

# Cross-Field Joint Image Restoration via Scale Map

Qiong Yan<sup>§</sup>   Xiaoyong Shen<sup>§</sup>   Li Xu<sup>§</sup>   Shaojie Zhuo<sup>†</sup>  
Xiaopeng Zhang<sup>†</sup>   Liang Shen<sup>†</sup>   Jiaya Jia<sup>§</sup>

<sup>§</sup>The Chinese University of Hong Kong      <sup>†</sup>Qualcomm Incorporated

<http://www.cse.cuhk.edu.hk/leo/jia/projects/crossfield/>

## Abstract

*Color, infrared, and flash images captured in different fields can be employed to effectively eliminate noise and other visual artifacts. We propose a two-image restoration framework considering input images in different fields, for example, one noisy color image and one dark-flashed near-infrared image. The major issue in such a framework is to handle structure divergence and find commonly usable edges and smooth transition for visually compelling image reconstruction. We introduce a scale map as a competent representation to explicitly model derivative-level confidence and propose new functions and a numerical solver to effectively infer it following new structural observations. Our method is general and shows a principled way for cross-field restoration.*

## 1. Introduction

Images captured in dim light are hardly satisfactory. They could be very noisy when increasing ISO in a short exposure duration. Using flash might improve lighting; but it creates unwanted shadow and highlight, or changes tone of the image. The methods of [6, 14, 1] restore a color image based on flash and non-flash inputs of the same scene. Recently, because of the popularity of other imaging devices, more computational photography and computer vision solutions based on images captured under different configurations were developed.

For example, near infrared (NIR) images are with a single channel recording infrared light reflected from objects with spectrum ranging from 700nm-1000nm in wavelength. NIR images contain many similar structures as visible color ones when taken from the same camera position. This enables a configuration to take an NIR image with less noisy details by dark flash [11] to guide corresponding noisy color image restoration. The main advantage is on only using NIR flash invisible to naked human eyes, making

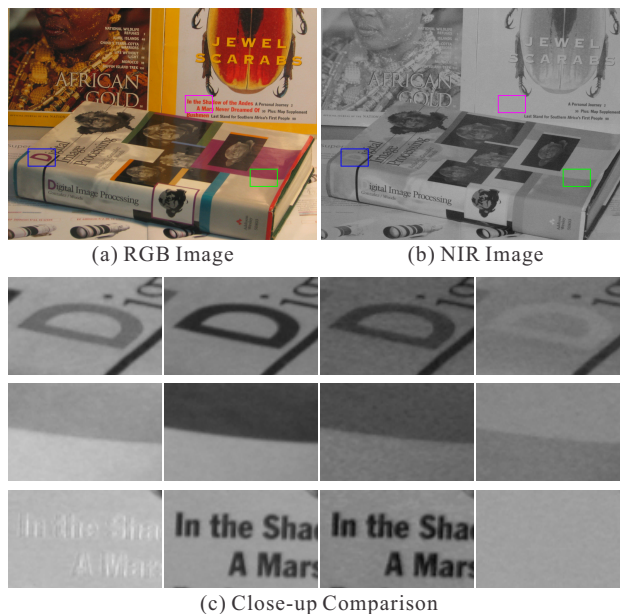


Figure 1. Appearance comparison of RGB and NIR images. (a) RGB image. (b) Corresponding NIR image. (c) Close-ups. The four columns are for the R, G, B, and NIR channels respectively.

it a suitable way for daily portrait photography and of remarkable practical importance.

In previous methods, Krishnan et al. [11] used gradients of a dark-flashed image, capturing ultraviolet (UV) and NIR light to guide noise removal in the color image. Considering rich details in NIR images, Zhang et al. [20] enhanced the RGB counterpart by transferring contrast and details via Haar wavelets. In [21] and [16], the detail layer was manipulated differently for RGB and haze image enhancement.

Several methods also explore other image fusion applications in two-image deblurring [19], matting [17], tone mapping [7], upsampling [10], context enhancement [15], relighting [2], to name a few. Bhat et al. [3] proposed GradientShop to edit gradients, which can also be used to enhance images.

We note existing methods work well for their respective applications by handling different detail layers or gradients from multiple images. But in terms of two-image high-quality restoration, there remain a few major and fundamental issues that were not sufficiently addressed. We take the RGB-NIR images shown in Fig. 1 as an example to reveal the noticeable difference in detail distribution and intensity formation. Structure inconsistency existing for many pixels can be categorized as follows.

- **Gradient Magnitude Variation.** In the first row of Fig. 1(c), letter “D” is with different contrast. It is due to varied reflectance to infrared and visible light.
- **Gradient Direction Divergence.** In the second row, edge gradients have opposite directions in the two images, which cause structural deviation.
- **Gradient Loss.** In the last row, the characters are completely lost in the NIR image.
- **Shadow and Highlight by Flash.** If one uses flash only for the NIR image, it inevitably generates highlight/shadow that is not contained in the other image. Examples are presented later.

These issues are caused by inherent discrepancy of structures in different types of images, which we call *cross-field* problems. The algorithms to address them can be generally referred to as *cross-field image restoration*. Simple joint image filtering [18, 8] could blur weak edges due to the inherent smoothing property. Directly transferring guidance gradients to the noisy field also results in unnatural appearance.

In this paper, we propose a framework via novel scale map construction. This map captures the nature of structure discrepancy between images and has clear statistical and numerical meanings. Based on its analysis, we design functions to form an optimal scale map considering adaptive smoothing, edge preservation, and guidance strength manipulation. Aforementioned cross-field issues are discussed and addressed in this framework. We also develop an effective solver via robust function approximation and problem decomposition, which converges in less than 5 passes compared to other gradient decent alternatives that may need tens or hundreds of iterations.

## 2. Modeling and Formulation

Our system takes the input of a noisy RGB image  $I_0$  and a guidance image  $G$  captured from the same camera position.  $G$  can be a dark-flashed NIR image or others with possible structure variation as discussed above. Other cross-field configurations are allowed in our framework, presented in Section 4. Pixel values in each channel are scaled to  $[0, 1]$ .  $G$  and  $I_0$  could have different number of channels. Our goal is to recover an image from  $I_0$  with



Figure 2. Optimal scale map  $s$  computed from images in Fig. 1 according to Eq. (1). Dark to bright pixels correspond to negative to positive values in different scales.

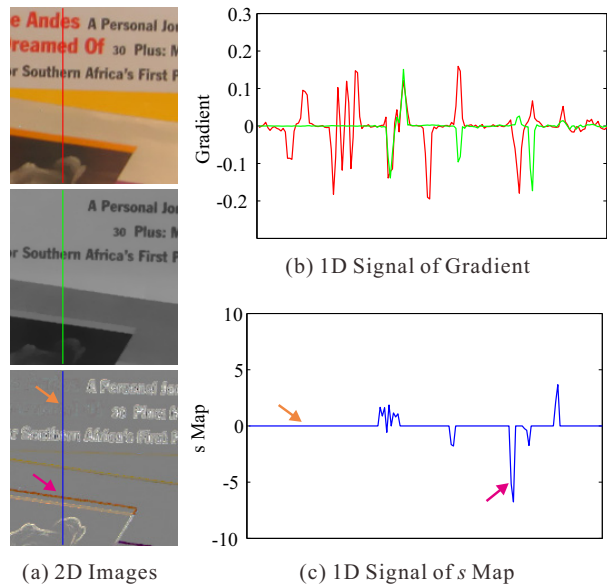


Figure 3. 1D illustration. (a) Patch in the color image, NIR image and  $s$  map. Plot (b) contains gradients along the vertical line in the top two patches. (c) shows corresponding  $s$  values. Most of them are zeros; positive and negative values also exist.

noise removed and structure retained. We process color channels separately.

We introduce an auxiliary map  $s$  with the same size as  $G$ , which is key to our method, to adapt structure of  $G$  to that of  $I^*$  – the ground truth noise-free image. The  $s$  map is defined under condition

$$\min \|\nabla I^* - s \cdot \nabla G\|. \quad (1)$$

Here  $\nabla$  is an operator forming a vector with  $x$ - and  $y$ -direction gradients. Each element  $s_i$  in map  $s$ , where  $i$  indexes pixels, is a scalar, measuring robust difference between corresponding gradients in the two images. Simply put,  $s$  is a ratio map between the guidance and latent images. The optimal  $s$  corresponding to the cross-field example in Fig. 1 is shown in Fig. 2, visualized as a color image after

pixel-wise value normalization to  $[0,1]$ .

We analyze the properties of  $s$  with regard to structure discrepancy between  $\nabla G$  and  $\nabla I^*$ , and present them as follows with the illustration in Fig. 3.

**Property of  $s$**  First, sign of each  $s_i$  can be either positive or negative. A negative  $s_i$  means edges exist in the two images, but with opposite directions, as demonstrated in Fig. 3(c). Second, when the guidance image  $G$  contains extra shadow and highlight caused by flash, which are absent in  $\nabla I^*$ ,  $s_i$  with value 0 can help ignore them. Finally,  $s_i$  can be any value when  $\nabla G_i = 0$  – that is, guidance edge does not exist, such as the red letters in Fig. 3(a). In this case, under local smoothness,  $s_i$  being 0 is a good choice.

In short, an optimal  $s$  map should be able to represent all these structure discrepancies. It is first-of-a-kind to avail cross-field restoration. Its additional benefit is the special role as latent variables to develop an efficient optimization procedure.

**More of the Function** We denote by  $I$  our estimate towards  $I^*$ . Eq. (1) is updated to

$$\min \|\nabla I - s \cdot \nabla G\|. \quad (2)$$

As it involves unknowns  $\nabla I$  and  $s$ , which correlate, the function is ill-posed. We take its variation as a data term expression, together with regularization on  $s$ , to construct an objective function.

### 2.1. Data Term about $s$

In  $|s_i \nabla G_i - \nabla I_i|$ , where  $i$  indexes pixels,  $\nabla G_i$  can be analogously regarded as a scale map for  $s_i$  due to the dual relation between  $s_i$  and  $\nabla G_i$ . It controls the penalty when computing  $s_i$  for different pixels. The final cost resulted from  $|s_i \nabla G_i - \nabla I_i|$  is dependent on the value of  $\nabla G_i$ . For example, if  $\nabla G_i$  and  $\nabla I_i$  are doubled simultaneously, although  $s$  remains the same, the cost from  $|s_i \nabla G_i - \nabla I_i|$  will get twice larger.

To stabilize costs w.r.t.  $s_i$ , we perform normalization

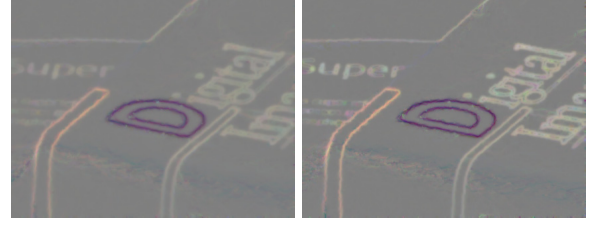
$$\sum_i \left| s_i - \frac{\nabla_x I_i}{\nabla_x G_i} \right| + \left| s_i - \frac{\nabla_y I_i}{\nabla_y G_i} \right|, \quad (3)$$

which is modulated by the two components of  $\nabla G_i$ . It removes the unexpected scaling effect caused by  $\nabla G_i$ . Further to avoid the extreme situation when  $\nabla_x G_i$  or  $\nabla_y G_i$  is close to zero, and enlist the ability to reject outliers, we define our data term as

$$E_1(s, I) = \sum_i \left( \rho(|s_i - p_{i,x} \nabla_x I_i|) + \rho(|s_i - p_{i,y} \nabla_y I_i|) \right), \quad (4)$$

where  $\rho$  is a robust function defined as

$$\rho(x) = |x|^\alpha, \quad 0 < \alpha < 1. \quad (5)$$



(a) Isotropic Smoothing (b) Anisotropic Smoothing

Figure 4. Isotropic versus anisotropic smoothing of the  $s$  map. Result in (b) from anisotropic smoothing contains higher contrast structure. The input images are shown in Fig. 5(a).

It is used to remove estimation outliers. We set  $\alpha = 0.9$  in experiments.  $p_{i,k}$ , where  $k \in \{x, y\}$ , is a truncation function

$$p_{i,k} = \frac{1}{\text{sign}(\nabla_k G_i) \cdot \max(|\nabla_k G_i|, \varepsilon)}, \quad (6)$$

where  $\text{sign}(x)$  is the sign operator, outputting 1 if  $\nabla_k G_i$  is positive or zero and outputting -1 otherwise.  $\max(|\nabla_k G_i|, \varepsilon)$  returns the larger value between  $|\nabla_k G_i|$  and  $\varepsilon$ . The threshold  $\varepsilon$  is used to avoid division by zero and is set to 0.004 empirically.

### 2.2. Data Term for $I$

The data term for  $I$  is simply set as

$$E_2(I) = \sum_i \rho(|I_i - I_{0,i}|), \quad (7)$$

where  $\rho$  is the same robust function and  $I_{0,i}$  is the color of pixel  $i$  in  $I_0$ .  $E_2(I)$  requires the restoration result not to wildly deviate from the input noisy image  $I_0$  especially along salient edges. The robust function  $\rho$  helps reject part of the noise from  $I_0$ .

### 2.3. Regularization Term

Our regularization term is defined with anisotropic gradient tensors [13, 4]. It is based on the fact that  $s$  values are similar locally only in certain directions. For instance,  $s$  values should change smoothly or be constant along an edge more than those across it. As shown in Fig. 4, uniformly smoothing  $s$  in all directions blurs sharp edges.

Our anisotropic tensor scheme preserves sharp edges according to gradient directions of  $G$ . By a few algebraic operations, an anisotropic tensor is expressed as

$$D(\nabla G_i) = \frac{1}{(\nabla G_i)^2 + 2\eta^2} ((\nabla G_i^\perp)(\nabla G_i^\perp)^T + \eta^2 \mathbf{1}), \quad (8)$$

where  $\nabla G_i^\perp = (\nabla_y G_i, -\nabla_x G_i)^T$  is a vector perpendicular to  $\nabla G_i$ ,  $\mathbf{1}$  is an identity matrix and scalar  $\eta$  controls the isotropic smoothness. When  $\nabla G_i$  is much smaller than

$\eta$ , Eq. (8) degrades to  $0.5 \cdot \mathbf{1}$  and the structure tensor is therefore isotropic.

Generally, the two orthogonal eigenvectors of  $D(\nabla G_i)$  are

$$\mathbf{v}_{i,1} = \frac{\nabla G_i}{|\nabla G_i|}, \quad \mathbf{v}_{i,2} = \frac{\nabla G_i^\perp}{|\nabla G_i|}, \quad (9)$$

with corresponding eigenvalues

$$\mu_{i,1} = \frac{\eta^2}{(\nabla G_i)^2 + 2\eta^2}, \quad \mu_{i,2} = \frac{(\nabla G_i)^\perp{}^2 + \eta^2}{(\nabla G_i)^\perp{}^2 + 2\eta^2}. \quad (10)$$

This decomposes the tensor to

$$D(\nabla G_i) = (\mathbf{v}_{i,1} \quad \mathbf{v}_{i,2}) \begin{pmatrix} \mu_{i,1} & 0 \\ 0 & \mu_{i,2} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{i,1}^T \\ \mathbf{v}_{i,2}^T \end{pmatrix}. \quad (11)$$

This form makes it possible to express regularization for each  $\nabla s_i$  as

$$E_3(\nabla s_i) = \mu_{i,1}(\mathbf{v}_{i,1}^T \nabla s_i)^2 + \mu_{i,2}(\mathbf{v}_{i,2}^T \nabla s_i)^2. \quad (12)$$

Different smoothing penalties are controlled by  $\mu_{i,1}$  and  $\mu_{i,2}$  in directions  $\mathbf{v}_{i,1}$  and  $\mathbf{v}_{i,2}$ , across and along edges respectively. Stronger smoothness is naturally imposed along edges. The final smoothing term is thus defined as

$$E_3(\nabla s) = \sum_i (\mu_{i,1}(\mathbf{v}_{i,1}^T \nabla s_i)^2 + \mu_{i,2}(\mathbf{v}_{i,2}^T \nabla s_i)^2). \quad (13)$$

## 2.4. Final Objective Function

The final objective function to estimate the  $s$  map and restore image  $I$  is written as

$$E(s, I) = E_1(s, I) + \lambda E_2(I) + \beta E_3(\nabla s), \quad (14)$$

where  $\lambda$  controls the confidence on noisy image  $I_0$ , and  $\beta$  corresponds to smoothness of  $s$ . We describe their setting in Section 4.

This objective function is non-convex due to the involvement of sparsity terms. Joint representation for  $s$  and  $I$  in optimization further complicates it. Naive gradient decent cannot guarantee optimality and leads to very slow convergence even for a local minimum. We contrarily propose an iterative method, which finds constraints to shape the  $s$  map according to its characteristics and yields the effect to remove intensive noise from input  $I_0$ .

## 3. Numerical Solution

To solve the non-convex function  $E(s, I)$  defined in Eq. (14), we employ the iterative reweighted least squares (IRLS), which make it possible to convert the original problem to a few corresponding linear systems without losing generality. This process, however, is still nontrivial and needs a few derivations.

Initially, robust function  $\rho(x)$  in Eq. (5) for any scalar  $x$  can be written as  $x^2/|x|^{2-\alpha}$ , further approximated as

$$\rho(x) \approx \phi(x) \cdot x^2, \quad (15)$$

where  $\phi(x)$  is defined as

$$\phi(x) = \frac{1}{|x|^{2-\alpha} + \epsilon}. \quad (16)$$

$\epsilon$  is a small number to avoid division by 0. We set it to  $1E-4$  empirically. This form splits the robust function into two parts where  $\phi(x)$  can be regarded as a weight for  $x^2$ . In our method, following the tradition of IRLS,  $\phi(x)$  and  $x^2$  are updated alternatively during optimization because each of them can work together with other necessary terms to form simpler representations, profiting optimization.

**Vector Form** To ease derivation, we re-write Eq. (14) in the vector form by taking the expression in Eq. (15) into computation. It yields

$$\begin{aligned} E(\mathbf{s}, \mathbf{I}) = & (\mathbf{s} - P_x C_x \mathbf{I})^T A_x (\mathbf{s} - P_x C_x \mathbf{I}) \\ & + (\mathbf{s} - P_y C_y \mathbf{I})^T A_y (\mathbf{s} - P_y C_y \mathbf{I}) \\ & + \lambda (\mathbf{I} - \mathbf{I}_0)^T B (\mathbf{I} - \mathbf{I}_0) + \beta \mathbf{s}^T L \mathbf{s}, \end{aligned} \quad (17)$$

where  $\mathbf{s}$ ,  $\mathbf{I}$ , and  $\mathbf{I}_0$  are vector representations of  $s$ ,  $I$ , and  $I_0$ .  $C_x$  and  $C_y$  are discrete backward difference matrices that are used to compute image gradients in the  $x$ - and  $y$ -directions.  $P_x$ ,  $P_y$ ,  $A_x$ ,  $A_y$  and  $B$  are diagonal matrices, whose  $i$ -th diagonal elements are defined as

$$\begin{aligned} (P_x)_{ii} &= p_{i,x}, & (A_x)_{ii} &= \phi(s_i - p_{i,x} \nabla_x I_i), \\ (P_y)_{ii} &= p_{i,y}, & (A_y)_{ii} &= \phi(s_i - p_{i,y} \nabla_y I_i), \\ B_{ii} &= \phi(I_i - I_{0,i}). \end{aligned}$$

Among them,  $A_x$ ,  $A_y$  and  $B$  account for the re-weighting process and are typically computed using estimates from previous iterations –  $P_x$  and  $P_y$  are normalization terms from the guidance image. The first three terms in Eq. (17) correspond to terms  $E_1$  and  $E_2$ ;  $\mathbf{s}^T L \mathbf{s}$  is created by  $E_3$ .

Note the last term  $\mathbf{s}^T L \mathbf{s}$  controls spatial smoothness of  $\mathbf{s}$ , where matrix  $L$  is a smoothing Laplacian, expressed as

$$\begin{aligned} L = & C_x^T (\Sigma_1 V_x^2 + \Sigma_2 V_y^2) C_x \\ & + C_y^T (\Sigma_2 V_x^2 + \Sigma_1 V_y^2) C_y + 2C_y^T (\Sigma_1 - \Sigma_2) V_x V_y C_x \end{aligned} \quad (18)$$

after a bit complicated derivations.  $\Sigma_1$ ,  $\Sigma_2$ ,  $V_x$ , and  $V_y$  are all diagonal matrices. Their  $i$ -th diagonal elements are

$$\begin{aligned} (\Sigma_1)_{ii} &= \mu_{i,1}, & (V_x)_{ii} &= \nabla_x G_i / \max(|\nabla G_i|, \epsilon), \\ (\Sigma_2)_{ii} &= \mu_{i,2}, & (V_y)_{ii} &= \nabla_y G_i / \max(|\nabla G_i|, \epsilon). \end{aligned}$$



---

**Algorithm 1** Cross-Field Image Restoration.

---

- 1: **input:** noisy image  $I_0$ , guidance image  $G$ , parameters  $\beta$  and  $\lambda$
  - 2: initialize  $I \leftarrow I_0$ ,  $s \leftarrow 1$
  - 3: **repeat**
  - 4:   estimate  $s$  according to Eq. (21)
  - 5:   estimate  $I$  according to Eq. (23)
  - 6: **until** convergence
  - 7: **output:**  $s$  map and restored image  $I$
- 

**Analysis** We note  $L$  is actually an inhomogeneous term, reflecting the anisotropic property of our smoothing regularizer. To understand it, consider the extreme case that  $\nabla G$  approaches zero. It leads to  $\Sigma_1 = \Sigma_2$  and  $V_x = V_y = 0$ , making  $L$  a homogenous Laplacian. The resulting  $s$  map is therefore smooth in all directions. But in natural images,  $\nabla G$  on an edge is not isotropic and should be with nonuniform regularization strength. Also, sparse  $C_x$  and  $C_y$  lead to the sparse Laplacian matrix  $L$ , which facilitates optimization because many mature sparse-matrix solvers exist in this community already.

### 3.1. Solver

We solve for  $\mathbf{s}$  and  $\mathbf{I}$  based on above derivations. Results of  $\mathbf{s}$  and  $\mathbf{I}$  in each iteration  $t$  are denoted as  $\mathbf{s}^{(t)}$  and  $\mathbf{I}^{(t)}$ . Initially, we set  $\mathbf{s}^{(0)} = \mathbf{1}$ , whose elements are all 1s and  $\mathbf{I}^{(0)} = \mathbf{I}_0$ .

By setting all initial  $s_i$  to 1s, total smoothness is obtained. It yields zero cost for  $E_3(s)$ , a nice starting point for optimization. This initialization also makes the starting  $\nabla I$  same as  $\nabla G$  with many details. Then at iteration  $t + 1$ , we solve two subproblems alternatively

- Given  $\mathbf{s}^{(t)}$  and  $\mathbf{I}^{(t)}$ , minimize  $E(\mathbf{s}, \mathbf{I}^{(t)})$  to get  $\mathbf{s}^{(t+1)}$ .
- Given  $\mathbf{s}^{(t+1)}$  and  $\mathbf{I}^{(t)}$ , minimize  $E(\mathbf{s}^{(t+1)}, \mathbf{I})$  to update  $\mathbf{I}^{(t+1)}$ .

The procedure is repeated until  $\mathbf{s}$  and  $\mathbf{I}$  do not change too much. Usually, 4-6 iterations are enough to generate visually compelling results. The algorithm is depicted in Algorithm 1, with the solvers elaborated on as follows.

**Solve for  $\mathbf{s}^{(t+1)}$**  The energy function with respect to  $\mathbf{s}$  can be expressed as

$$E(\mathbf{s}) = (\mathbf{s} - P_x C_x \mathbf{I})^T A_x (\mathbf{s} - P_x C_x \mathbf{I}) + (\mathbf{s} - P_y C_y \mathbf{I})^T A_y (\mathbf{s} - P_y C_y \mathbf{I}) + \beta \mathbf{s}^T L \mathbf{s}. \quad (19)$$

Computation of  $A_x$  and  $A_y$  depends on estimates  $\mathbf{s}$  and  $\mathbf{I}$  from the previous iteration. We denote by  $A_x^{t,t}$  and  $A_y^{t,t}$  the

matrices computed with  $\mathbf{s}^{(t)}$  and  $\mathbf{I}^{(t)}$ , which lead to

$$\begin{aligned} \tilde{E}(\mathbf{s}) &= (\mathbf{s} - P_x C_x \mathbf{I}^{(t)})^T A_x^{t,t} (\mathbf{s} - P_x C_x \mathbf{I}^{(t)}) \\ &\quad + (\mathbf{s} - P_y C_y \mathbf{I}^{(t)})^T A_y^{t,t} (\mathbf{s} - P_y C_y \mathbf{I}^{(t)}) + \beta \mathbf{s}^T L \mathbf{s}. \end{aligned} \quad (20)$$

It is simply quadratic. Taking derivatives on  $\mathbf{s}$  and setting them to 0s, we obtain the sparse linear system

$$(A_x^{t,t} + A_y^{t,t} + \beta L) \mathbf{s} = A_x^{t,t} P_x C_x \mathbf{I}^{(t)} + A_y^{t,t} P_y C_y \mathbf{I}^{(t)}. \quad (21)$$

We solved it using pre-conditioned conjugate gradient (PCG). The solution is denoted as  $\mathbf{s}^{(t+1)}$ .

**Solve for  $\mathbf{I}^{(t+1)}$**  Similarly, the energy function to solve for  $\mathbf{I}$  is given by

$$\begin{aligned} \tilde{E}(\mathbf{I}) &= (\mathbf{s}^{(t+1)} - P_x C_x \mathbf{I})^T A_x^{t+1,t} (\mathbf{s}^{(t+1)} - P_x C_x \mathbf{I}) \\ &\quad + (\mathbf{s}^{(t+1)} - P_y C_y \mathbf{I})^T A_y^{t+1,t} (\mathbf{s}^{(t+1)} - P_y C_y \mathbf{I}) \\ &\quad + \lambda (\mathbf{I} - \mathbf{I}_0)^T B^{t+1,t} (\mathbf{I} - \mathbf{I}_0), \end{aligned} \quad (22)$$

where  $A_x^{t+1,t}$  and  $A_y^{t+1,t}$  are calculated with available  $\mathbf{s}^{(t+1)}$  and  $\mathbf{I}^{(t)}$ .  $B^{t+1,t}$  depends on  $\mathbf{I}^{(t)}$ . The final linear system in the matrix form is

$$\begin{aligned} &((C_x^T (P_x)^2 A_x^{t+1,t} C_x + C_y^T (P_y)^2 A_y^{t+1,t} C_y) + \lambda B^{t+1,t}) \mathbf{I} \\ &= (C_x^T P_x A_x^{t+1,t} + C_y^T P_y A_y^{t+1,t}) \mathbf{s} + \lambda B^{t+1,t} \mathbf{I}_0. \end{aligned} \quad (23)$$

The linear system is also solved using PCG and the solution is denoted as  $\mathbf{I}^{(t+1)}$ .

### 3.2. Why Does It Work?

According to the linear system defined in Eq. (21), the resulting  $s_i$  for pixel  $i$  is a weighted average of  $p_{i,x} \nabla_x I_i \approx \nabla_x I_i / \nabla_x G_i$  and  $p_{i,y} \nabla_y I_i \approx \nabla_y I_i / \nabla_y G_i$ , whose weights are determined by  $(A_x)_{ii}$  and  $(A_y)_{ii}$ . Even if these weights are quite different due to noise or other aforementioned issues described in Section 1, our method can still get a reasonable solution. We explain why this happens.

Assuming  $p_{i,x} \nabla_x I_i$  is larger than the other term, in solving for  $I$  according to Eq. (23),  $s_i$  reduces the gradient in the  $x$ -direction and increases the other so that  $\nabla I_i$  lies close to  $s \nabla G_i$ . In the meantime, noise is reduced. Then after each iteration, a less noisy  $I$  is put into Eq. (21) to produce new  $p_{i,x} \nabla_x I_i$  and  $p_{i,y} \nabla_y I_i$ , which are closer than those in previous iterations.

Eventually when the two estimates meet each other,  $s$  converges;  $I$  is accordingly optimal. The smoothness term  $L$  in Eq. (21) helps avoid discontinuity in the  $s$  map along edges of  $G$ .

We show in Fig. 5(e) the initial constant  $s$  map. (f)-(g) are maps produced in two iterations, and (h) shows the final  $s$ . Initially the map is noisy because of confusing or contradictive gradient magnitudes and directions in the

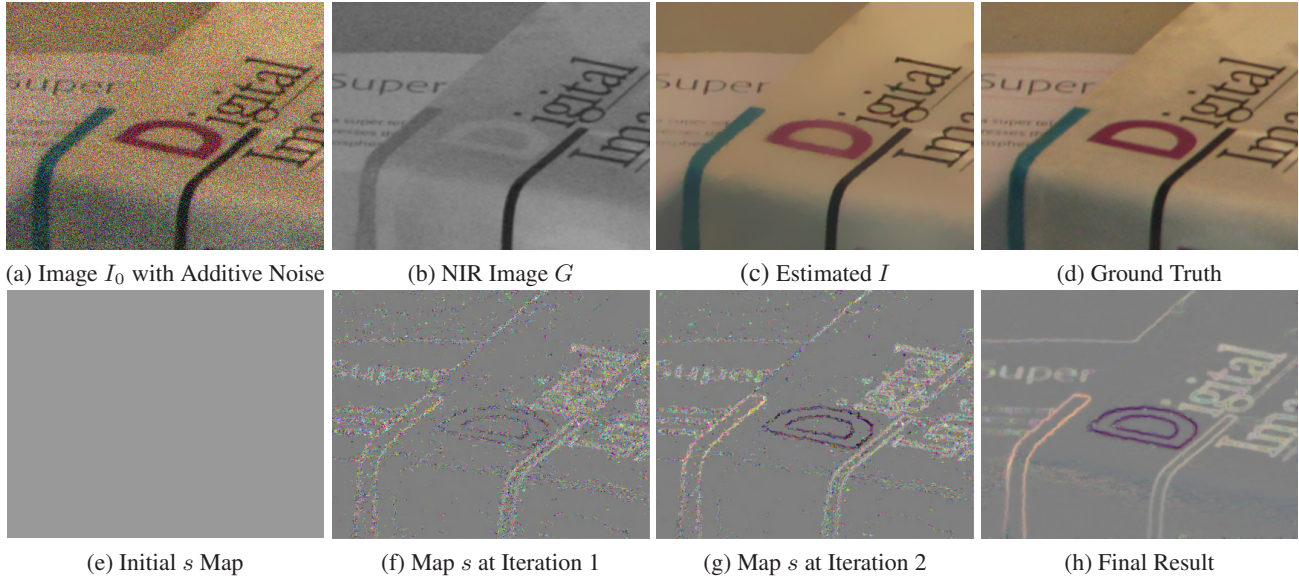


Figure 5.  $s$  map estimation in iterations. Given image pairs in (a) and (b), our method can get the high-quality restoration result in (c). The  $s$  maps in different iterations are shown in (e)-(h).

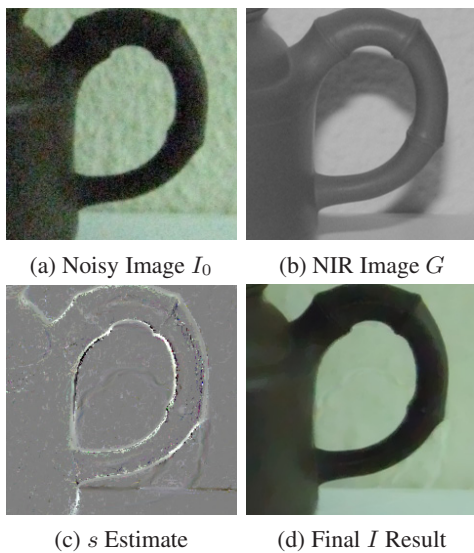


Figure 6. Handling shadow and highlight only existing in the guidance image  $G$ . Our  $s$  map automatically suppresses them.

input images. As of more iterations being taken, it becomes better regularized while not overly smoothed. Our final scale map adapts the gradients of  $G$  to match  $I_0$  with noise removed.

## 4. Experiments

Suppose the two input images – one is noisy and the other is clean – are aligned. They can be a RGB/NIR pair, flash/non-flash ones, or other cross-field images. We explain our algorithm on noisy RGB and flashed NIR images due to its generality of structure discrepancy.

**Experiment Setting and Running Time** Our method has two parameters  $\beta$  and  $\lambda$ , controlling smoothness of  $s$  and confidence of the noisy input.  $\lambda$  ranges from 1 – 10 and the value of  $\beta$  is in  $[0.2, 0.8]$ . We implement our method in MATLAB. Current un-optimized implementation takes about 30 seconds to process an image of size  $800 \times 600$ .

**Color and Flashed NIR Image Restoration** In the example shown in Fig. 5, some gradients of guidance NIR image are reversed or weak compared to the noisy color image. Reversed gradients for the letter “D” are corrected with the negative values in the resulting scale map  $s$ .

In Fig. 6, we show another example with highlight and shadow only in the flashed NIR image. Our estimated  $s$  map shown in (c) contains large values along object boundaries, and has close-to-zero values for highlight and shadow. The restoration result shown in (d) is with much less highlight and shadow, which is impossible to achieve by gradient transfer or joint filtering. Fig. 7 gives comparisons with BM3D [5] and the method of [21], which do not handle gradient variation. The difference is clear.

We also compare our result with the one presented in [11], which was generated by taking both UV and IR flashed image as guidance. Our method, by only taking the IR flashed image as  $G$ , accomplishes the comparable result shown in Fig. 8.

**Flash and Non-Flash Images** Our method is applicable to image restoration using flash/non-flash image pairs. Since the two input images are color ones under visible light, we use each channel from the flash image to guide image restoration in the corresponding channel of the non-flash noisy image. A result with comparison to that of [14]





Figure 7. Tea-bag example. Whole images are included in our project website.

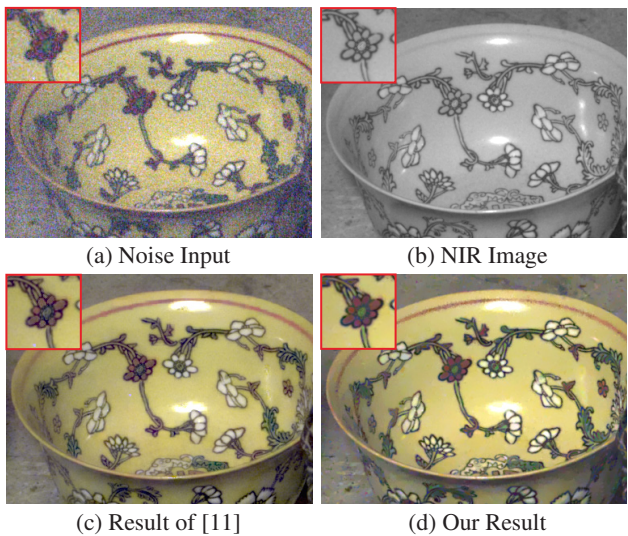


Figure 8. Comparison with [11]. Whole images are available in our project website.

is presented in Fig. 9. Overall, our recovered structures are sharp. Gradient reversion in input images also happens in this configuration due to strong flash. Without handling it, it is hard to preserve these sharp edges as gradients averaging

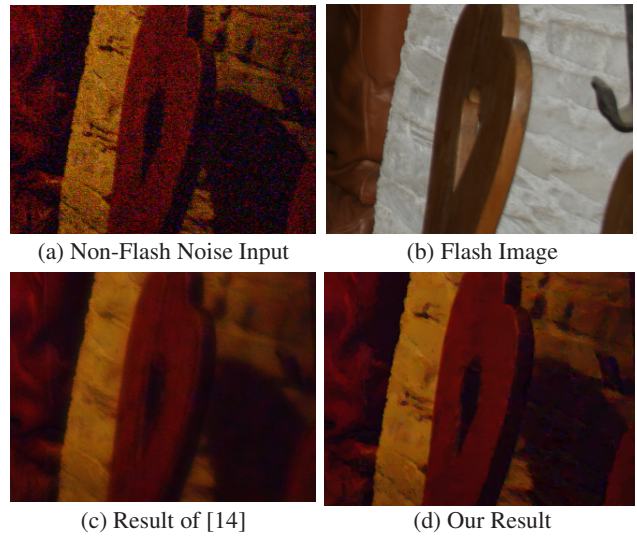


Figure 9. Image restoration from flash/non-flash image pairs. Complete results are available in our project website.

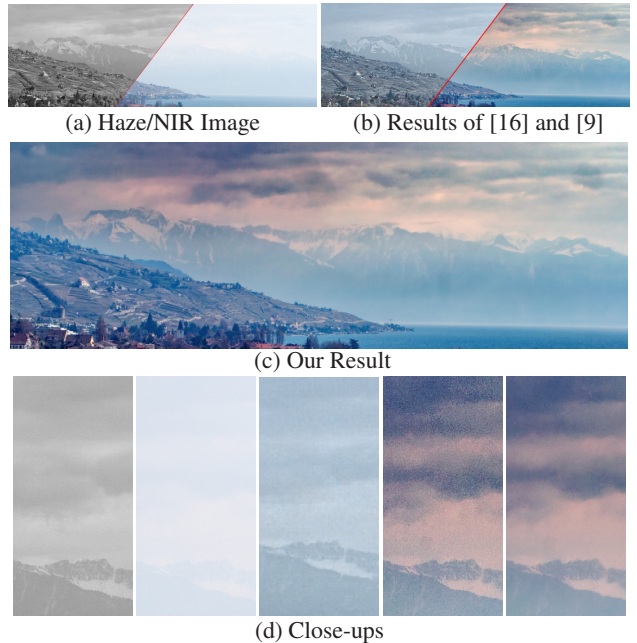


Figure 10. Image restoration from haze images. Close-ups shown in (d) are cut from (a-c). The left two are input NIR and haze images. The right three patches are results of [16], [9], and our method.

to nearly zeros are commonly resulted in.

**Other Cross-Field Restoration Examples** Our method finds many other applications. We apply it to cross-field dehazing with color and NIR images captured in haze. An image recovered from low visibility caused by haze could suffer from noise and compression artifacts due to significant gradient enhancement in low contrast regions. The NIR correspondence however is a hardware solution to

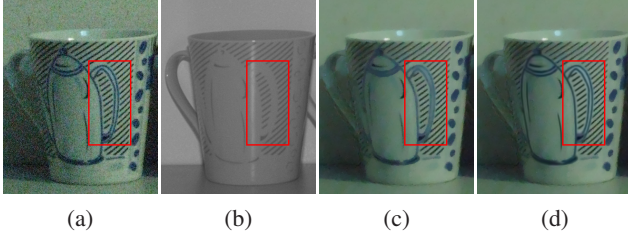


Figure 11. Limitation. There is no guidance structure in the rectangle of (b), making restoration less-constrained. Our two-image restoration result is shown in (c). This region can be further single-image denoised, as shown in (d).

see more details in haze. By applying our method to single-image dehazing result that is noisy and the NIR input, we can improve the quality. An example is shown in Fig. 10. The single-image dehazing result of [9] contains noise, and the result of [16], differently, changes the tone. Our restoration result with an NIR image as guidance  $G$  is more visually pleasing.

More results from our system are available in the project website (see the title page), including those of depth image enhancement using Kinect.

## 5. Conclusion and Limitation

We have presented a system effective for cross-field joint image restoration. Unlike transferring details or applying joint filtering, we explicitly take the possible structural discrepancy between input images into consideration. It is encoded in a scale map  $s$  that can represent all challenging cases. Our objective functions and optimization make good use of the guidance from other domains and preserve necessary details and edges.

The limitation of our current method is on the situation that the guidance does not exist, corresponding to zero  $\nabla G$  and non-zero  $\nabla I^*$  pixels. One example is shown in Fig. 11. Because the guidance does not exist, image restoration naturally degrades to single-image denoising.

## Acknowledgements

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 412911).

## References

[1] A. K. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.*, 24(3):828–835, 2005.

[2] D. Akers, F. Losasso, J. Klingner, M. Agrawala, J. Rick, and P. Hanrahan. Conveying shape and features with image-based relighting. In *IEEE Visualization*, pages 349–354, 2003.

[3] P. Bhat, C. L. Zitnick, M. F. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.*, 29(2), 2010.

[4] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, 1998.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[6] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, 2004.

[7] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Trans. Graph.*, 21(3):249–256, July 2002.

[8] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV (1)*, pages 1–14, 2010.

[9] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, 2011.

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

[11] D. Krishnan and R. Fergus. Dark flash photography. *ACM Trans. Graph.*, 28(3), 2009.

[12] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. In *ECCV (1)*, pages 602–613, 2004.

[13] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.

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

[15] R. Raskar, A. Ilie, and J. Yu. Image fusion for context enhancement and video surrealism. In *NPAR*, pages 85–152, 2004.

[16] L. Schaul, C. Fredembach, and S. Süsstrunk. Color image dehazing using the near-infrared. In *ICIP*, pages 1629–1632, 2009.

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

[18] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[19] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3):1, 2007.

[20] X. Zhang, T. Sim, and X. Miao. Enhancing photographs with near infra-red images. In *CVPR*, 2008.

[21] S. Zhuo, X. Zhang, X. Miao, and T. Sim. Enhancing low light images using near infrared flash images. In *ICIP*, pages 2537–2540, 2010.