High Resolution Photography with an RGB-Infrared Camera

Huixuan Tang[1] Xiaopeng Zhang[2] Shaojie Zhuo[2] Feng Chen[2] Kiriakos N. Kutulakos[1] Liang Shen[2]
[1] University of Toronto [2] Qualcomm Canada Inc.

Abstract

A convenient solution to RGB-Infrared photography is to extend the basic RGB mosaic with a fourth filter type with high transmittance in the near-infrared band. Unfortunately, applying conventional demosaicing algorithms to RGB-IR sensors is not possible for two reasons. First, the RGB and near-infrared image are differently focused due to different refractive indices of each band. Second, manufacturing constraints introduce crosstalk between RGB and IR channels. In this paper we propose a novel image formation model for RGB-IR cameras that can be easily calibrated, and propose an efficient algorithm that jointly addresses three restoration problems—channel deblurring, channel separation and pixel demosaicing—using quadratic image regularizers. We also extend our algorithm to handle more general regularizers and pixel saturation. Experiments show that our method produces sharp, full-resolution images of pure RGB color and IR.

1. Introduction

The last few years have seen a wealth of new camera and sensor technologies, with consumer-level photography being a major driver of these efforts. An important development in this direction are sensors that record short-wave infrared (IR) and color (RGB) in one shot. Although infrared imaging has a long history in remote sensing and the physical and biological sciences [25], recent work in computational photography [8, 21, 22] suggests it has great untapped potential in consumer imaging applications—from flash photography and reduced-blur imaging to 3D sensing and biometrics.

The conventional approach to joint infrared and color imaging is either to swap color filters on a camera sensitive to infrared or to use one camera dedicated to infrared imaging and another one for color. Unfortunately, taking sequential shots after swapping filters is problematic when imaging moving subjects, and using two cameras raises a host of problems of its own (e.g., higher cost, harder to miniaturize, misaligned infrared and color images due to differences in camera viewpoint).

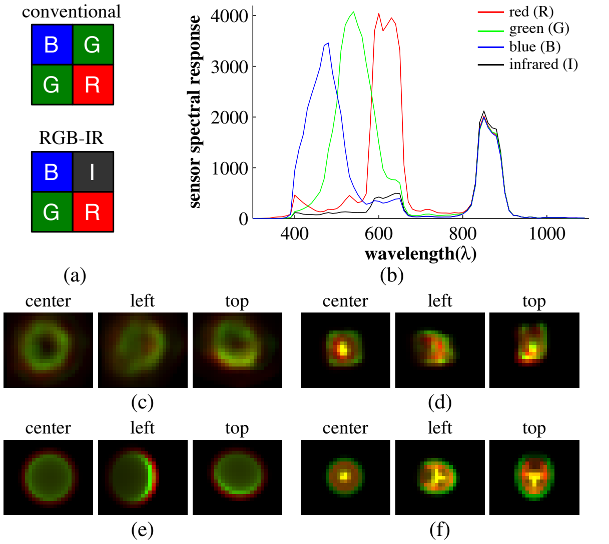

Figure 1: Pixel multiplexing, channel crosstalk and chromatic aberration of our prototype RGB-IR camera. **(a)** Color filter array. **(b)** Spectral response curves. **(c), (d)** Defocus kernel at three spatial locations for depths 44 cm and 18cm away, respectively, with the lens focused at 20cm. In each case, we overlay two kernels: a combined RGB kernel (green) and an IR kernel (red). **(e), (f)** Result of fitting the PSF model of Tang and Kutulakos [23], and evaluating it at the same locations as in (c), (d).

Cameras utilizing RGB-IR sensors, on the other hand, can be thought of as two cameras in one: they record both photos on the same sensor with the same lens and thus can be readily used in cellphone systems with conventional optics, producing perfectly-aligned RGB and IR images. As shown in Figure 1a, RGB-IR sensors extend the basic RGB mosaic with a fourth filter type whose transmittance is high in the short-wave IR band (800-950nm).

Producing a full-resolution RGB-IR photo from the output of such a sensor requires inferring the three missing channels at every pixel. Although individual solutions to this color demosaicing problem for conventional cameras [14] differ in many respects, they all rely on the fact that the spectral power distribution of light arriving at nearby pixels is often highly correlated. These correlations make it possible to infer a pixel's missing color channels from nearby pixels where those channels are sensed directly.

Unfortunately, applying conventional demosaicing algorithms to data from an RGB-IR sensor is not possible for two reasons.

First, manufacturing constraints cause the R, G and B filters of a color mosaic to transmit in the IR band. Although this is easily corrected in conventional RGB cameras by placing an IR-blocking filter in the optical path, this cannot be done for RGB-IR cameras. These cameras must therefore contend with the fact that their R, G, and B pixels actually record a superposition of the IR channel and a pure R, G, and B channel, respectively (Figure 1b). Because the RGB filters have high IR transmittance, they are also more prone to saturation and noise, reducing dynamic range.

Second, it is not always possible for both the RGB and the IR image of a subject to be focused on the sensor at the same time, since light refracts differently depending on wavelength. As a result, a major challenge in capturing high-quality photos with an RGB-IR sensor is computational correction of the inevitable defocus blur. This blur may be present either in the color component or the infrared component, or potentially in both (Figure 1c-f). In practice, this means that for subjects closer than the hyperfocal distance [20], adjusting the lens to bring the RGB channels into focus will cause the IR channel to be defocused and vice versa. Naive demultiplexing of RGB-IR data under such conditions yields poor-quality photos in which blur from one channel is transferred to the others. This dependence is especially problematic in high-resolution RGB-IR cameras, where the IR band is far from the other three and where even tiny levels of defocus blur will be captured by the sensor.

Thus, high-quality RGB-IR photography requires simultaneous solution of three basic computational problems: (1) *channel separation*, whereby the sensor's measurements are decomposed into pure R, G, B and IR channels, (2) *pixel demultiplexing*, where the value of these channels is estimated at every sensor pixel, and (3) *channel deblurring*, where wavelength-dependent defocus blur in the IR and/or RGB channels is removed. We study their joint solution from both a theoretical and an algorithmic perspective.

We propose a novel image formation model for RGB-IR cameras and then simplify it so it can be readily used with contemporary restoration algorithms [4]. As a side benefit, our model enables efficient restoration as well. Specifically, we show that, in the absence of saturation, our model reduces to a set of 4×16 linear constraints in the Fourier domain and thus can be directly inverted under quadratic regularization. Moreover, although iterations are necessary to handle non-quadratic image regularizers and pixel saturation, this formulation still enables efficient implementation of the individual iterations. We finally extend our restoration method to handle pixel saturation, which often occurs to RGB-IR cameras.

1.1. Related work

Individually, signal separation [3], demultiplexing [7, 13] and deblurring [11] have received considerable attention in the computational photography and image processing communities. We are not aware, however, of solutions applicable to RGB-IR cameras, where all three problems must be understood and solved jointly.

Specifically, recent work in NIR imaging has proposed color filter arrays for one-shot RGB and IR capture but either does not consider image restoration [12], or studies demosaicing or crosstalk in isolation [6, 9]. Although [18] addresses both crosstalk and demosaicing, it assumes crosstalk between the green and IR channels only. Moreover, these methods do not account for channel-dependent defocus blur and do not use Fourier-domain analysis either, which can offer significant computational efficiencies.

Our restoration algorithm is closely related to recent work on convex optimization for image restoration [2, 4]. In particular, our work can be thought of as an extension of Heide *et al.*'s optimization framework [4], designed to exploit the special properties of image formation in cameras with RGB-IR mosaics.

2. Image formation model

We start by modeling the formation of the input RGB-IR image $\mathbf{i}$ from the ideal (*i.e.*, sharp and full-resolution) images of the scene $\mathbf{l}_\lambda$ at various wavelengths λ. We then simplify it further by assuming the sensor's spectral responses are approximately the superposition of four narrow-band response curves. This simplified model can be readily inverted using existing restoration algorithms.

Continuous image formation model Let us first consider the unsaturated version of an input raw image $\mathbf{j}$. Representing all images as column vectors, we can express $\mathbf{j}$ as the result of three successive linear operations on the ideal image $\mathbf{l}_\lambda$

$$\mathbf{j} = \underbrace{\sum_{n=R,G,B,I} \mathbf{S}n \underbrace{\left(\int \mathrm{r}_n(\lambda) \underbrace{\mathbf{K}_\lambda \mathbf{l}_\lambda}_{\text{wavelength-dependent blur}} \mathrm{d}\lambda \right)}_{\text{irradiance of } n\text{-th filter}}}_{\text{result of RGB-IR multiplexing}} + \mathbf{e} \qquad (1)$$

where $\mathbf{e}$ denotes noise. We now describe in detail the three operations:

Wavelength-dependent lens blur: subjects close to the camera will be out of focus in at least one of the R, G, B and IR channels no matter where the camera's lens is focused. This is because the index of refraction of the lens—and thus the distance of the camera from the plane of perfect focus—depends on wavelength. In addition, optical aberrations will produce blur even at the in-focus depth, and the blur kernel will vary according to both image position and depth. We characterize the wavelength-dependent blur for each wavelength λ with a convolution matrix $\mathbf{K}_\lambda$.

Channel crosstalk: Manufacturing constraints prevent the filters of a color mosaic from blocking other bands, *e.g.*, as shown in Figure 1b. This superposition of R, G, B and IR makes it impossible to treat each channel individually, and also limits the effective dynamic range of the image. Consequently, the sensor irradiance of the n-th filter integrates that of all wavelengths, weighted by the sensor response curve r_n.

Pixel multiplexing: Due to the mosaic pattern of the color filter array, the narrow R, G, B bands are only measured at every other pixel. Therefore, the highest frequencies in an individual channel cannot be measured even if perfect channel separation and deblurring were possible. To model this, we multiply with a binary mask the image irradiance due to each color filter and then sum over all four filters to produce the linear sensor image $\mathbf{j}$. The corresponding matrix $\mathbf{S}_n$ in Eq. (1) is a diagonal matrix that stores the binary indicator of the n-th filter

$$\mathbf{S}_n[x,x] = \begin{cases} 1 & \text{if filter is } n \text{ at pixel } x \text{ ,} \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

Finally, the captured image $\mathbf{i}$ saturates at pixels where the irradiance is larger than the maximal pixel intensity $i_{\max}$

$$\mathbf{i} = \min(\mathbf{j}, i_{\max}) \ . \tag{3}$$

Approximation for restoration purposes Since the wavelength λ is continuous, the ideal image $\mathbf{l}$ has an infinite number of channels and is difficult to restore. In practice, however, we can "discretize" the multi-spectral image $\mathbf{l}$ into four pure color channels. This simplifies the model and makes it amenable to image restoration.

We assume the four sensor response functions $\mathrm{r}(\lambda) = [\mathrm{r}_R(\lambda)\ \mathrm{r}_G(\lambda)\ \mathrm{r}_B(\lambda)\ \mathrm{r}_I(\lambda)]^\top$ ($\top$ indicates matrix transpose) can be modeled as the superposition of four narrow-band response functions $\mathrm{q}(\lambda) = [\mathrm{q}_R(\lambda)\ \mathrm{q}_G(\lambda)\ \mathrm{q}_B(\lambda)\ \mathrm{q}_I(\lambda)]^\top$ through a 4×4 channel crosstalk matrix $\mathbf{C}$:

$$\mathrm{r}(\lambda) = \mathbf{C}\ \mathrm{q}(\lambda) \ . \tag{4}$$

Because we expect the bands to be narrow, we can assume that the blur kernels are constant within the band of q_n. We denote them as $\mathbf{K}_n$.

Thus we can "discretize" the ideal image into a hidden image $\mathbf{h} = \begin{bmatrix} \mathbf{h}_R \\ \mathbf{h}_G \\ \mathbf{h}_B \\ \mathbf{h}_I \end{bmatrix}$ where the four channels are

$$\mathbf{h}_n = \int \mathrm{q}_n(\lambda)\mathbf{l}(\lambda)\ \mathrm{d}\lambda \ . \tag{5}$$

Thus we simplify Eq. (1) as

$$\mathbf{j} = \mathbf{F}\mathbf{h} + \mathbf{e} \ , \tag{6}$$

where matrix $\mathbf{F}$ models the entire image formation process

$$\mathbf{F} = \mathbf{S}(\mathbf{C} \otimes \mathbf{I})\mathbf{K} \ . \tag{7}$$

Here the block diagonal matrix

$$\mathbf{K} = \mathrm{diag}(\mathbf{K}_R, \mathbf{K}_G, \mathbf{K}_B, \mathbf{K}_I) \tag{8}$$

accounts for band-specific defocus, matrix $\mathbf{C} \otimes \mathbf{I}$ models channel crosstalk ($\otimes$ denotes tensor product of matrices)

$$\mathbf{C} \otimes \mathbf{I} = \begin{bmatrix} c_{RR}\mathbf{I} & c_{RG}\mathbf{I} & c_{RB}\mathbf{I} & c_{RI}\mathbf{I} \\ c_{GR}\mathbf{I} & c_{GG}\mathbf{I} & c_{GB}\mathbf{I} & c_{GI}\mathbf{I} \\ c_{BR}\mathbf{I} & c_{BG}\mathbf{I} & c_{BB}\mathbf{I} & c_{BI}\mathbf{I} \\ c_{IR}\mathbf{I} & c_{IG}\mathbf{I} & c_{IB}\mathbf{I} & c_{II}\mathbf{I} \end{bmatrix} , \tag{9}$$

and

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_R & \mathbf{S}_G & \mathbf{S}_B & \mathbf{S}_I \end{bmatrix} \tag{10}$$

is the subsampling matrix that produces the final mosaic.

In the following we assume that defocus is a known constant, so matrix $\mathbf{F}$ in Eq. (7) is also known. We compute this matrix using a process akin to sensor and lens calibration for conventional cameras. Thus computing the hidden image $\mathbf{h}$ from input image $\mathbf{i}$ becomes a *non-blind* restoration problem.

3. Calibration

We compute matrix $\mathbf{F}$ by computing the matrices $\mathbf{K}$ and $\mathbf{C}$ that model band-specific blur and crosstalk. The subsampling matrix is fixed by the 2×2 mosaic pattern of the color filter array. Since $\mathbf{K}$ depends on the lens optics whereas $\mathbf{C}$ depends on the sensor responses, each is calibrated separately.

Blur kernel calibration We use the technique of Joshi *et al.* [5] to estimate the non-parametric blur kernels. We then fit them to the compact PSF model of Tang and Kutulakos [23] to account for defocus and aberrations.

First we need to eliminate sensor crosstalk to allow estimating the blur kernels for each channel individually. We do this by placing color filters in front of the camera to allow only a narrow band of wavelengths. The channel crosstalk for our particular prototype camera mainly occurs between RGB and infrared. Therefore, we use IR-only and RGB-only filters for this purpose.

We perform PSF calibration for individual focus settings independently. For each focus setting, we estimate blur kernels at 22 depths spanning about 1.5 meters around the in-focus position. At each depth, we estimate non-parametric blur kernels at 9×13 image positions for the RGB and the IR channel independently [5].

We then account for monochromatic and chromatic aberrations as well as defocus [23]. We fit the RGB and infrared blur kernels simultaneously, constraining all four channels to have the same monochromatic aberration parameters, and to differ only in the defocus parameter (Figure 1c-f).

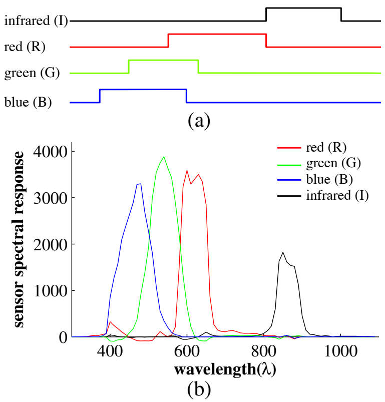

Figure 2: Crosstalk calibration results for our RGB-IR camera. **(a)** Manually specified approximate intervals of support for each channel. The support for the R, G, B, and IR channel is $[550, 800]$, $[400, 650]$, $[0, 600]$, and $[800, 1000]$ nm respectively. **(b)** Decomposed sensor spectral response functions.

Characterizing channel crosstalk We assume the sensor responses $\mathrm{r}(\lambda)$ are known for a dense sampling of wavelength λ.[1] Our goal is to find a decomposition $\mathbf{D} = \mathbf{C}^{-1}$ of spectral responses into four independent narrow bands q. We manually define a binary function δ_n to indicate the approximate support of each channel, where $\delta_n(\lambda) = 1$ if λ is within the support of channel n and $\delta_n(\lambda) = 0$ otherwise.

We then minimize the amount of demultiplexed sensor response outside the support. This is computed by solving a least-squares problem for each row $\mathbf{d}_n^\top$ of $\mathbf{D}$

$$\min_{\mathbf{d}_n} \sum_{\lambda:\delta_n(\lambda)=0} \left|\mathbf{d}_n^\top \mathrm{r}(\lambda)\right|^2 \quad \text{s.t. } \mathbf{d}_n[n] = 1 \tag{11}$$

and then computing $\mathbf{C}$ by matrix inversion

$$\mathbf{C} = \begin{bmatrix} \mathbf{d}_R & \mathbf{d}_G & \mathbf{d}_B & \mathbf{d}_I \end{bmatrix}^{-\top} \tag{12}$$

Figure 2 shows the crosstalk calibration results for our sensor. In our implementation, we multiply the crosstalk matrix $\mathbf{C}$ with a manually-defined diagonal matrix so that the hidden image is properly white balanced.

4. RGB-IR image restoration

In the absence of pixel saturation, we solve the following optimization problem to estimate the hidden image $\mathbf{h}$ in Eq. (6)

$$\min_{\mathbf{h}} |\mathbf{j} - \mathbf{F}\mathbf{h}|^2 + \rho(\mathbf{h}) \tag{13}$$

where ρ is a regularization function of the hidden image $\mathbf{h}$.

The structure of matrix $\mathbf{F}$ allows efficient solution of Eq. (13) under a Gaussian gradient prior as well as more general ones. We consider both cases below.

4.1. Direct restoration with quadratic regularizers

We first consider the following minimization problem

$$\min_{\mathbf{h}} |\mathbf{j} - \mathbf{F}\mathbf{h}|^2 + \sum_m w_m \left(|\mathbf{R}_m\mathbf{h} - \mathbf{t}_m|^2\right) . \tag{14}$$

The regularization function ρ is of the form

$$\rho(\mathbf{h}) = \sum_m w_m |\mathbf{R}_m\mathbf{h} - \mathbf{t}_m|^2 . \tag{15}$$

The weights w_m, circulant matrices $\mathbf{R}_m$, and vectors $\mathbf{t}_m$ control the specific choice of ρ. For example, if ρ is a Gaussian prior on image gradients then $m \in \{1, 2\}$, matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ implement the gradients ∇_x and ∇_y, $\mathbf{t}_1$ and $\mathbf{t}_2$ are zero, respectively, and w_1 and w_2 control the strength of regularization.

From Eqs. (7)-(10) it follows that $\mathbf{F}$ is a banded matrix with dimensions $WH \times 4WH$ for an input image of width W and height H. Because of its large size, the optimization in Eq. (14) cannot be solved directly in general. In the specific case of spatially-invariant blur, however, the optimization can be expressed in the Fourier domain and solved for each frequency independently, in one step.

Specifically, let $\hat{\mathbf{h}}_R, \hat{\mathbf{h}}_G, \hat{\mathbf{h}}_B, \hat{\mathbf{h}}_I$ and $\hat{\mathbf{j}}$ be the Fourier transform of the hidden image components $\mathbf{h}_R, \mathbf{h}_G, \mathbf{h}_B, \mathbf{h}_I$ and input image $\mathbf{j}$ respectively. A key property of the Fourier-domain image $\hat{\mathbf{j}}$ is that the elements of the four spatial frequencies

$$(u, v), (u + \pi, v), (u, v + \pi), (u + \pi, v + \pi) \tag{16}$$

depend only on the corresponding elements in the hidden image for any (u, v). In particular, if $\hat{\mathbf{j}}_{uv}$ and $\hat{\mathbf{h}}_{uv}$ collect the elements of $\hat{\mathbf{j}}$ and $\hat{\mathbf{h}}$ for those frequencies, we have

$$\hat{\mathbf{j}}_{uv} = \mathbf{F}_{uv}\hat{\mathbf{h}}_{uv} + \hat{\mathbf{e}}_{uv} , \tag{17}$$

where

$$\hat{\mathbf{F}}_{uv} = \hat{\mathbf{S}}(\mathbf{C} \otimes \mathbf{I})\hat{\mathbf{K}}_{uv} . \tag{18}$$

The matrix $\hat{\mathbf{K}}_{uv}$ is a 16×16 diagonal matrix that models per-frequency modulation due to defocus blur. The matrix $\hat{\mathbf{S}}$ is a 4×16 matrix that models subsampling in the frequency domain as a mixture of frequencies of the hidden image:

$$\hat{\mathbf{S}} = \frac{1}{4}\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} . \tag{19}$$

See Appendix A for a derivation of Eqs. (17)-(19).

We can now reformulate the optimization of Eq. (14) into $\frac{WH}{4}$ independent subproblems, each of which involves only the tuple of frequencies in Eq. (16) for some (u, v):

$$\min_{\hat{\mathbf{h}}_{uv}} |\hat{\mathbf{j}}_{uv} - \hat{\mathbf{F}}_{uv}\hat{\mathbf{h}}_{uv}|^2 + \sum_m w_m |\hat{\mathbf{R}}_{muv}\hat{\mathbf{h}}_{uv} - \hat{\mathbf{t}}_{muv}|^2 \tag{20}$$

[1] In practice we can obtain this from the filter manufacturer or through additional calibration [16].

where $\hat{\mathbf{R}}_{muv}$ is a 16×16 diagonal matrix storing Fourier transform elements of the filters corresponding to $\mathbf{R}_m$ at the four frequencies in Eq. (16), and $\hat{\mathbf{t}}_{muv}$ stores the Fourier transform elements of $\mathbf{t}_m$ at these frequencies. See Appendix B for more details on how this optimization can be performed efficiently.

In practice, the blur kernel varies due to optical aberrations. Accordingly, we relax the defocus kernel's spatial invariance by treating the input image as a collection of local patches, each of which is blurred by a different—but known—blur kernel.

4.2. Restoration with non-quadratic regularizers

Since quadratic regularization tends to oversmooth image discontinuities, we use a more general, robust regularizer. Specifically, we use a linear combination of an L1 norm on image gradients and a denoisng term [4] based on the graph Laplacian matrix

$$\rho(\mathbf{h}) = w_1 \underbrace{(|\nabla_x \mathbf{h}| + |\nabla_y \mathbf{h}|)}_{\text{L1 on gradients}} + w_2 \mathbf{h}^\top \underbrace{(\text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A})}_{\text{Laplacian matrix}} \mathbf{h} \tag{21}$$

where $\mathbf{1}$ denotes a vector of all ones and $\mathbf{A} = \text{diag}(\mathbf{A}_R, \mathbf{A}_G, \mathbf{A}_B, \mathbf{A}_I)$ is a pixel affinity matrix that takes into account pixel separation as well as intensity differences in the hidden image:

$$\mathbf{A}_n[x,y] = \exp\left(-\frac{1}{2\alpha^2}\text{dist}(x,y)^2 - \frac{1}{2\beta^2}|\mathbf{h}[x] - \mathbf{h}[y]|^2\right), \tag{22}$$

Here α and β are spatial and intensity variance parameters, respectively.

To solve Eq. (13) we use the optimization approach of Heide *et al.* [4]. See Algorithm 1 for the exact sequence of steps. This algorithm includes two steps beyond those in [4] that handle pixel saturation and are discussed in Section 5.

The most computationally expensive steps in Algorithm 1 are Steps 2, 9 and 11. To update the slack variables $\mathbf{t}_i$ in Step 9 we set $\mathbf{t}'_i = \mathbf{t}_i + \tau\bar{\mathbf{h}}$, filter it with a bilateral filter [24] of spatial variance α and intensity variance τ^{-1}, and finally subtract the result from $\mathbf{t}'_i$. We use the permutohedral lattice [1] to implement the bilateral filter efficiently. To compute the hidden image in Steps 2 and 11, we observe that this involves a quadratic optimization that can be efficiently solved with the method in Section 4.1.

Algorithm 1: Our final restoration algorithm.

```
input : input image i, image formation matrix F, weights $w_1, w_2$
parameter $\sigma = 1/400$, $\tau = 40$, $\alpha = 5$
// initialization
1  set $\mathbf{j} \leftarrow \mathbf{i}$
2  compute $\mathbf{h} \leftarrow \arg\min_{\mathbf{h}} |\mathbf{j} - \mathbf{F}\mathbf{h}|^2 + w_1|\nabla_x\mathbf{h}|^2 + w_1|\nabla_y\mathbf{h}|^2$
3  process h by bilateral filtering and inpainting saturated areas
4  set $\bar{\mathbf{h}} \leftarrow \mathbf{h}, t_x \leftarrow 0, t_y \leftarrow 0, t_i \leftarrow 0$
repeat
5      $\mathbf{h}_{\text{last}} \leftarrow \mathbf{h}$
       // enforce inequality constraint
6      $\mathbf{j} \leftarrow \mathbf{P}\mathbf{i} + (\mathbf{I} - \mathbf{P})\max(\mathbf{F}\mathbf{h}, i_{\max})$
       // regularization penalty
7      $\mathbf{t}_x \leftarrow \min(\max(\mathbf{t}_x + \tau\nabla_x\bar{\mathbf{h}}, -1), 1)$
8      $\mathbf{t}_y \leftarrow \min(\max(\mathbf{t}_y + \tau\nabla_y\bar{\mathbf{h}}, -1), 1)$
9      update $\mathbf{t}_i \leftarrow (\mathbf{t}_i + \tau(\mathbf{I} - \text{diag}(\mathbf{A}\mathbf{1})^{-1}\mathbf{A})\bar{\mathbf{h}})$
       // data fidelity
10     update $\mathbf{z} \leftarrow \mathbf{h} - \sigma w_1\nabla_x^\top\mathbf{t}_x - \sigma w_1\nabla_y^\top\mathbf{t}_y - \sigma w_2\mathbf{t}_i$
11     update $\mathbf{h} \leftarrow \arg\min_{\mathbf{h}} |\mathbf{j} - \mathbf{F}\mathbf{h}|^2 + \frac{1}{\sigma}|\mathbf{h} - \mathbf{z}|^2$
       // extrapolation
12     extrapolate $\bar{\mathbf{h}} \leftarrow 2\mathbf{h} - \mathbf{h}_{\text{last}}$
until convergence
output: estimated hidden image h
```

5. Handling pixel saturation

Because the RGB filters do not block IR, the RGB pixel intensities include an IR contribution that reduces the dynamic range of the image. In particular, saturation occurs frequently and must be handled.[2] We address this by turning Eq. (13) into a constrained minimization problem:

$$\min_{\mathbf{h},\mathbf{j}} |\mathbf{j} - \mathbf{F}\mathbf{h}|^2 + \rho(\mathbf{h}) \tag{23}$$

$$\text{s.t. } \mathbf{j} \geq \mathbf{i}, \mathbf{P}\mathbf{j} \leq \mathbf{P}\mathbf{i} \tag{24}$$

where $\mathbf{i}$ is an input image containing saturated pixels, the matrix $\mathbf{P}$ extracts unsaturated pixels, inequality constraints are applied element-wise, and the regularization function $\rho(\mathbf{h})$ is given by Eq. (21). The image $\mathbf{j}$ represents an irradiance image that does not saturate, and becomes an additional unknown that must be estimated jointly with the hidden image $\mathbf{h}$.

The inequality constraints ensure that $\mathbf{i}$ is less than $\mathbf{j}$ at all saturated pixels, and is equal to $\mathbf{j}$ otherwise. Formulating pixel saturation as inequality constraints is essential in our case because it allows us to update $\mathbf{h}$ using efficient Fourier-domain operations (Section 4.1) instead of solving a large linear system in the spatial domain.

We extend the method in Section 4.2 to solve the above problem with two steps in Algorithm 1:

- *Inpainting large saturated regions (Step 3).* We initialize $\mathbf{h}$ by inpainting the unknown color and intensity of saturated pixels. This helps Algorithm 1 converge faster. More details are given below.

[2] Of course, if it were possible to balance the dynamic range of RGB and IR bands by attenuating the stronger of RGB and IR (*e.g.*, with RGB- or IR-blocking filters), we could reduce the chance of saturation. This is hard to do in practice because the relative magnitudes of RGB and IR vary spatially, and from scene to scene.

- *Enforcing inequality constraint (Step 6).* We initialize $\mathbf{j}$ with the input image $\mathbf{i}$ in Step 1; at each iteration we then fix $\mathbf{h}$ and update each pixel of $\mathbf{j}$ to minimize $|\mathbf{j} - \mathbf{F}\mathbf{h}|^2$ within their feasible interval—$[i_{\max}, \infty)$ for saturated pixels and $\mathbf{i}$ for unsaturated pixels. Thus we do not update unsaturated pixels, and saturated pixels are updated to $\max(\mathbf{F}\mathbf{h}, i_{\max})$.

Partial saturation model Partially-saturated pixels, *i.e.* pixels whose intensity is missing in some channels but not others, are common in RGB-IR images. Although we cannot completely infer the color of such pixels, the incomplete color information they carry can inform pixel affinity—and is thus useful for inpainting. To make the most of the available color information, we represent pixel colors as 4D lines in R, G, B and IR :

$$\boldsymbol{\Theta} \begin{bmatrix} \mathbf{h}_R[x] \\ \mathbf{h}_G[x] \\ \mathbf{h}_B[x] \\ \mathbf{h}_I[x] \end{bmatrix} = \mathbf{0} \,, \tag{25}$$

where the matrix

$$\boldsymbol{\Theta} = \begin{bmatrix} \cos(\theta_{RG}) & -\sin(\theta_{RG}) & 0 & 0 \\ \cos(\theta_{RB}) & 0 & -\sin(\theta_{RB}) & 0 \\ \cos(\theta_{RI}) & 0 & 0 & -\sin(\theta_{RI}) \\ 0 & \cos(\theta_{GB}) & -\sin(\theta_{GB}) & 0 \\ 0 & \cos(\theta_{GI}) & 0 & -\sin(\theta_{GI}) \\ 0 & 0 & \cos(\theta_{BI}) & -\sin(\theta_{BI}) \end{bmatrix} \tag{26}$$

is controlled by six parameters $\theta_{RG}, \theta_{RB}, \theta_{RI}, \theta_{GB}, \theta_{GI}$ and θ_{BI}. This is inspired by the color line model [15] which has been used to handle saturation in RGB images. Specifically, given known $\mathbf{h}_m$ and $\mathbf{h}_n$, θ_{mn} can be estimated by

$$\theta_{mn}[x] = \arctan \frac{\mathbf{h}_m[x] + \epsilon}{\mathbf{h}_n[x] + \epsilon} \,. \tag{27}$$

The constant ϵ biases the color of low-intensity pixels toward gray. In practice, we first detect missing entries in θ, then inpaint them, and finally restore $\mathbf{h}$ by fixing $\boldsymbol{\Theta}$ and using Eq. (25).

Support region of saturated pixels We compute a binary map $\omega_n[x]$ to indicate which pixels in channel n of the hidden image have saturated pixels in their neighborhood:

$$\omega_n[x] = \begin{cases} 1 & \text{if } \max_{y \in \Omega_x, c_{mn} \geq 0.5} \mathbf{i}[y] < i_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

Here subscript m of c_{mn} denotes the filter type at pixel y in the color filter array. The neighborhood Ω_x corresponds to pixels at a distance less than the radius of the blur kernel. So the map ω marks pixels that carry significant color information about the saturated region.

Then we compute the support region ω'_{mn} for θ_{mn} using

$$\omega'_{mn}[x] = \omega_m[x]\omega_n[x] \tag{29}$$

since $\mathbf{h}_m[x]$ and $\mathbf{h}_n[x]$ must both be known to compute $\theta_{mn}[x]$.

Inpainting color and intensity For each pair of channels (m, n) we then inpaint the missing θ_{mn} using the algorithm of [10]. Specifically we solve the following optimization problem

$$\min_{\theta_{mn}} |\mathbf{U}\theta_{mn} - \theta_{mn}|^2 \tag{30}$$

subject to the constraint that θ_{mn} are updated only at pixels where $\omega'_{mn}[x] = 0$. The matrix $\mathbf{U}$ in Eq. (30) is an affinity matrix that takes into account pixel separation and color difference in the support region θ_{mn}:

$$\begin{aligned} \mathbf{U}[x, y] \propto \exp \Bigg(& - \frac{1}{2\alpha^2} \mathrm{dist}(x, y)^2 \\ & - \sum_{mn} \frac{\omega'_{mn}[x]\omega'_{mn}[y]}{2\beta^2} (\theta_{mn}[x] - \theta_{mn}[y])^2 \Bigg) \,, \end{aligned} \tag{31}$$

where the affinity weights are normalized to ensure $\mathbf{U}\mathbf{1} = \mathbf{1}$.

Finally, we solve for $\mathbf{h}_n$ at each pixel using Eq. (25) by fixing $\boldsymbol{\Theta}$ and solving with least squares. To deal with pixels that saturate in all channels, we simply treat them as if the IR channels were not saturated. Although this may underestimate pixel intensity, it produces visually pleasing results.

6. Results

To test our method, we ran experiments with a prototype RGB-IR camera.

Saturation handling Figure 3 shows restoration results before and after running Step 3 of Algorithm 1. Our color inpainting algorithm produces reasonable color variation in both textureless and textured multi-color regions.

Restoration quality Figures 4-6 show restoration results in the absence of defocus. These results show the notable quality improvements achieved with Algorithm 1.

Ground-truth comparisons Figure 7 and 8 show ground-truth comparisons between actual and estimated RGB and IR channels. To capture the "ground-truth" channels we used IR- and RGB-blocking filters respectively. These filters guaranteed that the captured images were free of crosstalk but did not prevent blur due to aberrations.

Computation time Matlab implementation of the direct method (Section 4.1) takes about 1 second to process a 672×760 pixel patch on a desktop computer. In comparison, a mathematically equivalent image-domain implementation takes about a minute to finish under the same settings. Full restoration is still far slower because of its iterative nature and the use of bilateral filtering. It takes about 14 seconds per iteration, with about 10 of those seconds spent on bilateral filtering, and typical images requiring 20 iterations. Removal of the denoising term from step 9 results in an operation that is three times faster, at the expense of noisier results.

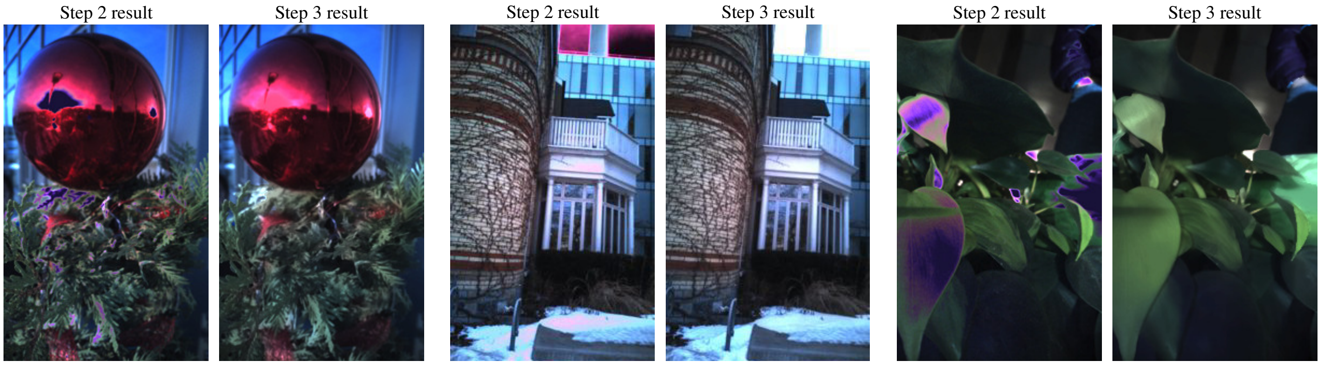

Figure 3: We compare results before and after color inpainting: the estimate of hidden image $\mathbf{h}$ by Step 2 of Algorithm 1 has errors in saturated regions. Step 3 corrects the color and intensity of such regions. We only show RGB bands since no IR saturation occurs in these examples.

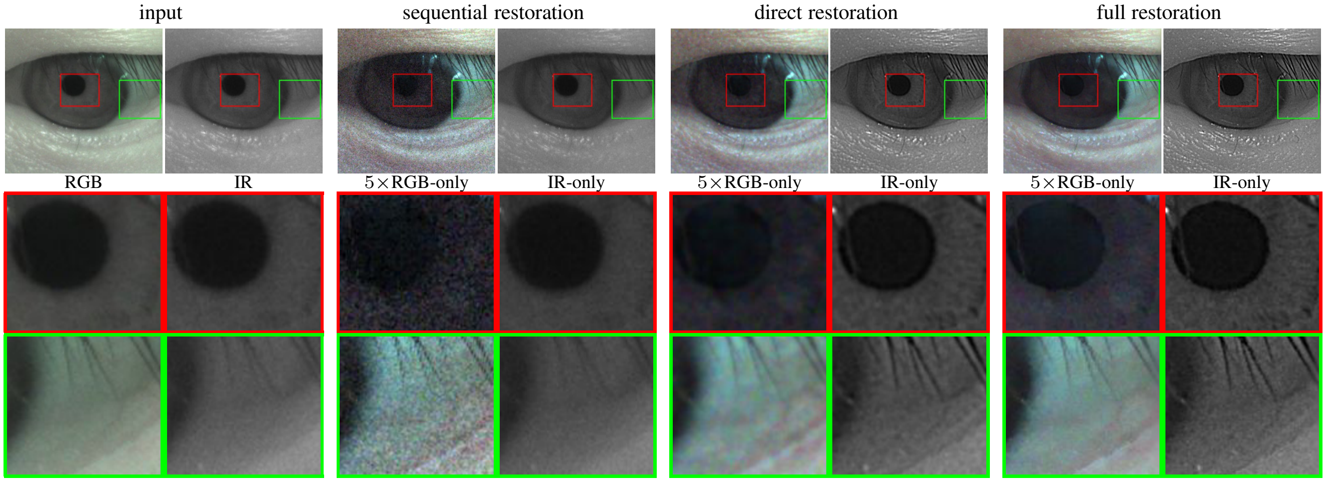

Figure 4: Restoration results. **Input:** We turn on camera autofocus to minimize defocus blur. We visualize the image mosaic with two sub-images: the three channels of the RGB image store pixels under the R, G, B color filters, and the IR image stores pixels under the IR filter. We compare three methods: **Sequential restoration** first performs demosaicing by spline interpolation, and then performs channel separation by inverting Eq.(4). Note the aliasing artifacts on eyelashes. **Direct restoration** corresponds to Steps 1-3 of Algorithm 1. It produces over-smoothed results. **Full restoration** corresponds to full execution of Algorithm 1. The final result contains clear skin and iris details that are invisible in the input, and does not suffer from noise or artifacts. Note that since the input RGB contains IR contributions as well, it is much brighter than the equivalent RGB-only images. To better visualize those, their intensities are scaled ×5.

7. Conclusion

RGB-IR cameras simultaneously suffer from three problems: pixel multiplexing, channel crosstalk and chromatic aberrations. The coupling of these problems makes application of conventional demosaicing algorithms to RGB-IR images difficult. Our key contribution is a novel image formation model for RGB-IR cameras that accounts for all three problems, allows easy calibration, and enables efficient restoration with common image regularization functions.

We believe that our approach paves the way for a variety of applications based on RGB-IR sensors. From a practical perspective, our algorithm can potentially be accelerated further by exploiting GPUs or using regularization functions that can be efficiently implemented [19].

Acknowledgements

We are grateful for the support of the Natural Sciences and Engineering Research Council of Canada under the RTI and Accelerator programs, and the MITACS Accelerate program.

References

[1] J. Baek, A. Adams, and J. Davis. Lattice-based high-dimensional gaussian filtering and the permutohedral lattice. *J. Math. Imaging Vision*, 46(2):211–237, 2013.

[2] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, May 2011.

[3] P. Common and C. Jutten. *Handbook of Blind Source Separation*. Academic Press, 2010.

[4] F. Heide, M. Steinberger, Y.-T. Tsai, N. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. FlexISP: A flexible camera image processing framework. *ACM SIGGRAPH Asia*, 33:1–13, 2014.

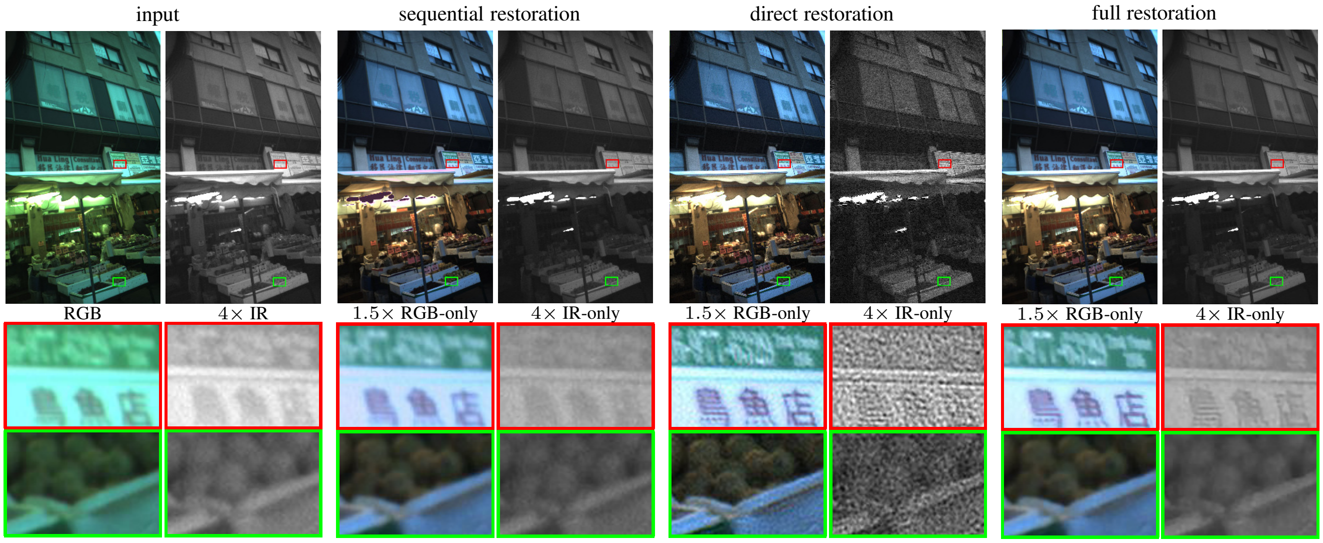

Figure 5: Restoration results for an RGB-IR image with a low-intensity IR channel. **Input:** We turn on camera autofocus, so blur is due to aberrations but not defocus. **Results:** The sequential and direct methods produce noisy IR estimates. Our full restoration, on the other hand, produces clean and sharp results.

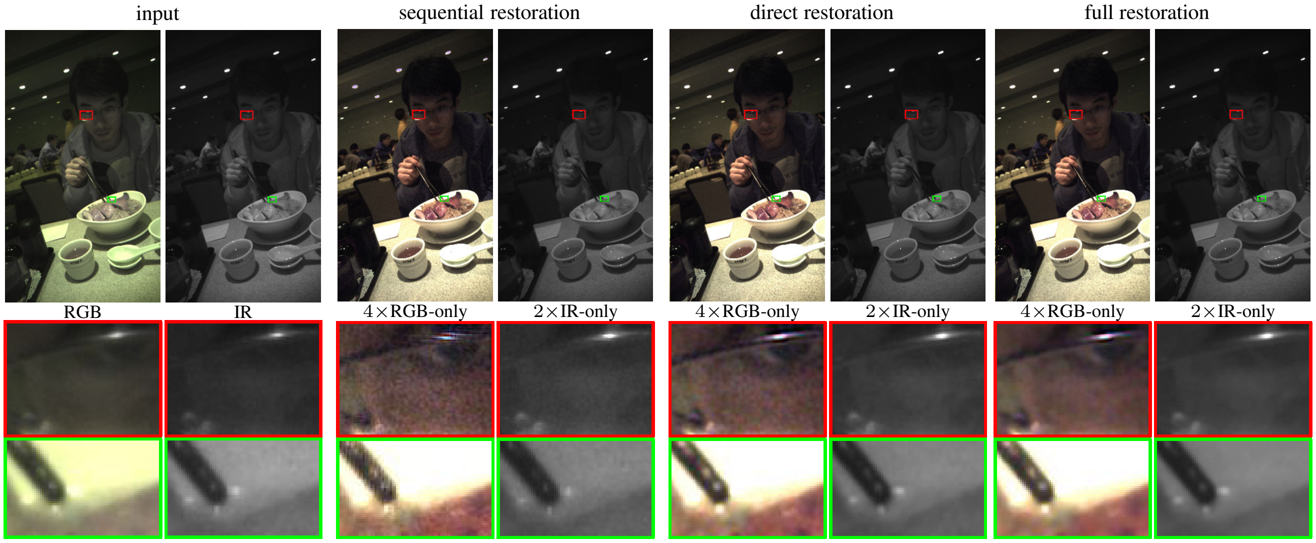

Figure 6: Restoration results for an RGB-IR image with low-intensity RGB contributions. **Input:** We turn on camera autofocus, so blur is due to aberrations but not defocus. **Results:** The sequential method causes artifacts near the specular highlight of eyeglass frame and chopsticks in the RGB-only estimate. Both the sequential and direct restoration produce noisy face estimates. In comparison, full restoration produces a clean RGB-only image with no artifacts.

[5] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *Proc. CVPR*, 2008.

[6] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi. Simultaneous capturing of RGB and additional band images using hybrid color filter array. In *Proc. SPIE*, volume 9023, 2014.

[7] R. Kimmel. Demosaicing: image reconstruction from color CCD samples. *IEEE Trans. Image Processing*, 8(9), 1999.

[8] D. Krishnan and R. Fergus. Dark flash photography. In *ACM SIGGRAPH*, 2009.

[9] G. Langfelder, T. Malzbender, A. F. Longoni, and F. Zaraga. Designing color filter arrays for the joint capture of visible and near-infrared images. In *Proc. SPIE*, pages 3797 – 3800, 2011.

[10] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM SIGGRAPH*, 23:689–694, 2004.

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. PAMI*, 33(12), 2011.

[12] Y. M. Lu, C. Fredembach, M. Vetterli, and S. Süsstrunk. Designing color filter arrays for the joint capture of visible and near-infrared images. In *Proc. ICIP*, pages 3797 – 3800, 2009.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proc. CVPR*, 2009.

[14] S. Narasimhan and S. Nayar. Enhancing resolution along multiple imaging dimensions using assorted pixels. *IEEE Trans. PAMI*, 27(4), 2005.

[15] I. Omer and M. Werman. Color lines: Image specific color representation. In *Proc. CVPR*, 2004.

[16] M. Parmar, F. Imai, S. H. Park, and J. Farrell. A database of high dynamic range visible and near-infrared multispectral images. In *Proc. SPIE*, volume 33, 2008.

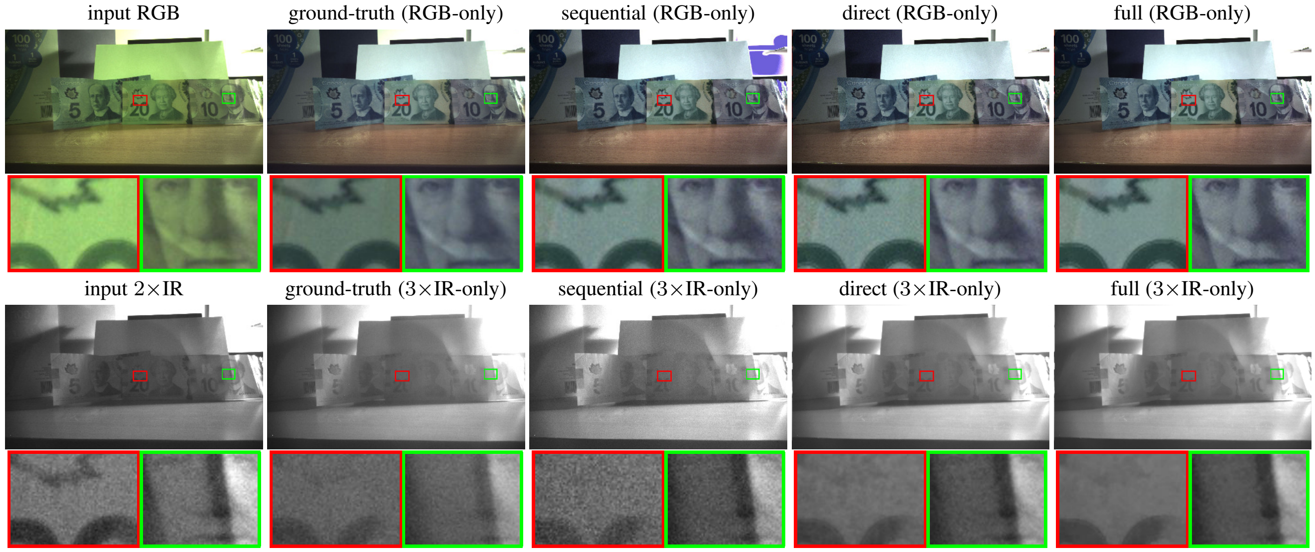

Figure 7: Recovering blur-free IR-only texture on banknotes. **Input:** We introduced defocus blur by manually focusing behind the scene. **Ground-truth images:** We captured images using IR- (top) and RGB-blocking filters (bottom). The RGB image was then white balanced to make visual comparison easier. Observe that the upper-right side of the "0" digit is washed out in the IR channel; the face and maple leaf are nearly invisible in the IR; and a strong vertical edge appears only in the IR-only image. **Results:** Although the RGB and IR channels are separated correctly in all cases, only full restoration provides clean and deblurred RGB and IR channels.

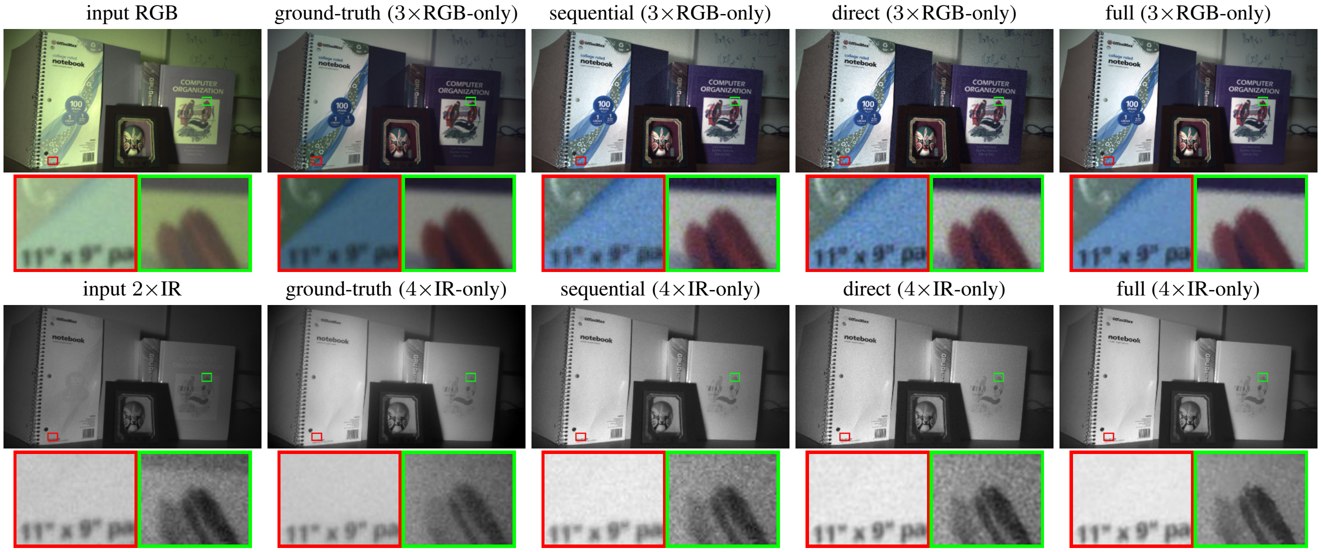

Figure 8: Recovering blur-free RGB-only textures. **Input:** We introduced defocus blur by manually focusing behind the scene. **Ground-truth images:** We follow the same procedure as in Figure 7. Observe that the texture of the left book and the purple pigment of the right book are both transparent to IR. **Results:** the RGB and IR channels are separated correctly in all cases, but only full restoration produces a sharp and clean result.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*. Cambridge University Press, 2007.

[18] Z. Sadeghipoor, Y. M. Lu, and S. Süsstrunk. A novel compressive sensing approach to simultaneously acquire color and near-infrared images on a single sensor. In *Proc. ICASSP*, 2013.

[19] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proc. CVPR*, 2014.

[20] W. Smith. *Modern Optical Engineering*. McGraw-Hill, 2000.

[21] S. Süsstrunk and C. Fredembach. Enhancing the visible with the invisible: Exploiting near-infrared to advance computational photography and computer vision. In *Proc. Symp. Information Display*, 2010.

[22] K. Tanaka, Y. Mukaigawa, Y. Matsushita, and Y. Yagi. Descattering of transmissive observation using parallel high-frequency illumination. In *Proc. ICCP*, 2013.

[23] H. Tang and K. N. Kutulakos. What does an aberrated photo tell us about the lens and the scene? In *Proc. ICCP*, 2013.

[24] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. ICCV*, 1998.

[25] M. Vollmer and K.-P. Mollmann. *Infrared Thermal Imaging: Fundamentals, Research and Applications*. Wiley, 2010.

A. Derivation of Eqs. (17)-(19)

Let $\mathbf{T}$ be the matrix implementing the Fourier transform of a single-channel image of width W and height H, Denote $\mathcal{T} = \mathrm{diag}(\mathbf{T}, \mathbf{T}, \mathbf{T}, \mathbf{T})$ the matrix implementing the Fourier transform of an RGB-IR image in each band.

We first turn Eq. (6) into a Fourier-domain formulation by multiplying both sides of Eq. (6) with $\mathbf{T}$ noting that $\mathcal{T}^\top \mathcal{T} = \mathbf{I}$

$$\begin{aligned} \hat{\mathbf{j}} &= \mathbf{Tj} = \mathbf{TFh} + \mathbf{Te} \\ &= (\mathbf{TF}\mathcal{T}^\top)\mathcal{T}\mathbf{h} + \mathbf{Te} = \hat{\mathbf{F}}\hat{\mathbf{h}} \ + \ \hat{\mathbf{e}} \ , \end{aligned} \tag{32}$$

where $\hat{\mathbf{j}}$, $\hat{\mathbf{h}}$ and $\hat{\mathbf{e}}$ denote the Fourier transform of (unsaturated) input, hidden and noise image, respectively. The matrix $\hat{\mathbf{F}}$ models image formation in the Fourier domain

$$\hat{\mathbf{F}} = \mathbf{TF}\mathcal{T}^\top \ . \tag{33}$$

By plugging Eq. (7) into Eq. (33), we have

$$\begin{aligned} \hat{\mathbf{F}} &= \mathbf{TS}(\mathbf{C} \otimes \mathbf{I})\mathbf{K}\mathcal{T}^\top \\ &= (\mathbf{TS}\mathcal{T}^\top)(\mathcal{T}(\mathbf{C} \otimes \mathbf{I})\mathcal{T}^\top)(\mathcal{T}\mathbf{K}\mathcal{T}^\top) \ . \end{aligned} \tag{34}$$

First, from Eq. (10) the Fourier-domain subsampling becomes

$$\mathbf{TS}\mathcal{T}^\top = \begin{bmatrix} \mathbf{TS}_R\mathbf{T}^\top & \mathbf{TS}_G\mathbf{T}^\top & \mathbf{TS}_B\mathbf{T}^\top & \mathbf{TS}_I\mathbf{T}^\top \end{bmatrix} . \tag{35}$$

The matrices $\mathbf{TS}_n\mathbf{T}^\top$ are circulant since $\mathbf{S}_n$ are diagonal matrices. Each row of $\mathbf{TS}_n\mathbf{T}^\top$ is a shifted version of the Fourier transform of the diagonal of $\mathbf{S}_n$, which marks the pixel locations under the n-th color filter. For the particular 2×2 mosaic pattern we discuss in this paper, the row associated with the frequency (u, v) is non-zero only in columns corresponding to frequencies at (u, v), $(u + \pi, v)$, $(u, v + \pi)$ and $(u + \pi, v + \pi)$, as in Eq. (16). The values of these entries are $\frac{1}{4}, -\frac{1}{4}, -\frac{1}{4}, \frac{1}{4}$ for $\mathbf{T}^\top\mathbf{S}_R\mathbf{T}$, $\frac{1}{4}, \frac{1}{4}, -\frac{1}{4}, -\frac{1}{4}$ for $\mathbf{T}^\top\mathbf{S}_G\mathbf{T}$, $\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$ for $\mathbf{T}^\top\mathbf{S}_B\mathbf{T}$, and $\frac{1}{4}, -\frac{1}{4}, \frac{1}{4}, -\frac{1}{4}$ for $\mathbf{T}^\top\mathbf{S}_I\mathbf{T}$, respectively.

Second, from Eq. (9), the Fourier-domain crosstalk causes superposition among different bands at each frequency

$$\mathcal{T}(\mathbf{C} \otimes \mathbf{I})\mathcal{T}^\top = \mathbf{C} \otimes (\mathbf{TT}^\top) = \mathbf{C} \otimes \mathbf{I} \ . \tag{36}$$

Third, from Eq. (8), the Fourier-domain defocus matrix is

$$\mathcal{T}\mathbf{K}\mathcal{T}^\top = \mathrm{diag}(\mathbf{TK}_R\mathbf{T}^\top, \mathbf{TK}_G\mathbf{T}^\top, \mathbf{TK}_B\mathbf{T}^\top, \mathbf{TK}_I\mathbf{T}^\top) \ , \tag{37}$$

where each matrix $\mathbf{K}_n$ is circulant when the blur kernels are spatially-invariant. Consequently $\mathcal{T}\mathbf{K}\mathcal{T}^\top$ is a diagonal matrix that stores the Fourier transform of the blur kernels in the four bands. This is simply the convolution theorem, *i.e.*, image-domain convolution corresponds to Fourier-domain modulation.

The above shows that a row in $\hat{\mathbf{F}}$ corresponding to the frequency (u, v) is non-zero only in the columns corresponding to (u, v), $(u + \pi, v)$, $(u, v + \pi)$ and $(u + \pi, v + \pi)$, *i.e.*, Eq. (16). Consequently, we can decompose Eq. (33) into $\frac{WH}{4}$ linear systems, each relevant to a particular (u, v), by extracting rows and columns corresponding to Eq. (16). This leads to Eq. (17) and Eq. (18), with $\hat{\mathbf{S}}$ defined by Eq. (19).

B. Efficient solution of Eq. (20)

For each frequency tuple (u, v), the quadratic optimization problem in Eq. (20) has a closed-form solution

$$\hat{\mathbf{h}}_{uv} = \left(\hat{\mathbf{F}}_{uv}^\top \hat{\mathbf{F}}_{uv} + \hat{\mathbf{R}}_{uv}\right)^{-1} \mathbf{z}_{uv} \tag{38}$$

where the diagonal matrix $\hat{\mathbf{R}}_{uv}$ and the vector $\mathbf{z}_{uv}$ are defined as

$$\hat{\mathbf{R}}_{uv} = \sum_m w_m \hat{\mathbf{R}}_{muv}^\top \hat{\mathbf{R}}_{muv} \ , \tag{39}$$

$$\mathbf{z}_{uv} = \hat{\mathbf{F}}_{uv}^\top \hat{\mathbf{j}}_{uv} + \sum_m w_m \hat{\mathbf{R}}_{muv}^\top \hat{\mathbf{t}}_{muv} \ . \tag{40}$$

Our key idea is to use to the Woodbury matrix identity [17]

$$\left(\hat{\mathbf{F}}_{uv}^\top \hat{\mathbf{F}}_{uv} + \hat{\mathbf{R}}_{uv}\right)^{-1} = \hat{\mathbf{R}}_{uv}^{-1} - \hat{\mathbf{R}}_{uv}^{-1}\hat{\mathbf{F}}_{uv}^\top \mathbf{Q}_{uv}^{-1} \hat{\mathbf{F}}_{uv}\hat{\mathbf{R}}_{uv}^{-1} \tag{41}$$

to simplify the inverse of 16×16 matrices $\hat{\mathbf{F}}_{uv}^\top \hat{\mathbf{F}}_{uv} + \hat{\mathbf{R}}_{uv}$ to the inverse of 4×4 matrices

$$\mathbf{Q}_{uv} = \mathbf{I} + \hat{\mathbf{F}}_{uv}\hat{\mathbf{R}}_{uv}^{-1}\hat{\mathbf{F}}_{uv}^\top . \tag{42}$$

The matrices $\mathbf{Q}_{uv}$ are small enough to allow explicit solution of its inverse, and to implement the inverse with element-wise operations.

Thus we can compute Eq. (38) as

$$\hat{\mathbf{h}}_{uv} = \hat{\mathbf{R}}_{uv}^{-1}\mathbf{z}_{uv} - (\hat{\mathbf{R}}_{uv}^{-1}\hat{\mathbf{F}}_{uv}^\top)\mathbf{Q}_{\mathbf{uv}}{}^{-1}(\hat{\mathbf{F}}_{uv}\hat{\mathbf{R}}_{uv}^{-1}\mathbf{z}_{uv}) \ . \tag{43}$$

Algorithm 2 shows the exact steps to compute Eq. (43) for all (u, v) simultaneously. Specifically, all $\mathbf{Q}_{uv}$ can be computed at the same time by linearly combining 16 basis matrices

$$\mathbf{Q}_{uv} = \mathbf{I} + \sum_l \mathbf{q}_{uv}[l]\mathbf{\Pi}_l \ . \tag{44}$$

The frequency-specific coefficients are

$$\mathbf{q}_{uv} = \mathrm{diag}(\hat{\mathbf{K}}_{uv}\hat{\mathbf{R}}_{uv}^{-1}\hat{\mathbf{K}}_{uv}) \tag{45}$$

while the basis matrices are shared by all frequencies

$$\mathbf{\Pi}_l[m, n] = \mathbf{\Psi}[m, l]\mathbf{\Psi}[n, l] \tag{46}$$

where

$$\mathbf{\Psi} = \hat{\mathbf{S}}(\mathbf{C} \otimes \mathbf{I}) \ . \tag{47}$$

This follows from substituting $\hat{\mathbf{F}}_{uv} = \mathbf{\Psi}\hat{\mathbf{K}}_{uv}$ into Eq. (42).

Algorithm 2: Computing Eq. (43) for all (u, v)

```
input : input image frequencies $\hat{\mathbf{j}}_{uv}$, camera parameters $\mathbf{C}$, $\hat{\mathbf{S}}$ and
        $\mathbf{k}_{uv} = \mathrm{diag}(\hat{\mathbf{K}}_{uv})$, regularization parameters
        $\mathbf{r}_{muv} = \mathrm{diag}(\hat{\mathbf{R}}_{muv})$, $\hat{\mathbf{t}}_{muv}$ and $w_m$
("·" and "/" denote elementwise multiplication and division)
precompute matrices $\mathbf{\Psi}$ and $\mathbf{\Pi}$ by Eqs. (46),(47)
$\mathbf{r}_{uv} \leftarrow \sum_m w_m \cdot \mathbf{r}_{muv} \cdot \mathbf{r}_{muv}$
$\mathbf{z}_{uv} \leftarrow \mathbf{\Psi}(\mathbf{k}_{uv} \cdot \hat{\mathbf{j}}_{uv}) + \sum_m w_m(\mathbf{r}_{muv} \cdot \mathbf{t}_{muv})$
compute $\mathbf{q}_{uv} \leftarrow \mathbf{k}_{uv} \cdot \mathbf{k}_{uv}/\mathbf{r}_{uv}$ and $\mathbf{Q}_{uv}$ by Eq. (44)
solve $\mathbf{x}_{uv}$ for all linear systems: $\mathbf{Q}_{uv}\mathbf{x}_{uv} = (\mathbf{z}_{uv} \cdot \mathbf{k}_{uv}/\mathbf{r}_{uv})$
compute $\hat{\mathbf{h}}_{uv} \leftarrow \mathbf{z}_{uv}/\mathbf{r}_{uv} - \mathbf{\Psi}(\mathbf{k}_{uv} \cdot \mathbf{x}_{uv}/\mathbf{r}_{uv}$
output: estimated hidden image frequencies $\hat{\mathbf{h}}_{uv}$
```