Computational Photography Using a Pair of Flash/No-flash Images by Iterative Guided Filtering

Hae Jong Seo and Peyman Milanfar
Electrical Engineering Department
University of California, 1156 High Street,
Santa Cruz, CA, 95064
{rokaf, milanfar}@soe.ucsc.edu

Abstract

In this paper, we present an iterative improvement of the guided image filter for flash/no-flash photography. The guided filter [8] utilizes a guide image to enhance a corrupted input image as similarly done in the joint bilateral filter [3, 12]. The guided filter has proved to be effective for such applications as high dynamic range compression, image matting, haze removal, and flash/no-flash denoising etc. In this paper, we analyze the spectral behavior of the guided filter kernel in matrix formulation and introduce a novel iterative application of the guided filtering which significantly improves it. Iterations of the proposed method consist of a combination of diffusion and residual iterations. We demonstrate that the proposed approach outperforms state of the art methods in both flash/no-flash image denoising and deblurring.

1. Introduction

Recently, several techniques [12, 3, 1, 21] to enhance the quality of flash/no-flash image pairs have been proposed. The no-flash image tends to have a relatively low signal-to-noise ratio (SNR) while containing the natural ambient lighting of the scene. The key idea of flash/no-flash photography is to create a new image that is closest to the look of the real scene by having detail from the flash image and the ambient illumination of the no-flash image. Eisemann and Durand [3] used (joint) bilateral filtering [17] to give the flash image the ambient tones from the no-flash image. On the other hand, Petschnigg et al. [12] focused on reducing noise in the no-flash image and transferring details from the flash image to the no-flash image by applying joint (or cross) bilateral filtering. Agrawal et al. [1] tried to remove flash artifacts, but did not test their method on no-flash images containing severe noise. As opposed to a visible flash used in [3, 12, 1], recently Krishnan and Fergus [11] used both near-infrared and near-ultraviolet illumination for low light image enhancement. Their so-called "dark flash" provides high-frequency detail in a less intrusive way than a visible flash does even though it results in incomplete color information. All these methods ignored any blur, by either depending on a tripod setting or choosing sufficiently fast shutter speed. However, in practice, the captured images under low-light conditions using a hand-held camera often suffer from motion blur caused by camera shake. More recently, Zhuo et al. [21] proposed a flash deblurring method that recovers a sharp image by combining a blurry image and a corresponding flash image. They integrated a so-called flash gradient into a maximum-a-posteriori framework and solved the optimization problem by alternating between blur kernel estimation and sharp image reconstruction. This method outperformed many state of the art single image deblurring [14, 4, 20] and color transfer methods [15]. However, the final output of this method is not entirely free of artifacts because the model only deals with a spatially invariant motion blur.

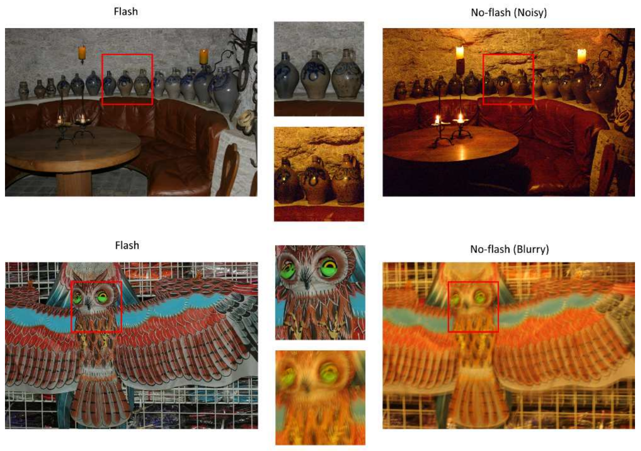


Figure 1. Flash/no-flash pairs. No-flash image can be noisy or blurry.

Others have used multiple pictures of a scene taken at

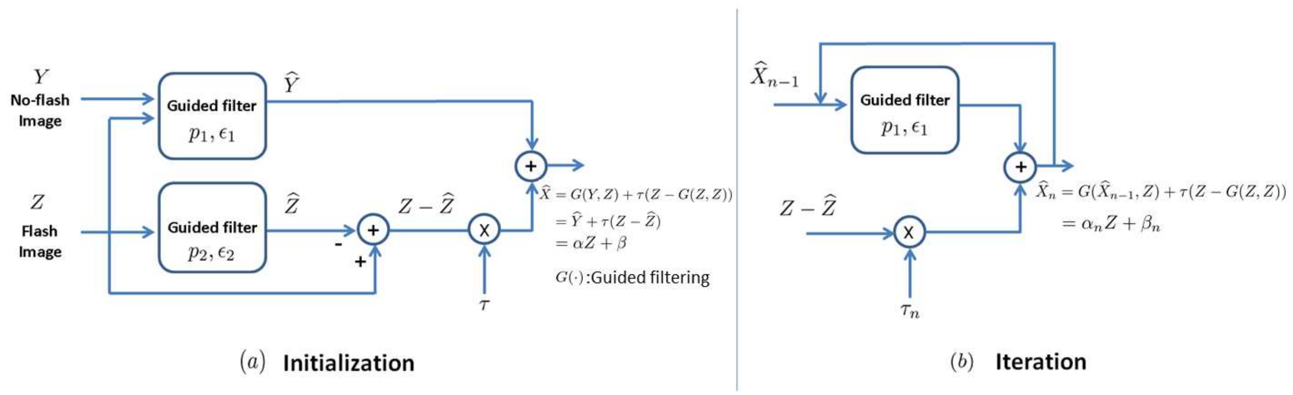

Y
No-flash Image
Guided filter p_1, ϵ_1
$\widehat{Y}$
Z
Flash Image
Guided filter p_2, ϵ_2
$\widehat{Z}$
-
+
$Z - \widehat{Z}$
X
τ
$\widehat{X} = G(Y,Z) + \tau(Z - G(Z,Z))$
$= \widehat{Y} + \tau(Z - \widehat{Z})$
$= \alpha Z + \beta$
$G(\cdot)$:Guided filtering
(a) Initialization
$\widehat{X}_{n-1}$
Guided filter p_1, ϵ_1
$Z - \widehat{Z}$
X
τ_n
$\widehat{X}_n = G(\widehat{X}_{n-1}, Z) + \tau(Z - G(Z,Z))$
$= \alpha_n Z + \beta_n$
(b) Iteration

Figure 2. Overview of our algorithms for flash/no-flash enhancement.

different exposures to generate high dynamic range images. This is called multi-exposure image fusion [6] which shares some similarity with our problem in that it seeks a new image that is of better quality than any of the input images. However, flash/no-flash photography is generally more difficult since that there are only a pair of images. It is still a challenging open problem to enhance a low SNR no-flash image with a spatially variant motion blur only with the help of a single flash image.

In this paper, we propose a unified iterative framework that deals with both denoising and deblurring. Before we begin a more detailed description, we highlight some novel aspects of the proposed framework.

- As opposed to [3, 12] which relied on the (joint) bilateral filter, our approach adopts the guided filter [8] that has proved to be superior.
- We improve the performance of the guided filter by analyzing its spectral behavior and applying it iteratively.
- We show that iterative application of the guided filter corresponds to a combination of two iterative processes applied to the two given images: a nonlinear anisotropic diffusion to the no-flash image and a nonlinear residual itteratoin applied to the flash image.
- While [3, 12, 1, 21, 11] demonstrated their results on either noise or blur case, we show that our method produces a high-quality output in both cases, and outperforms state of the art methods.

2. Overview of the Proposed Approach

We address the problem of generating a high quality image from two captured images: a flash image (Z) and a no-flash image (Y) (See Fig 1.) The task at hand is to generate a new image (X) that contains the ambient lighting of the no-flash image (Y) and preserves the details of the flash-image (Z). As in [12], the new image X can be decomposed into two layers; a base layer and a detail layer:

$$\widehat{X} = \underbrace{\widehat{Y}}_{base} + \tau \underbrace{(Z - \widehat{Z})}_{detail}. \tag{1}$$

Here, Y might be noisy or blurry (possibly both), and $\widehat{Y}$ is an estimated version of Y, enhanced with the help of the flash image. $\widehat{Z}$ represents a nonlinear, (low-pass) filtered version of Z so that $Z - \widehat{Z}$ can provide details. Note that τ is a small constant that strikes a balance between the two parts. In order to estimate $\widehat{Y}$ and $\widehat{Z}$, we employ local linear minimum mean square error (LMMSE) predictors[1] which generalize the idea of *guided filtering* as proposed in [8]. More specifically, we assumed that $\widehat{Y}$ and $\widehat{Z}$ are linear functions of Z in a window ω_k centered at the pixel k:

$$\widehat{y}_i = az_i + b, \quad \widehat{z}_i = cz_i + d, \quad \forall i \in \omega_k, \tag{2}$$

where $\widehat{y}_i, \widehat{z}_i, z_i$ are samples of $\widehat{Y}, \widehat{Z}, Z$ respectively at pixel i and (a, b, c, d) are coefficients assumed to be constant in ω_k (a square window of size $p \times p$). Once we estimate a, b, c, d, equation (1) can be rewritten as:

$$\begin{aligned} \widehat{X} &= \widehat{Y} + \tau(Z - \widehat{Z}) = aZ + b + \tau Z - \tau cZ - \tau d, \\ &= (a - \tau(c-1))Z + b - \tau d = \alpha Z + \beta. \end{aligned} \tag{3}$$

In fact, $\widehat{X}$ is a linear function of Z. While it is not possible to estimate α and β directly from this linear model (since they in turn depend on X), the coefficients α, β can be expressed in terms of a, b, c, d which are optimally estimated from two different local linear models shown in equation (2). Naturally, the simple linear model has its limitations in capturing complex behavior. Hence, by initializing $\widehat{X}_0 = Y$, we propose an iterative approach to boost its performance as follows:

$$\begin{aligned} \widehat{X}_n &= G(\widehat{X}_{n-1}, Z) + \tau_n(Z - G(Z,Z)) \\ &= G(\widehat{X}_{n-1}, Z) + \tau_n(Z - \widehat{Z}) = \alpha_n Z + \beta_n, \end{aligned} \tag{4}$$

[1]More detail is provided in Section 4.

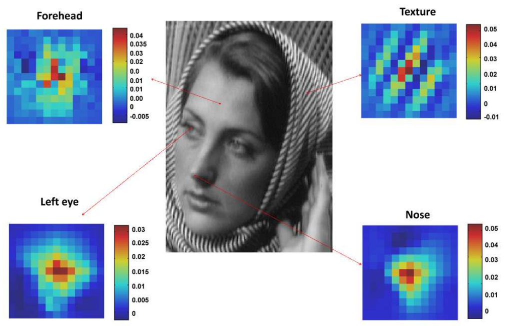


Figure 3. Examples of guided filter kernel weights in four different patches. The kernel weights represent underlying structures well.

where $G(\cdot)$ is LMMSE (guided filtering), and α_n, β_n, and τ_n evolve with the iteration number n. A block-diagram of our approach is shown in Fig. 2. The proposed method effectively removes noise and deals well with spatially variant motion blur without the need to estimate any blur kernel or to accurately register flash/no-flash image pairs when there is a modest displacement between them.

In Section 3, we outline the guided filter and study its statistical properties. We describe how we actually estimate the linear model coefficients α, β, and we provide an interpretation of the proposed iterative framework in matrix form in Section 4. In Section 5, we demonstrate the performance of the system with some experimental results, and finally we conclude the paper in Section 6.

3. The Guided Filter and Its Properties

Several recent space-variant, nonparametric denoising filters such as the bilateral filter [17], non-local means filter [2], and locally adaptive regression kernel filters [16] have been proposed for denoising, where the kernels are directly computed from the noisy image. However, the guided filter can be distinguished from these in the sense that the filter kernel weights are computed from a (second) "guide" image which is presumably cleaner. The idea is to apply filter kernels W_{ij} computed from the guide image (e.g. flash) Z to the more noisy image (e.g. no-flash) Y. Specifically, the filter output sample $\widehat{y}$ at a pixel i is computed as a weighted average:

$$\widehat{y}_i = \sum_j W_{ij}(Z) y_j. \quad (5)$$

Cross (or joint) bilateral filter [3, 12] is another related example of this type of filtering. The guided filter kernel can be explicitly expressed as:

$$W_{ij}(Z) = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} \left(1 + \frac{(z_i - E[Z]_k)(z_j - E[Z]_k)}{\mathbf{var}(Z)_k + \epsilon}\right), \quad (6)$$

where $|\omega|$ is the total number of pixels $(= p^2)$ in ω_k, ϵ is a global smoothing parameter, $E[Z]_k \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} z_l$, and

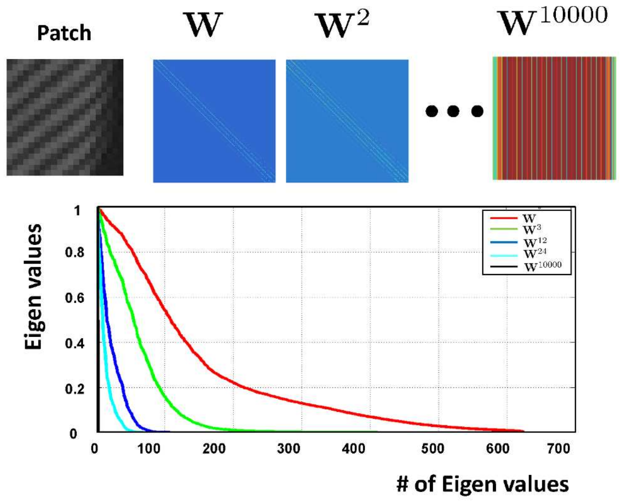


Figure 4. Examples of $\mathbf{W}$ and its powers in a patch of size 25×25. The largest eigenvalue of $\mathbf{W}$ is one and the rank of $\mathbf{W}$ asymptotically becomes one. This figure is better viewed in color.

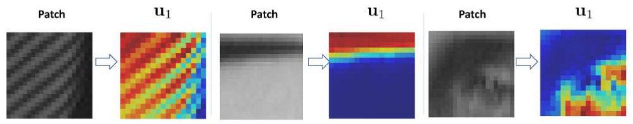


Figure 5. Examples of the 1^{st} left eigenvector $\mathbf{u}$ in three patches. The vector was reshaped into an image for illustration purpose.

$\mathbf{var}(Z)_k \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} z_l^2 - E[Z]_k^2$. Note that W_{ij} are normalized weights, that is, $\sum_{i,j} W_{ij}(Z) = 1$. Fig. 3 shows examples of guided filter weights in four different areas. We can see that the guided filter kernel weights neatly capture underlying geometric structures as do other data-adaptive kernel weights [17, 2, 16].

Next, we study some fundamental properties of the guided filter kernel in matrix form. We adopt a convenient vector form of equation (5) as follows:

$$\widehat{y}_j = \mathbf{w}_j^T \mathbf{y}, \quad (7)$$

where $\mathbf{y}$ is a column vector comprised of the pixels in Y and $\mathbf{w}_j^T = [W(1,j), W(2,j), \cdots, W(N,j)]$ is a vector of weights for each j. Note that N is the dimension of $\mathbf{y}$ ($N \geq p$). Writing the above at once for all j we have,

$$\widehat{\mathbf{y}} = \begin{bmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_N^T \end{bmatrix} = \mathbf{W}(\mathbf{z})\ \mathbf{y}, \quad (8)$$

where $\mathbf{z}$ is a vector comprised of the pixels in Z and $\mathbf{W}$ is only a function of $\mathbf{z}$. The filter output can be analyzed as the product of a matrix of weights $\mathbf{W}$ with the vector of the given input image $\mathbf{y}$.

The matrix $\mathbf{W}$ is symmetric positive definite and the sum of each row of $\mathbf{W}$ is equal to one $(\mathbf{W}\mathbf{1}_N = \mathbf{1}_N)$ by definition. All eigenvalues λ_i $(i = 1, \cdots, N)$ of $\mathbf{W}$

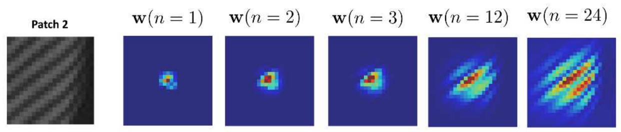


Figure 6. The guided filter kernel matrix $\mathbf{W}$ captures the underlying data structure, but powers of $\mathbf{W}$ provides even better structure by generating larger (but more sophisticated) kernel shapes. $\mathbf{w}$ is a (center) row vector of $\mathbf{W}$. $\mathbf{w}$ was reshaped into an image for illustration purposes.

in a patch of size 25×25 are real, and the largest eigenvalue is exactly one ($\lambda_1 = 1$), with corresponding eigenvector $\mathbf{v}_1 = (1/\sqrt{N})\mathbf{1}_N$ as shown in Fig. 4. Intuitively, this means that filtering by $\mathbf{W}$ will leave a constant signal (i.e., a "flat" image) unchanged. In fact, with the rest of its spectrum inside the unit disk, powers of $\mathbf{W}$ converge to a matrix of rank one, with identical rows which (still) sum to one:

$$\lim_{n\to\infty} \mathbf{W}^n = \mathbf{1}_N \mathbf{u}_1^T. \tag{9}$$

So the dominant left eigenvector $\mathbf{u}_1$ summarizes the asymptotic effect of applying the filter $\mathbf{W}$ many times. Fig. 5 shows what a typical $\mathbf{u}_1$ looks like. The vector was reshaped into an image for illustration purposes. Fig. 6 shows examples of a (center) row vector ($\mathbf{w}^T$) from $\mathbf{W}$'s powers in the same patch as Fig. 4. We can see that powers of $\mathbf{W}$ provide even better structure by generating larger (but more sophisticated) kernels. This insight reveals that applying $\mathbf{W}$ multiple times can possibly improve performance, which leads us to the iterative use of the guided filter. This approach will produce the evolving coefficients α_n, β_n introduced in (4). In the following section, we describe how we actually compute these coefficients based on mean square error (MSE) predictions.

4. Iterative Local LMMSE Predictors

The coefficients[2] a_k, b_k, c_k, d_k in equation (3) are chosen so that "on average" the estimated value $\widehat{Y}$ is close to the observed value of Y (=y_i) in ω_k, and the estimated value $\widehat{Z}$ is close to the observed value of Z (=z_i) in ω_k. To determine these coefficients, we adopt a regularized (stabilized) MSE criterion in the window ω_k as our measure of closeness:

$$\begin{aligned} \mathrm{MSE}(a_k, b_k) &= E[(Y - a_kZ - b_k)^2] + \epsilon_1 a_k^2, \\ \mathrm{MSE}(c_k, d_k) &= E[(Z - c_kZ - d_k)^2] + \epsilon_2 c_k^2, \end{aligned} \tag{10}$$

where ϵ_1 and ϵ_2 are small constants that prevent estimated a_k, c_k from being too large[3]. By setting partial derivatives

[2]Note that k is used to clarify that the coefficients are estimated for the window ω_k.

[3] The effect of the regularization parameters ϵ_1 and ϵ_2 is quite the opposite in each case in the sense that the higher ϵ_2 is, the more detail through $\widehat{z}_i - z_i$ can be obtained; whereas the lower ϵ_1 ensures that the image content in $\widehat{Y}$ is not over-smoothed.

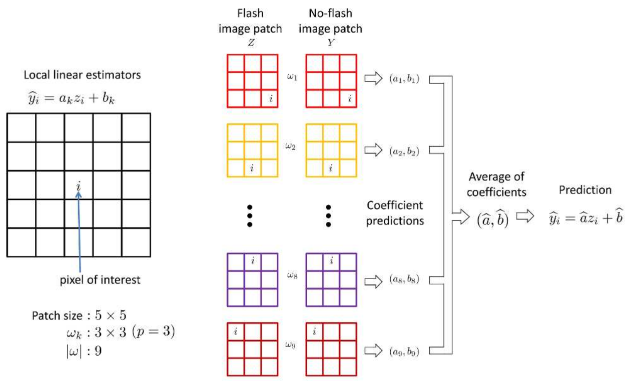


Figure 7. LMMSE: $\hat{a}_k, \hat{b}_k$ are estimated from 9 different windows ω_k and averaged coefficients $\hat{a}, \hat{b}$ are used to predict $\hat{y}_i$. This figure is better viewed in color.

of $\mathrm{MSE}(a_k, b_k)$ with respect to a_k, b_k and partial derivatives of $\mathrm{MSE}(c_k, d_k)$ with respect to c_k, d_k respectively to zero, the solutions to minimum MSE prediction in (10) are

$$\begin{aligned} \widehat{a}_k &= \frac{E[ZY] - E[Z]E[Y]}{E[Z^2] - E^2[Z] + \epsilon_1} = \left[\frac{\mathbf{cov}(Z,Y)}{\mathbf{var}(Z) + \epsilon_1}\right], \\ \widehat{b}_k &= E[Y] - \widehat{a}_k E[Z] = E[Y] - \left[\frac{\mathbf{cov}(Z,Y)}{\mathbf{var}(Z) + \epsilon_1}\right] E[Z], \\ \widehat{c}_k &= \frac{E[Z^2] - E^2[Z]}{E[Z^2] - E^2[Z] + \epsilon_2} = \left[\frac{\mathbf{var}(Z)}{\mathbf{var}(Z) + \epsilon_2}\right], \\ \widehat{d}_k &= E[Z] - \widehat{c}_k E[Z] = E[Z] - \left[\frac{\mathbf{var}(Z)}{\mathbf{var}(Z) + \epsilon_2}\right] E[Z], \end{aligned} \tag{11}$$

where we compute $E[Z] \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} z_l$, $E[Y] \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} y_l$, $E[ZY] \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} z_l y_l$, $E[Z^2] \approx \frac{1}{|\omega|}\sum_{l\in\omega_k} z_l^2$.
Note that the use of different ω_k results in different predictions of these coefficients. For instance, consider a case where we predict $\widehat{y}_i$ using observed values of Y in ω_k of size 3×3 as shown in Fig. 7. There are 9 possible windows that involve the pixel of interest i. Therefore, we must take into account all 9 a_k, b_k's to predict $\widehat{y}_i$. The simple strategy taken by He at al. [8] is to average them as follows:

$$\widehat{a} = \frac{1}{|\omega|}\sum_{k=1}^{|\omega|} \widehat{a}_k, \quad \widehat{b} = \frac{1}{|\omega|}\sum_{k=1}^{|\omega|} \widehat{b}_k. \tag{12}$$

As such, the resulting prediction of $\widehat{Y}$ given $Z = z_i$ is

$$\begin{aligned} \widehat{y}_i &= \widehat{a} z_i + \widehat{b} = \frac{1}{|\omega|}\sum_{k=1}^{|\omega|}(\widehat{a}_k z_i + \widehat{b}_k), \\ \widehat{z}_i &= \widehat{c} z_i + \widehat{d} = \frac{1}{|\omega|}\sum_{k=1}^{|\omega|}(\widehat{c}_k z_i + \widehat{d}_k). \end{aligned} \tag{13}$$

The idea of using these averaged coefficients $\widehat{a}, \widehat{b}$ is analogous to the simplest form of aggregating multiple local estimates from overlapped patches in image denoising and super-resolution literature [13]. The aggregation helps the filter output look locally smooth and contain fewer artifacts.

These local linear models work well when the window size p is small and the underlying data has a simple pattern. However, these models are too simple to deal effectively with more complicated structures, and thus there is a need to use larger window sizes. As we alluded to earlier in (4), the estimation of these linear coefficients in an iterative fashion can handle significantly more complex behavior of the image content as follows:

$$\begin{aligned}\widehat{X}_n &= (\widehat{a}_n - \tau_n(\widehat{c}-1))Z + \widehat{b}_n - \tau_n\widehat{d}, \\ &= \widehat{\alpha}_n Z + \widehat{\beta}_n, \end{aligned} \tag{14}$$

where n is the iteration number and $\tau_n > 0$ is set to be a monotonically decaying function[4] of n so that $\sum_{n=1}^{\infty}\tau_n$ converges. This iteration is closely related to *diffusion* and *residual iteration* which are two fundamental smoothing methods which we briefly describe below.

Recall that equation (14) can also be written in matrix form as done in Section 3:

$$\widehat{\mathbf{x}}_n = \underbrace{\mathbf{W}\widehat{\mathbf{x}}_{n-1}}_{\text{base layer}} + \tau_n \underbrace{(\mathbf{z} - \mathbf{W}_d\ \mathbf{z})}_{\text{detail layer}}, \tag{15}$$

where $\mathbf{W}$ and $\mathbf{W}_d$ are guided filter kernel matrices constructed from the guided filter kernels W and W_d respectively. The difference between W and W_d lies in one parameter (ϵ_2 of $W_d > \epsilon_1$ of W). Explicitly writing the iterations, we observe:

$$\begin{aligned}
\widehat{\mathbf{x}}_0 &= \mathbf{y} \\
\widehat{\mathbf{x}}_1 &= \mathbf{W}\mathbf{y} + \tau_1(\mathbf{z} - \mathbf{W}_d\mathbf{z}), \\
\widehat{\mathbf{x}}_2 &= \mathbf{W}\widehat{\mathbf{x}}_1 + \tau_2(\mathbf{z} - \mathbf{W}_d\mathbf{z}), \\
&= \mathbf{W}^2\mathbf{y} + (\tau_1\mathbf{W} + \tau_2\mathbf{I})(\mathbf{z} - \mathbf{W}_d\mathbf{z}), \\
&\ \ \vdots \\
\widehat{\mathbf{x}}_n &= \mathbf{W}\widehat{\mathbf{x}}_{n-1} + \tau_n(\mathbf{z} - \mathbf{W}_d\mathbf{z}), \\
&= \mathbf{W}^n\mathbf{y} + (\tau_1\mathbf{W}^{n-1} + \tau_2\mathbf{W}^{n-2} + \cdots + \tau_n\mathbf{I})(\mathbf{z} - \mathbf{W}_d\mathbf{z}), \\
&= \underbrace{\mathbf{W}^n\mathbf{y}}_{\text{diffusion}} + \underbrace{P_n(\mathbf{W})(\mathbf{z} - \mathbf{W}_d\mathbf{z})}_{\text{residual iteration}},
\end{aligned} \tag{16}$$

where P_n is a polynomial function of $\mathbf{W}$, and $\mathbf{I}$ is an Identity matrix. The first term $\mathbf{W}^n\mathbf{y}$ in equation (16) is an anisotropic *diffusion* process that enhances SNR. The second term is the *residual iteration* [18]. The key idea behind this iteration is to filter the residual $(Z - \widehat{Z})$ to extract detail. By combining *diffusion* and *residual iteration*, we achieve an image between the flash image $\mathbf{z}$ and the no-flash image $\mathbf{y}$, but of better quality than both.

5. Experimental Results

In this section, we apply the proposed approach to flash/no-flash image pairs for denoising and deblurring. We convert images Z and Y from RGB color space to YCbCr, and perform iterative filtering separately in each resulting channel. The final result is converted back to RGB space for display. Note that all figures in this section are better viewed in color.

[4]We use $\tau_n = \frac{1}{n^2}$ for the all experiments.

5.1. Flash/No-flash Denoising

5.1.1 Visible Flash [12]

We show experimental results on two flash/no-flash image pairs. We compare our results with the method based on joint bilateral filter [12] in Fig. 8. Our proposed method effectively denoised the no-flash image while transferring the fine detail of the flash image and maintaining the ambient lighting of the no-flash image. We point out that the proposed iterative application of the guided filtering yielded much better results than one time iteration of either the joint bilateral filtering [12] or the guided filter [8].

5.1.2 Dark Flash [11]

Here, we use the *dark flash* approach of [11]. Let us call the dark flash image Z. Dark flash may introduce shadows and specularities in images, which affect the results of both the denoising and detail transfer. We detect those regions using the same methods proposed by [12]. Shadows are detected by finding the regions where $|Z - Y|$ is small and specularities are found by detecting saturated pixels in Z. After combining the shadow and specularities mask, we blur it using a Gaussian filter to feather the boundaries. By using the resulting mask, the output $\widehat{X}_n$ at each iteration is alpha-blended with a (nonlinear) low-pass filter version of Y as similarly done in [12, 11]. In order to realize ambient lighting conditions, we applied the same mapping function to the final output as in [11]. Fig. 9 shows that our results yield better detail with less color artifacts.

5.2. Flash/No-flash Deblurring

Motion blur due to camera shake is an annoying yet common problem in low-light photography. Our proposed method can also be applied to flash/no-flash deblurring. We show experimental results on two flash/no-flash image pairs where no-flash images suffer from mild noise and strong motion blur. We compare our method with Zhuo et al. [21]. As shown in Fig. 10, our method outperforms [21], obtaining much finer details with better color contrast even though our method does not estimate a blur kernel at all. The results of [21] tend to be somewhat blurry and distort the ambient lighting of the real scene.

6. Conclusion and Future Work

We analyzed the spectral behavior of the guided filter kernel using a matrix formulation and improved its perfor-

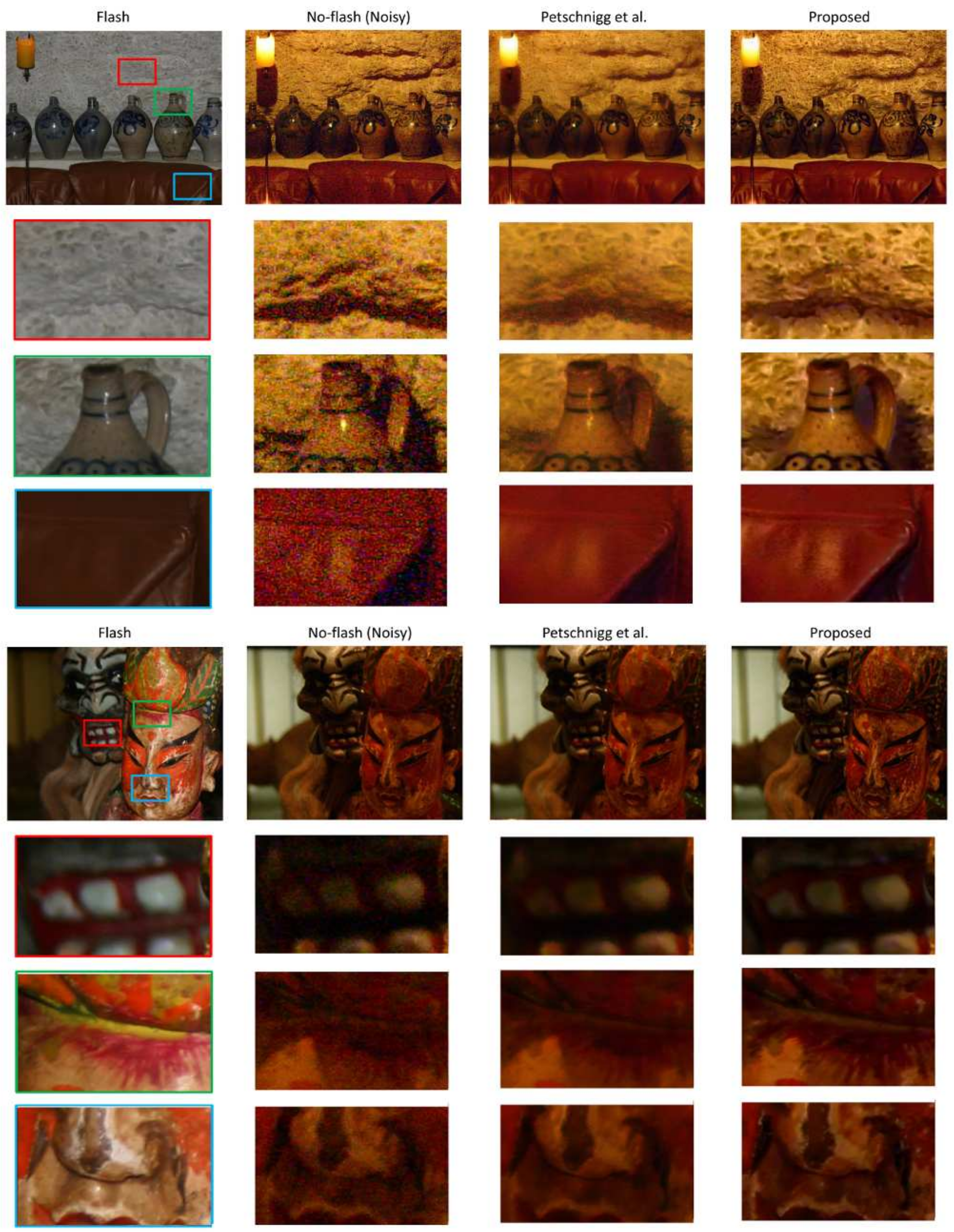


Figure 8. Flash/no-flash denoising examples compared to the state of the art method [12].

mance by applying it iteratively. Iterations of the proposed method consist of a combination of diffusion and residual iteration. We demonstrated that the proposed approach yields outputs that not only preserve fine details of the flash image, but also ambient lighting of the no-flash image. The proposed method outperforms state of the art methods in both flash/no-flash image denoising and deblurring. It is also worthwhile to explore several other applications such as joint upsampling [10], image matting [7], mesh smoothing [5, 9], and specular highlight removal [19] where the proposed method can be employed.

References

[1] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics*, 24:828–835, 2005. 1, 2

[2] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, 4(2):490–530, 2005. 3

[3] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 21(3):673–678, 2004. 1, 2, 3

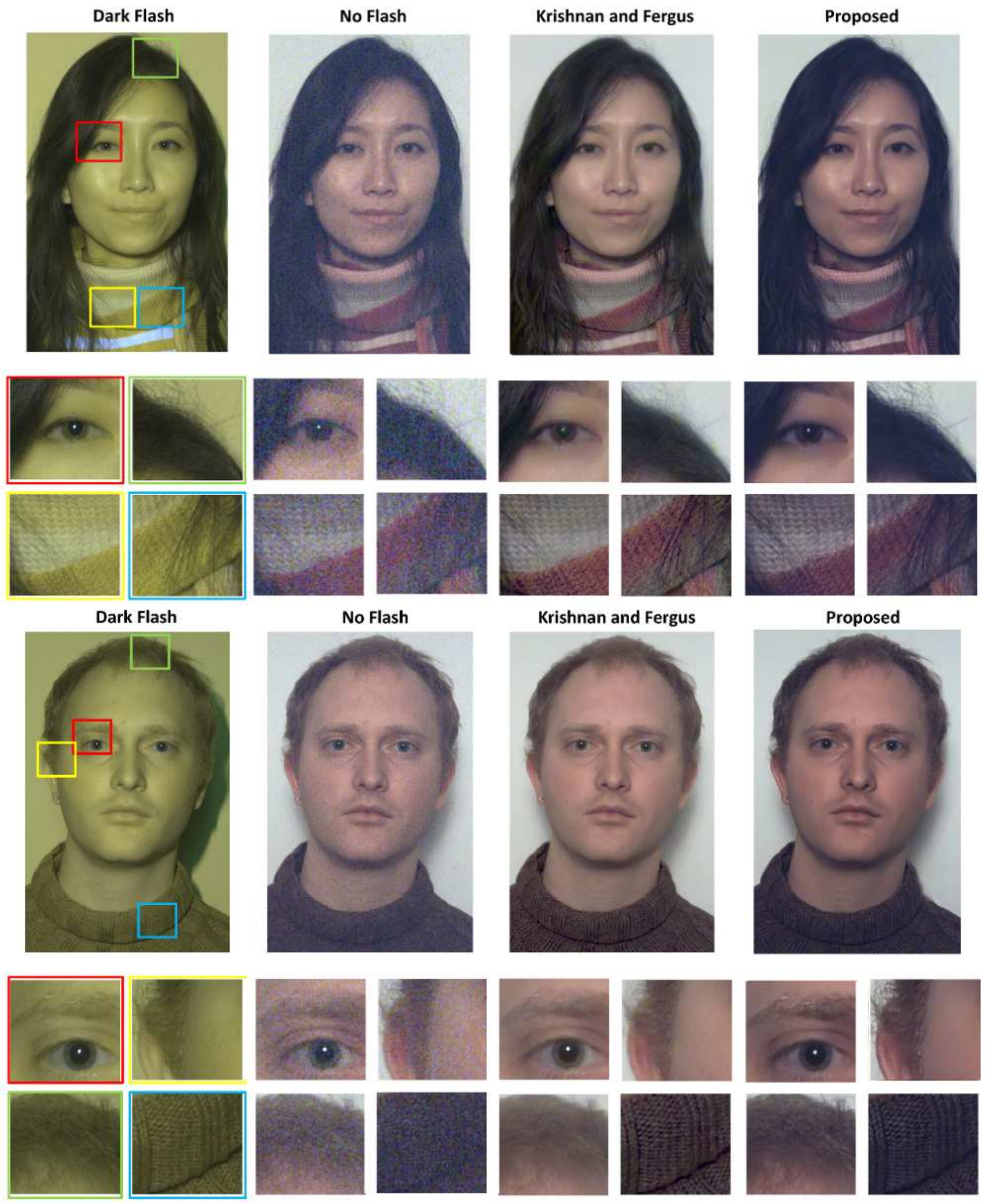


Figure 9. Dark flash/no-flash denoising examples compared to the state of the art method [11].

[4] R. Fergus, B. Singh, A. Hertsmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 2006. 1

[5] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3):950–953, 2003. 6

[6] W. Hasinoff. Variable-aperture photography. *PhD Thesis, University of Toronto, Dept. of Computer Science*, 2008. 2

[7] K. He, J. Sun, and X. Tang. Fast matting using large kernel matting Laplacian matrices. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 6

[8] K. He, J. Sun, and X. Tang. Guided image filtering. *In Proc. European Conference Computer Vision (ECCV)*, 2010. 1, 2, 4, 5

[9] T. Jones, F. Durand, and M. Desbrun. Non-iterative feature preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3):943–949, 2003. 6

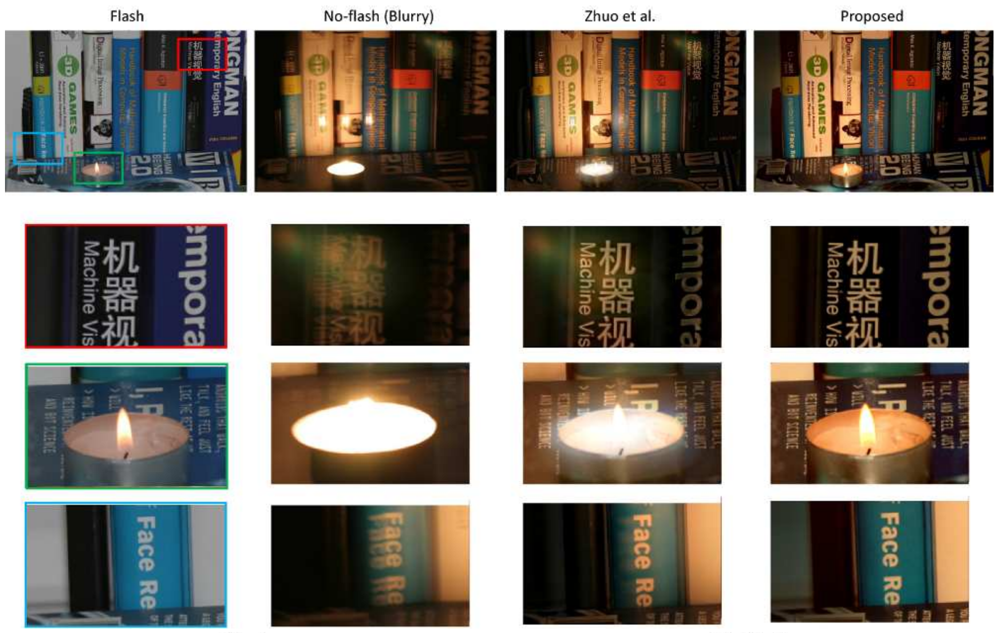
[10] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3), 2007. 6

[11] D. Krishnan. and R. Fergus. Dark flash photography. *ACM Transactions on Graphics*, 28, 2009. 1, 2, 5, 7

[12] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 21(3):664–672, 2004. 1, 2, 3, 5, 6

[13] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. *IEEE Trans. Image Processing*, 18(1):36–51, 2009. 4

[14] Q. Shan, J. Jia, and M. S. Brown. Globally optimized linear



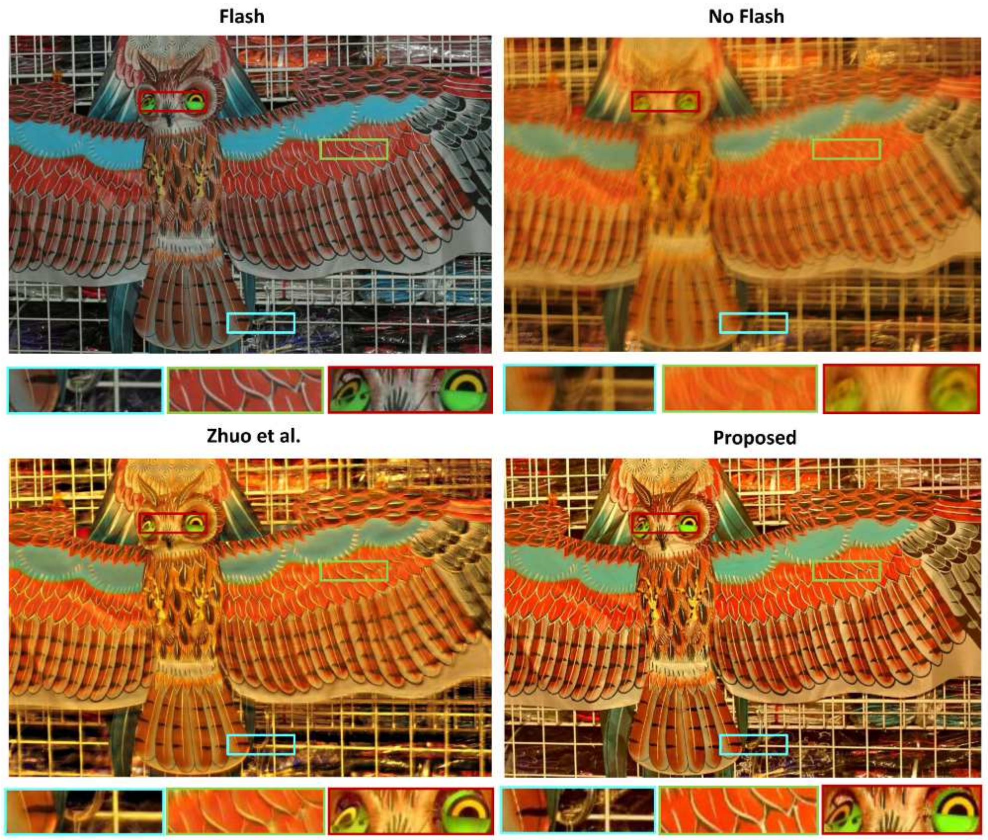


Figure 10. Flash/no-flash deblurring examples compared to the state of the art method [21].

windowed tone-mapping. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2010. 1

[15] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. *IEEE Conference on Computer Vison and Pattern Recognition*, 2005. 1

[16] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, February 2007. 3

[17] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceeding of the 1998 IEEE International Conference of Compute Vision, Bombay, India*, pages 836–

846, January 1998. 1, 3

[18] J. W. Tukey. *Exploratory Data Analysis*. Addison Wesley, 1977. 5

[19] Q. Yang, S. Wang, and N. Ahuja. Real-time specular highlight removal using bilateral filtering. In *ECCV*, 2010. 6

[20] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics (SIGGRAPH)*, 2007. 1

[21] S. Zhuo, D. Guo, and T. Sim. Robust flash deblurring. *IEEE Conference on Computer Vison and Pattern Recognition*, 2010. 1, 2, 5, 8