p}$, respectively. Defining $\lambda_k(\mathbf{p}) = \max(\mathbf{n}(\mathbf{p})^\top \mathbf{s}_k, 0)$ as the local shading for k^{th} light source at pixel $\mathbf{p}$, we can write

$$I^c_{nf}(\mathbf{p}) = R^c(\mathbf{p}) \sum_i \lambda_i(\mathbf{p})\, l^c_i. \tag{2}$$

Defining the orientation of the flash as $\mathbf{s}_f$ and its color as $\mathbf{l}_f$, the intensity observed at color channel c at pixel $\mathbf{p}$ in the flash image I_f, i.e., under additional flash illumination, is given by:

$$I_f^c(\mathbf{p}) = R^c(\mathbf{p}) \left(\sum_i \lambda_i(\mathbf{p}) l_i^c + \lambda_f(\mathbf{p}) l_f^c \right), \tag{3}$$

where $\lambda_f(\mathbf{p}) = \max(\mathbf{n}(\mathbf{p})^\top \mathbf{s}_f, 0)$ denotes the shading produced by the flash. We assume knowledge of the flash color $\mathbf{l}_f$ via a calibration step.

Problem definition. Our goal is to produce a per-pixel white balancing kernel $\mathbf{W}(\mathbf{p}) \in \mathbb{R}^3$, that, when applied to the input no-flash image, produces a white balanced image $\widehat{I}_{wb}$ as:

$$\widehat{I}_{wb}^c(\mathbf{p}) = W^c(\mathbf{p}) I_{nf}^c(\mathbf{p}). \tag{4}$$

We want the kernel $\mathbf{W}(\mathbf{p})$ to eliminate the color of the incident illumination at each surface point so that the color of the white balanced image depends only the albedo. However, we would like to retain the local shading of each light source. Hence, the intensity of the white balanced image at pixel $\mathbf{p}$ can be written as:

$$\widehat{I}_{wb}^c(\mathbf{p}) = R^c(\mathbf{p}) \sum_i \lambda_i(\mathbf{p})\, \eta_i. \tag{5}$$

This is identical to (2), with the exception that the color of each individual light source has been replaced by the scalar, η_k; this corresponds to assigning the k^{th} light source with the neutral color $[\eta_k, \eta_k, \eta_k]$. By combining (2), (4) and (5), the white balance kernel is constructed as:

$$W^c(\mathbf{p}) \sum_i \lambda_i(\mathbf{p})\, l_i^c = \sum_i \lambda_i(\mathbf{p})\, \eta_i. \tag{6}$$

In the general case, we do not have any knowledge of the number of illuminants in the scene, the color of each illuminant $\mathbf{l}_i$, or the spatially-varying local shading induced by each light source $\lambda_i(\mathbf{p})$. This makes white balancing under mixed lighting severely under-constrained. We solve this ill-conditioned problem by leveraging an additional image captured under flash lighting. In addition, we rely on an image intensity preservation constraint that we will detail shortly. Together, these constraints help us derive a closed form solution for the per-pixel white balancing kernel.

Chromaticity of the white balanced image. We now invoke a result that relates the chromaticity of the white balanced image to that of the albedo [4, 19]. The chromaticity for the white balanced image, C_{wb}^c, and albedo, C_R^c, in the color channel c are defined as:

$$\widehat{C}_{wb}^c(\mathbf{p}) = \frac{\widehat{I}_{wb}^c(\mathbf{p})}{\sum_c \widehat{I}_{wb}^c(\mathbf{p}))} = \frac{R^c(\mathbf{p}) \sum_i \lambda_i(\mathbf{p}) \eta_i(\mathbf{p})}{\sum_c R^c(\mathbf{p}) \sum_i \lambda_i(\mathbf{p}) \eta_i(\mathbf{p})},$$

$$C_R^c(\mathbf{p}) = \frac{R^c(\mathbf{p})}{\sum_c R^c(\mathbf{p})}.$$

Since the local shading terms $\lambda_i(\mathbf{p})$ are invariant to color, we can now write:

$$\widehat{C}_{wb}^c(\mathbf{p}) = \frac{R^c(\mathbf{p})}{\sum_c R^c(\mathbf{p})} = C_R^c(\mathbf{p}). \tag{7}$$

Hence, the chromaticity of the white balanced image is equal to the chromaticity of the albedo.

Intensity preservation. We also leverage the intensity preservation assumption [4, 19], which enforces that the sum of the intensity at each pixel remain unchanged after white balancing.

$$\sum_c \widehat{I}_{wb}^c(\mathbf{p}) = \sum_c I_{nf}^c(\mathbf{p})$$

This allows us to express (4) in terms of chromaticites:

$$\widehat{C}_{wb}^c(\mathbf{p}) = W^c(\mathbf{p}) C_{nf}^c(\mathbf{p}).$$

Combining with (7), we can obtain

$$W^c(\mathbf{p}) C_{nf}^c(\mathbf{p}) = C_R^c(\mathbf{p}). \tag{8}$$

Note that the chromaticity $C_{nf}^c(\mathbf{p})$ can be estimated from the no-flash photograph. However, to estimate the white balancing kernel $W(\mathbf{p})$, we need to estimate the chromaticity of the albedo. We now show that the chromaticity of the albedo can be estimated using the flash/no-flash image pair.

Solution outline. Given the flash image and the color of the flash, we can determine the $C_R^c(\mathbf{p})$ from (2) and (3). We first define the difference image $\mathbf{d}$ which can be obtained by subtracting the no-flash image from the flash image.

$$\begin{aligned} d^c(\mathbf{p}) &= I_f^c(\mathbf{p}) - I_{nf}^c(\mathbf{p}) \\ &= R^c(\mathbf{p}) \lambda_f(\mathbf{p})\, l_f^c \end{aligned} \tag{9}$$

We denote

$$\alpha^c(\mathbf{p}) = R^c(\mathbf{p}) \lambda_f(\mathbf{p}) = \frac{d^c(\mathbf{p})}{l_f^c},$$

and subsequently, obtain an expression for the chromaticity of the albedo:

$$C_R^c(\mathbf{p}) = \frac{R^c(\mathbf{p})}{\sum_c R^c(\mathbf{p})} = \frac{\alpha^c(\mathbf{p})}{\sum_c \alpha^c(\mathbf{p})}.$$

The value of α can be computed from a flash/no-flash image pair and the color of the flash. Combining with (8), we can obtain the closed form solution for the white balance kernel:

$$\boxed{W^c(\mathbf{p}) = \frac{\alpha^c(\mathbf{p})}{C_{nf}^c(\mathbf{p}) \sum_c \alpha^c(\mathbf{p})}.} \tag{10}$$

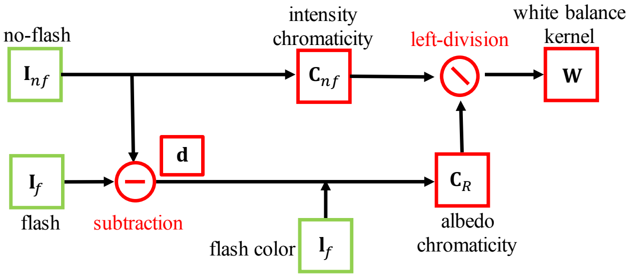


Figure 2: The proposed white balancing algorithm.

Note that the solution in (10) is completely automatic and requires little in terms of prior knowledge of the scene. In particular, it holds for arbitrarily complex spatially-varying lighting. We also make no assumptions about the geometry of the scene and can gracefully handle shading effects such as both cast and attached shadows in the no-flash photograph.

Figure 2 provides a schematic of the white balance algorithm. While our technique depends on knowing the color of the flash, this can be acquired via a one-time calibration using a color-checker chart. Our per-pixel framework is easy to implement simply by few element-wise operations (additions and divisions) and hence, can scale easily to very high resolution photographs and implementable in mobile and other resource-constrained platforms.

4. Extensions to real-world conditions

The white balancing technique described above assumes that the scene is Lambertian and that every scene point is illuminated by the flash. In addition, it relies on perfect alignment between the flash/no-flash image pair. Practical implementations of flash photography are bound to violate these assumptions in some parts of the scene. For example, the flash illumination can lead to specularities and shadows that are not present in the no-flash photograph. Further, we can expect scene motion between the two photographs as well as global motion due to handshake. These issues lead to wrong estimates of the white balance kernel at some scene points. In this section, we present simple heuristics to detect these errors and correct them.

4.1. Shadow and specularity detection

Our scheme aims to detect the specularities as well as the shadows caused by the flash. We rely on three metrics to achieve this:

Relative difference. We measure the difference between the flash and no-flash image at each pixel relative to the intensity of the no-flash image.

$$\mathcal{T}_d(\mathbf{p}) = \frac{\|\mathbf{I}_f(\mathbf{p}) - \mathbf{I}_{nf}(\mathbf{p})\|_2^2}{\|\mathbf{I}_{nf}(\mathbf{p}) + \epsilon\|_2^2},$$

where ϵ ensures that the denominator remain nonzero. For a registered image pair, the difference is simply the pure flash image. This difference is sensitive to the surface albedo which we attempt to remove by dividing by the magnitude of no-flash pixel values. Once normalized, we expect this metric to be very small in flash shadow regions (due to a small numerator) and large in flash specularities (due to a large numerator).

No-flash energy defined as $\mathcal{T}_{nf}(\mathbf{p}) = \|\mathbf{I}_{nf}(\mathbf{p})\|_2^2$.

Flash energy defined as $\mathcal{T}_f(\mathbf{p}) = \|\mathbf{I}_f(\mathbf{p})\|_2^2$.

We are only interested in detecting shadows and specularities in the flash photograph that do not occur in the no-flash photograph. To enable this, we first define Φ_{nf} as the intensity range of the pixels which does not involve shadow or specular in the no-flash image $\Phi_{nf} = [a, b]$, where $\Pr(\mathcal{T}_{nf}(\mathbf{p}) < a) = 0.05$ and $\Pr(\mathcal{T}_{nf}(\mathbf{p}) > b) = 0.05$.

For the shadow or specular free regions in the no-flash image ($\mathcal{T}_{nf} \in \Phi_{nf}$), we expect to detect the shadows for small values of $\mathcal{T}_d$ and $\mathcal{T}_f$, and specularities for large values of $\mathcal{T}_d$ and $\mathcal{T}_f$.

Penumbra detection. Robust shadow handling requires us to consider the penumbra region as well [8]. Let $\mathcal{S}$ be the set of all pixels that have been identified previously as being a shadow pixel. We search in the neighborhood of $\mathcal{S}$ to mark pixels where there is a significant difference across the flash/no-flash image pair in the gradient domain. We denote $\nabla\mathbf{I}_{nf}(\mathbf{p})$ and $\nabla\mathbf{I}_f(\mathbf{p})$ as the magnitude of gradient for the pixel $\mathbf{p}$ in the no-flash and flash images, respectively. We test pixels within a $\mathcal{N} \times \mathcal{N}$ neighborhood of $\mathcal{S}$, and mark them to be in the penumbra if

$$\mathcal{T}_g(\mathbf{p}) = \|\nabla\mathbf{I}_{nf}(\mathbf{p}) - \nabla\mathbf{I}_f(\mathbf{p})\|_2^2 \;\; > \;\; \tau_g,$$

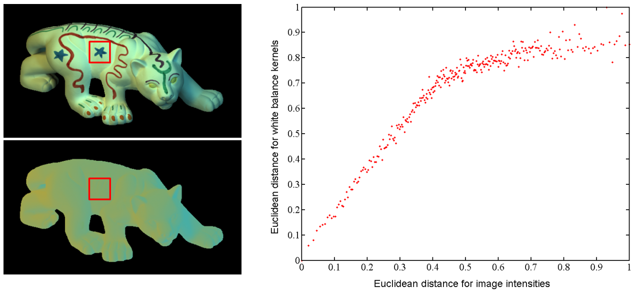
where we set $\tau_g = 0.1$ and $\mathcal{N} = 3$ pixels in this paper.

Once we have an estimate of the detected regions for shadows and specularities, we can interpolate the white balancing kernel from neighboring regions. We discuss the interpolation scheme in Section 4.3.

4.2. Motion compensation

We next outline a procedure to handle motion between the flash and no-flash photographs. We aim to handle two kinds of motions: rigid motion (due to camera movement), and non-rigid motion (movement of the subjects in the scene).

Camera motion. Camera motion, typically due to hand shake, can be approximated as a rigid motion; we align flash and no-flash image by finding SIFT features [23] and computing the homography model using RANSAC [13]. Since SIFT features are robust to illumination changes, they are



(a) Input image and white balance kernels

(b) Pixel similarity vs. white balance similarity

Figure 3: We observe that the pixel intensities and their corresponding white balance kernels show an appropriate linear relationship. This motivates our white balance kernel interpolation strategy.

able to find matching features between the no-flash and flash images. As a result, processing the registered images using our white balance algorithm can robustly reconstruct the white balance kernels.

Scene motion. While the above step accounts for global rigid motion, we still need to handle residual non-rigid motion caused by parallax and/or scene motion. This would require estimating per-pixel optical flow – a challenging problem that is made harder by the strong illumination changes between the flash and no-flash images. Instead, we first align the flash and no-flash image using rigid motion and compute an initial set of white balance kernel estimates. As in the case of shadows and specularities, we detect regions of the images where the white balance kernels might be corrupted by inaccurate motion, and correct them with an interpolation scheme. To detect the pixels with non-rigid motion, we exploit the displacements for the pixels in no-flash images by using forward and backward SIFT flow [22]. In particular, we first compute the forward flow for the rigidly aligned no-flash and flash images. We exploit the obtained warped no-flash image by computing the backward flow with respect to the rigidly aligned no-flash image. We denotes the forward and backward flows as $\mathcal{F}_f = [\mu_f, \nu_f]$ and $\mathcal{F}_b = [\mu_b, \nu_b]$, respectively. For a pixel $\mathbf{p}$, we identify non-rigid motion by using the metric defined as:

$$\mathcal{T}_{\mathcal{F}}(\mathbf{p}) = \|\mu_f(\mathbf{p}) + \mu_b(\mathbf{p})\|_2^2 + \|\nu_f(\mathbf{p}) + \nu_b(\mathbf{p})\|_2^2.$$

As is to be expected, the pixels with non-rigid motion can be detected by thresholding $\mathcal{T}_{\mathcal{F}}$ with large values. In this paper, we set the threshold as 5 pixels.

4.3. White balance kernel interpolation

Once we have detected pixels in the scene that violate any of the assumptions of our algorithm (i.e., flash specularities, flash shadows, or errors in alignment), we propose a simple interpolation scheme to correct the erroneous white balance kernel estimates at these pixels. Our interpolation is based on the intuition that two pixels that are spatially close and have similar appearance in the no-flash image, are likely to have similar reflectance and similar incident illumination, and therefore the same white balance kernel. We empirically tested this assumption on a ground truth image constructed by combining images captured under varying illumination from the MIT intrinsic image dataset [17]. For pixels in a local neighborhood, Figure 3 plots the difference in the ground truth white balance kernels, $\|\mathbf{W}(\mathbf{p}_i) - \mathbf{W}(\mathbf{p}_j)\|$, against the difference in image appearance, $\|\mathbf{I}_{nf}(\mathbf{p}_i) - \mathbf{I}_{nf}(\mathbf{p}_j)\|$. There is an approximately linear relationship between the white balance kernels and image intensities, and for pixels with similar intensities, the corresponding white balance kernels within a local neighborhood are always close to each other. We exploit this using a strategy that interpolates the white balance at erroneous pixels from their nearest neighbors defined using both spatial location and appearance in the no-flash image.

For a pixel $\mathbf{p}$ that is marked as corrupted due to shadow/specularity/motion, we first select K nearest neighbors from unmarked regions. For a pixel $\widehat{\mathbf{p}}$, we denote it as the neighbor of $\mathbf{p}$ if

$$\alpha\|\mathbf{I}_{nf}(\mathbf{p}) - \mathbf{I}_{nf}(\widehat{\mathbf{p}})\|_2^2 + \beta\|\mathbf{p} - \widehat{\mathbf{p}}\|_2 \ < \tau_{\mathcal{N}}$$

where α and β denote the corresponding weights to balance to the spatial location and intensities similarities. For each neighbor, we calculate the interpolation weights as:

$$w_i(\mathbf{p}_i) = 1 - \frac{\|\mathbf{I}_{nf}(\mathbf{p}_i) - \mathbf{I}_{nf}(\mathbf{p})\|_2^2}{\sum_{i=1}^{K}\|\mathbf{I}_{nf}(\mathbf{p}_i) - \mathbf{I}_{nf}(\mathbf{p})\|_2^2},$$

where $\mathbf{p}_i$ denotes the i^{th} neighbor for pixel $\mathbf{p}$. We normalize the weights to make the sum equal to one

$$\widehat{w}_i(\mathbf{p}_i) = \frac{w_i(\mathbf{p}_i)}{\sum_i w_i(\mathbf{p}_i)}.$$

The final white balance kernel at the pixel $\mathbf{p}$ is estimated as:

$$W^c(\mathbf{p}) = \sum_{i=1}^{K} \widehat{w}_i(\mathbf{p}_i) W^c(\mathbf{p}_i).$$

Interpolation is done, starting from the edges of the erroneous regions and continues into the interior (ensuring that every pixel has a set of neighbors to interpolate the kernels from). As observed from Figures 7 and 8, artifacts in the initial white balanced results caused by shadows, specularities or motion, can be effectively eliminated by using the white balance kernel interpolation.

5. Results and Discussion

We evaluate the performance of our proposed white balance technique using both the data collected under laboratory illumination with ground truth comparisons, and on real-world scenes under complex natural illumination. We also demonstrate the robustness of our solution to cast shadows, specularities and motion. We refer the reader to the supplemental material for additional results. To acquire the color of the flash, we place the color-checker chart in the scene and collect the image with flash on and all the other ambient lights off.

5.1. Evaluation of white balance algorithms

Controlled laboratory tests. In order to quantitatively evaluate the performance of our technique, we constructed a ground truth dataset from the images in the MIT Intrinsic Images database [17]. Each object in the MIT database was captured with a single light source positioned in different directions. We generated a no-flash image by modulating each of these images with arbitrary light colors, and adding them together. The corresponding ground-truth white balance results were produced by combining the images without applying any light color to them. For each object, we also chose the image lit by a frontal light source as the pure flash image, and added it to the no-flash result to generate the flash input image. Note that our technique does not make any assumptions about the natural of the scene illumination or the shading induced by the flash illumination, making this dataset an appropriate test for our technique.

Figure 4 evaluates the performance of white balance algorithms as the number of lights in the scene increases. In particular, we evaluate against the single-image single illuminant technique of Gijsenij et al. [15], the single-image local color shifts algorithm of Ebner [7], the mixed illuminant technique of Hsu et al. [19], and the flash photography based, single illuminant method of Petschnigg et al. [25]. Hsu et al. [19] rely on knowledge of the color of the illuminants in the scene; in our tests, we assigned them as combinations of the ground truth lights (e.g, for tests with 8 light sources, we set the two illuminants colors as the means of two sets of 4 lights since these lights get mixed at each surface point). For Petschnigg et al. [25], we supplied the same flash image that we use as an input. As can be seen here, our algorithm outperforms all the previous state-of-the-art techniques we compare against. In addition, the performance of our algorithm stays consistent even as the number of lights varies significantly; this is because the derivation of our closed-form solution makes no assumptions about the scene lighting. Other algorithms produce errors that vary greatly as the number of lights changes. In particular, Hsu et al. [19] exhibit a dramatic increase in error – a result of their technique assuming only two lights are present in the scene. Interestingly, most of the other techniques improve as the number of lights increase; we believe this is because as more lights are mixed, the aggregate approaches an approximately "white" illuminant (especially for largely convex objects).

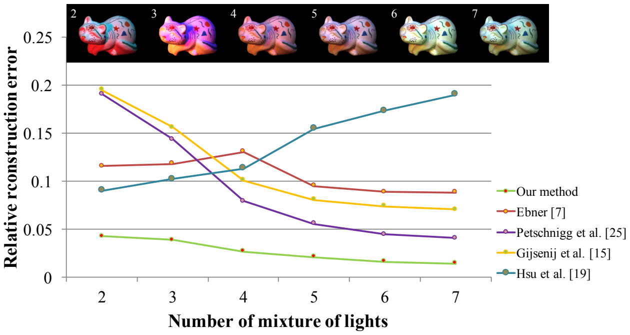
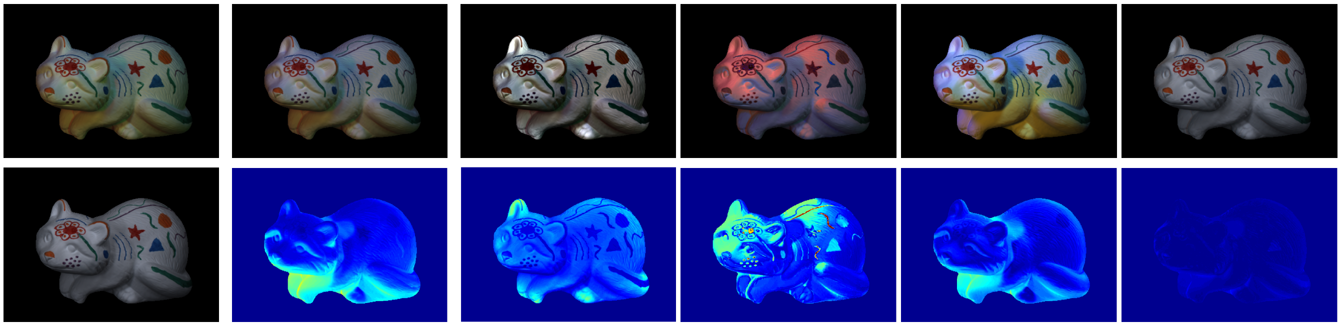


Figure 4: **Laboratory scene evaluation with varying number of lights.** We evaluate the performance of different white balance algorithms on objects from the MIT database [17] when imaged under different numbers of illuminants. For each object, we compute average RMSE (compared to the ground truth) over 5 different combination of the light colors. The insets show example input no-flash images for varying number of lights.

In Figure 5, we visually illustrate the performance of various algorithms for an image captured under eight lights in the scene. As expected, the single-light technique of Gijsenij et al. [15] produces poor results. Ebner's method [7] is better, but produces washed out colors; this may be attributed to the removal of orthogonal color components which are invaluable for precise white balance kernel estimation. The method of Hsu et al. [19] cannot cope with more than two light sources and produces very poor results. Like us, Petschnigg et al. [25] use flash photography for white-balancing; however, their technique is built on a single illuminant model, and only eliminates the average color of eight light sources (light yellow). In contrast, our method produces a result that is very close to the ground truth.

Table 1 summarizes our performance on a number of objects in the MIT Intrinsic Images database [17] for the specific cases of two or three lights. As can be seen from this table, our technique outperforms every single technique on every individual scene.

Real-world tests. We have also tested our algorithm on real-world indoor scenes with mixtures of natural outdoor illumination and indoor fluorescent and incandescent illuminants. We captured several shots of each scene with a color-checker chart placed at different locations; this allows us to evaluate our algorithm's ability to handle spatially-varying mixtures of incident illumination. Figure 6 shows the white balance results on one such scene and compares



(a) No-flash image and ground truth (b) Gijsenij et al. [15] (0.071) (c) Ebner [7] (0.091) (d) Hsu et al. [19] (0.166) (e) Petg et al. [25] (0.057) (f) Our result (0.008)

Figure 5: **Laboratory scene evaluation.** We compare the performance of our technique with a variety of white balance algorithms including Gijsenij et al. [15], Ebner's method [7], Hsu et al. [19], and Petschnigg et al. [25]. The input image (a, top) and corresponding ground truth (a, bottom) were generated using 8 lights in the scene. We show the white balance results (top) and errors with respect to the ground truth (bottom). The numbers in parentheses indicate the average RMSE error. Our technique produces results that are very close to the ground truth, and signficantly better than all other techniques.

dataset	**Gijsenij et al. [15]**		**Ebner [7]**		**Petschnigg et al. [25]**		**Hsu et al. [19]**		**proposed**	
	2 lights	3 lights	2 lights	3 lights	2 lights	3 lights	2 lights	3 lights	2 lights	3 lights
apple	0.145	0.103	0.163	0.141	0.146	0.067	0.103	0.084	0.040	0.035
box	0.259	0.091	0.147	0.107	0.269	0.082	0.107	0.189	0.060	0.028
cup1	0.145	0.062	0.038	0.030	0.145	0.064	0.012	0.140	0.009	0.016
frog1	0.251	0.132	0.140	0.089	0.233	0.122	0.124	0.143	0.036	0.024
panther	0.171	0.070	0.229	0.039	0.169	0.068	0.101	0.238	0.017	0.017
pear	0.145	0.117	0.142	0.135	0.116	0.070	0.077	0.175	0.046	0.050
potato	0.163	0.127	0.147	0.132	0.166	0.047	0.092	0.152	0.047	0.019

Table 1: **Ground truth evaluation for white balance algorithms.** Shown are aggregate statistics over the trails with different number of lights. For each method, we report the reconstruction error with respect to the ground truth.

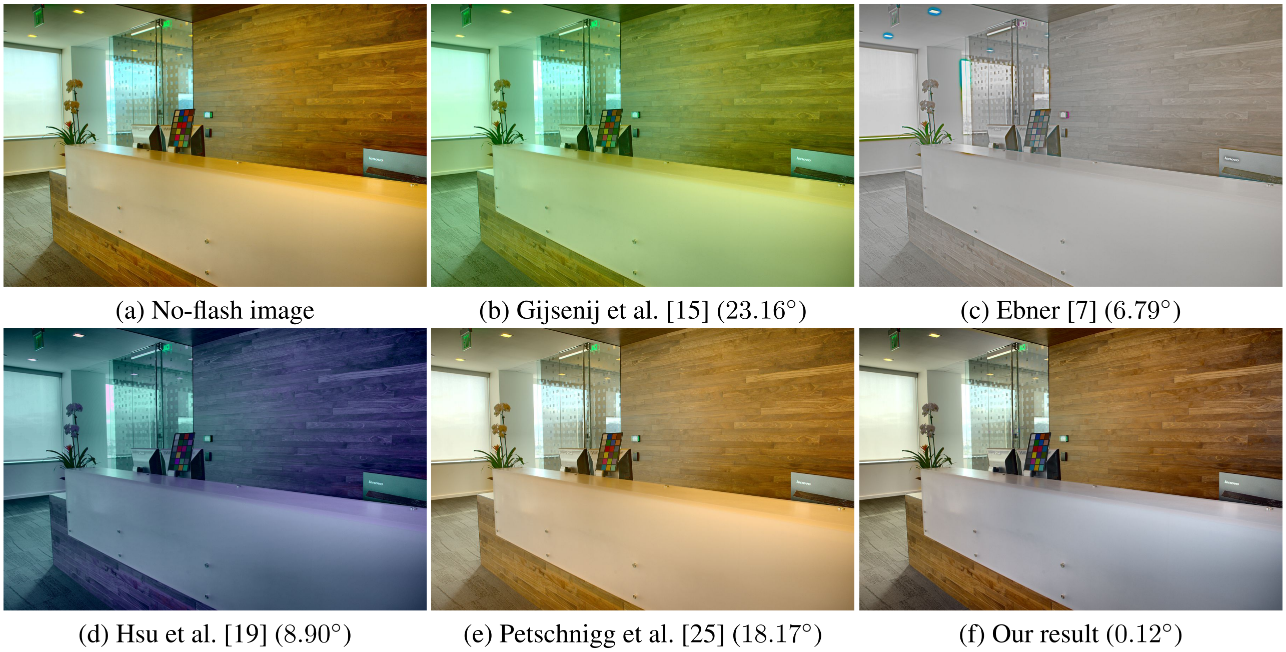
our results with previous techniques. The performance of these techniques on this dataset closely parallels the results we observed in the laboratory dataset. Techniques that rely on single illuminant assumptions ([15] and [25]) lead to significant errors. Hsu et al. [19] handle two light sources whose colors need to be specified by the user. We manually annotated these colors but still got poor results on these scenes. In addition, we found that the output results are very sensitive to the light colors specified. The technique of Ebner [7] has the most flexibility in terms of dealing with spatially-varying lighting. However, by leveraging a flash image, we are able to produce results that are, in terms of angular error on the embedded color-checker chart, an order of magnitude better than Ebner [7].

5.2. Handling shadows, specularities, and motion

We have also evaluated the ability of our proposed compensation scheme to correct errors caused by model violations like flash shadows, flash specularities, as well as the camera and scene motion.

Shadow and specularities. Figure 7 shows a scene with large depth discontinuities and objects with glossy reflectances. The objects in the foreground cast shadows on the background when they are lit by the flash. In addition, they exhibit small specular highlights under flash lighting. These shadows and specular highlights lead to visual artifacts in the white balanced result. However, our detection scheme is able to isolate these regions, and our white balance kernel interpolation scheme corrects these errors to produce visually pleasing results. Note that, this scene also has cast shadows and specular highlights caused by the scene illumination. Our technique is not affected by shadows already present in the scene because it does not make any assumptions about scene geometry or shading. While the specular highlights caused by scene illumination do violate our Lambertian shading model, we have found that they typically do not lead to artifacts in the white balanced result. This can be seen in Figure 7 and is corroborated by our other results.

Camera and scene motion. Figure 8 shows an example of a scene with both camera and scene motion between the flash and no-flash images. Naively applying our technique on these images leads to significant errors in regions where there are reflectance and/or geometry discontinuities. Rigidly aligning the images improves the result, especially near edges in the background. However, the motion of the person in the scene leads to visual artifacts. By detecting the regions that underwent non-rigid motion, and interpolating the white balance kernels in these regions from the rest of the image, we are able to eliminate most of these artifacts.



(a) No-flash image

(b) Gijsenij et al. [15] (23.16°)

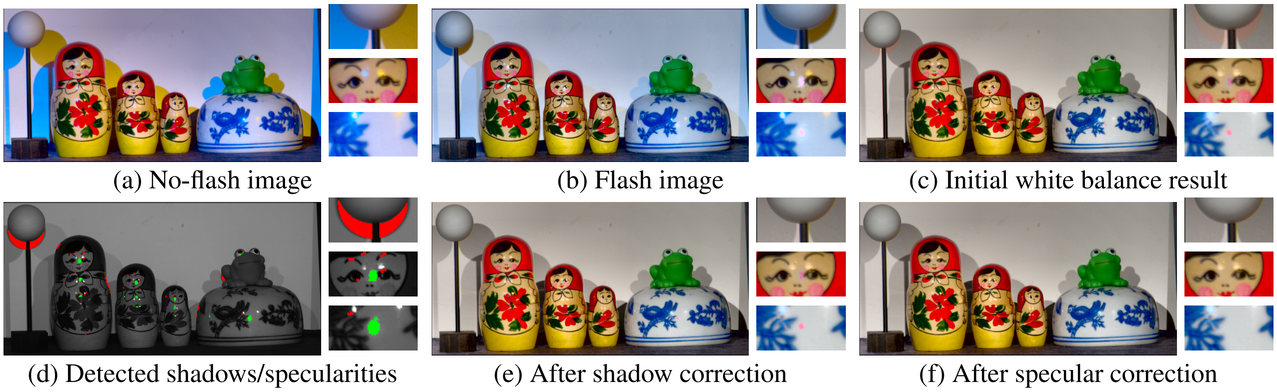
(c) Ebner [7] (6.79°)

(d) Hsu et al. [19] (8.90°)

(e) Petschnigg et al. [25] (18.17°)

(f) Our result (0.12°)

Figure 6: **Real-world scene evaluation.** We test different white balance algorithms on a real-world scene with mixed outdoor and indoor illumination (a). We evaluate the accuracy of each technique using the angle between the white color observed on the color-checker chart (embedded in the scene) to true white color. This error is shown in parentheses. As can be seen, our technique produces errors that are almost two orders of magnitude better than the next best result.



(a) No-flash image

(b) Flash image

(c) Initial white balance result

(d) Detected shadows/specularities

(e) After shadow correction

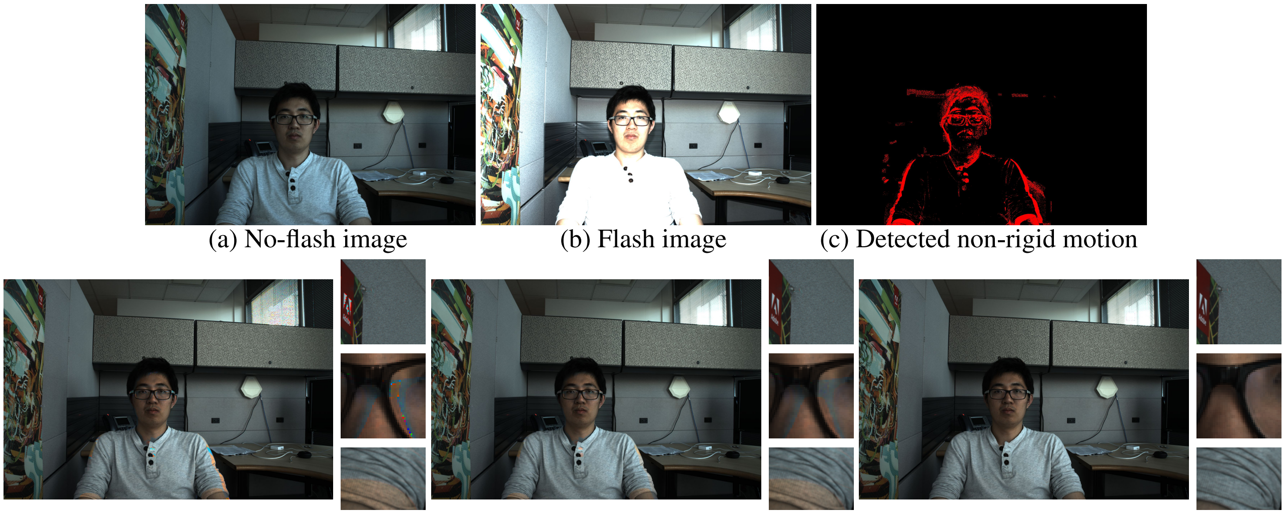
(f) After specular correction

Figure 7: **Shadow/specularity detection and correction.** In this scene, the flash lighting (b) creates shadows and specular highlights that are not present in the no-flash image (a), leading to artifacts in the white balanced result (c). By detecting these regions (d, shadows shown in red, specularities in green), and correcting them (e,f), we can produce high quality results. Note that the objects also have shadows and specular highlights from scene illumination (see the middle inset); however, these do not create issues in the final result.

5.3. Discussion

Flash calibration. Our technique relies on knowing the color of the flash lighting. This can be estimated with a one-time calibration step. However, if the color of the flash is not known, we can still apply our technique to produce results that are accurate up to a *single* global white balance kernel. Any single illuminant white balance technique can be used to remove this remaining ambiguity. Note that our technique does not depend on the shading induced by the flash and can therefore generalize to any shape, size, form of flash lighting.

Limitations. The proposed technique requires linear images (or equivalently, radiometrically calibrated camera response). Our technique also assumes that the flash lighting illuminates all the scene points, which is violated for the scenes that have regions at large distance to the camera. The


(a) No-flash image (b) Flash image (c) Detected non-rigid motion

(d) Initial white balance results (e) After rigid alignment (f) After scene motion correction

Figure 8: **Handling camera and scene motion.** The input no-flash (a) and flash (b) images are misaligned due to both camera and scene motion, resulting in artifacts in the white balanced result (d). Rigidly aligning the image pair improves the results (e). Using our flow metric, we can detect non-rigid motion (c), and correct the white balance kernel in these regions to produce the final artifact-free result (f).

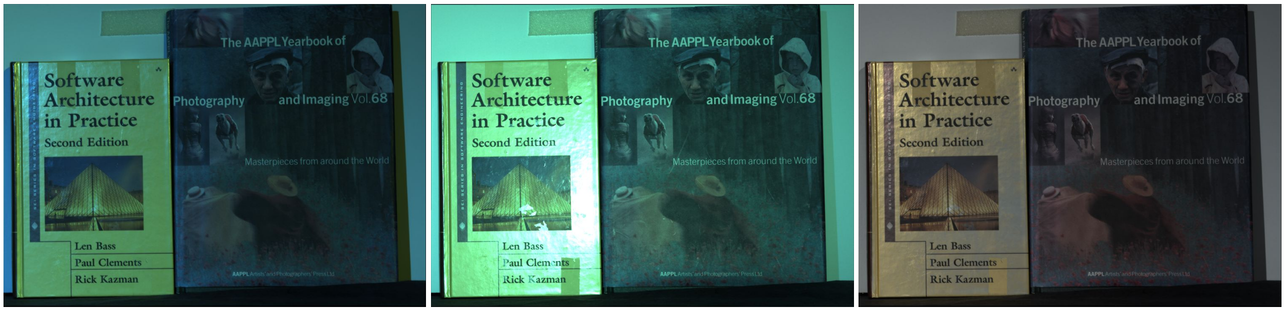

Software Architecture in Practice
Second Edition
Len Bass
Paul Clements
Rick Kazman
The AAPPL Yearbook of Photography and Imaging Vol.68
Masterpieces from around the World
(a) No-flash image (b) Flash image (c) Our result

Figure 9: **A failure case due to saturation of a large region.** In this scene, we test on objects with Lambertian (book in the right) and non-Lambertian reflectance (book in the left). The flash lighting (b) introduces large region of saturated pixels due to the non-Lambertian reflectance, resulting in artifacts in the white balanced result (c).

Lambertian shading model assumed in the paper is restrictive for many real-world scenes. While we have outlined efficient schemes to handle non-idealities like specularities, these schemes cannot handle large violations of the Lambertian assumption. An example of such failure is shown in Figure 9, where a large saturated region is present in the flash image which leads to failure in the interpolation of the white balance kernel.

Finally, bright illuminants in the scene will require the flash to be equivalently bright. Consider a pixel where the pure flash brightness is F and the scenes intensity level in the no flash image is I. Given that variance of photon noise is equal to the mean intensity level, the noise variance in the no flash and flash images are I and $I + F$, respectively. Hence, the noise variance in the difference image is $2I + F$ (under i.i.d. assumptions), and the SNR associated with the difference image is $\frac{F}{\sqrt{2I+F}}$. Hence, for reliable estimation of the difference image we need F to be proportional to I. This implies that scenes that are extremely bright (for example, outdoor scenes under direct sunlight) are hard to handle with our technique.

6. Conclusions

We have addressed the under-constrained problem of white balance under complex spatially-varying illumination, and shown that using flash photography results in a closed-form solution to this problem that results in state-of-the-art results on a wide range of real life cases. Our technique is automatic and does not rely on assumptions about the scene lighting or user inputs, which are endemic to all past works [3, 4, 15, 19, 25].

It would be interesting to extend our algorithm to other scenarios such as video capture. Also, our use of flash lighting is not special, and only serves as a way to probe the scene with a single illuminant. This might be interesting in outdoor scenes where the sun can be treated as flash illumination. In addition, we would like to explore the use of flash photography for other closely related problems like intrinsic images, and surface reconstruction.

References

[1] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. In *TOG*, volume 24, pages 828–835, 2005.

[2] M. Bleier, C. Riess, S. Beigpour, E. Eibenberger, E. Angelopoulou, T. Tröger, and A. Kaup. Color constancy and non-uniform illumination: Can existing algorithms work? In *ICCVW*, 2011.

[3] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. In *TOG*, volume 28, page 130, 2009.

[4] I. Boyadzhiev, K. Bala, S. Paris, and F. Durand. User-guided white balance for mixed lighting conditions. *TOG*, 31(6):200, 2012.

[5] R. Carroll, R. Ramamoorthi, and M. Agrawala. Illumination decomposition for material recoloring with consistent interreflections. *TOG*, 30(4):43, 2011.

[6] J. M. DiCarlo, F. Xiao, and B. A. Wandell. Illuminating illumination. In *Color and Imaging Conference*, 2001.

[7] M. Ebner. Color constancy using local color shifts. In *ECCV 2004*. 2004.

[8] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. In *TOG*, volume 23, pages 673–678, 2004.

[9] G. Finlayson and S. Hordley. Improving gamut mapping color constancy. *TIP*, 9(10):1774–1783, 2000.

[10] G. D. Finlayson. Color constancy in diagonal chromaticity space. In *ICCV*, 1995.

[11] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *PAMI*, 23(11):1209–1221, 2001.

[12] G. D. Finlayson, S. D. Hordley, and I. Tastl. Gamut constrained illuminant estimation. *IJCV*, 67(1):93–109, 2006.

[13] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *CVPR*, 2008.

[15] A. Gijsenij, T. Gevers, and J. Van De Weijer. Improving color constancy by photometric edge weighting. *PAMI*, 34(5):918–929, 2012.

[16] A. Gijsenij, R. Lu, and T. Gevers. Color constancy for multiple light sources. *TIP*, 21(2):697–707, 2012.

[17] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *CVPR*, 2009.

[18] S. He and R. W. Lau. Saliency detection with flash and no-flash image pairs. In *ECCV*. 2014.

[19] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. In *TOG*, volume 27, page 70, 2008.

[20] H. R. V. Joze and M. S. Drew. Exemplar-based color constancy and multiple illumination. *PAMI*, 36(5):860–873, 2014.

[21] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. *TOG*, 25(3):646–653, 2006.

[22] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[24] C. Lu and M. S. Drew. Practical scene illuminant estimation via flash/no-flash pairs. In *Color and Imaging Conference*, 2006.

[25] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *TOG*, 23(3):664–672, 2004.

[26] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *TOG*, volume 23, pages 679–688, 2004.

[27] C. Riess, E. Eibenberger, and E. Angelopoulou. Illuminant color estimation for real-world mixed-illuminant scenes. In *ICCVW*, 2011.

[28] H. Seo and P. Milanfar. Computational photography using a pair of flash/no-flash images by iterative guided filtering. In *ICCV*, 2011.

[29] J. Shen, X. Yang, Y. Jia, and X. Li. Intrinsic images using optimization. In *CVPR*, 2011.

[30] J. Sun, J. Sun, S. B. Kang, Z.-B. Xu, X. Tang, and H.-Y. Shum. Flash cut: Foreground extraction with flash and no-flash image pairs. In *CVPR*, 2007.

[31] S. Zhuo, D. Guo, and T. Sim. Robust flash deblurring. In *CVPR*, 2010.