Defocus Map Estimation and Deblurring from a Single Dual-Pixel Image: Supplementary Material

Shumian Xin[1] Neal Wadhwa[2] Tianfan Xue[2] Jonathan T. Barron[2]
Pratul P. Srinivasan[2] Jiawen Chen[3] Ioannis Gkioulekas[1] Rahul Garg[2]
[1]Carnegie Mellon University [2]Google Research [3]Adobe Inc.

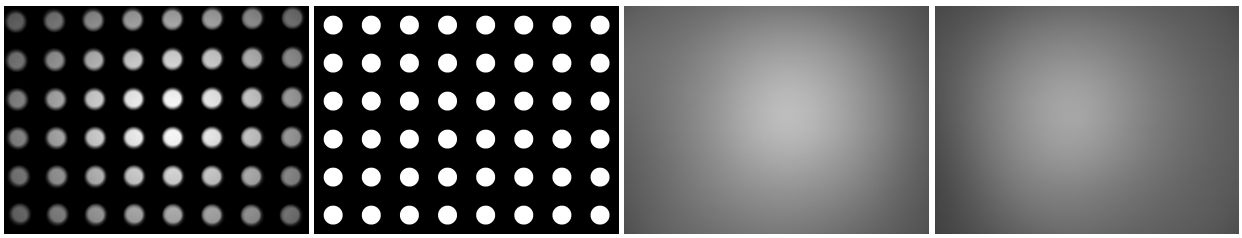
1. Introduction

In this supplementary material, we cover the following topics:

1. In Sec. 2, we describe blur kernel calibration in more detail, and explore how blur kernels change with respect to scene depth and focus distance.
2. In Sec. 3, we provide more technical details about our method. More specifically, we explain how we render defocus maps from the multi-plane image (MPI) representation, provide the derivation of the bias correction term, and define the total variation function $\boldsymbol{V}(\cdot)$ and the edge map $\boldsymbol{E}$ used in the regularization terms.
3. In Sec. 4, we provide additional implementation details, and show comparison results and ablation studies on more data in our collected Google Pixel 4 dataset. **To facilitate comparisons, we also provide an interactive HTML viewer [2] at the project website [11].**

2. Blur Kernel Calibration

We provide more information about our calibration procedure for the left and right blur kernels used as input to our method. We use a method similar to the one proposed by Mannan and Langer [6], and calibrate blur kernels for left and right dual-pixel (DP) images independently (Fig. 1) for a specific focus distance. Specifically, we image a regular grid of circular discs on a monitor screen at a distance of $\sim 45\,\mathrm{cm}$ from the camera. We apply global thresholding and binarize the captured image, perform connected component analysis to identify the individual discs and their centers, and generate and align the binary sharp image M with the known calibration pattern by solving for a homography between the calibration target disc centers and the detected centers. In order to apply radiometric correction, we also capture all-white and all-black images displayed on the same screen, and generate the grayscale latent sharp image as $I_l = M \odot I_w + (1 - M) \odot I_b$, where $\odot$ represents pixel-wise multiplication, and I_w and I_b are captured all-white and all-black images. Once we have the aligned latent image and the captured image, we can solve for spatially-varying blur kernels using the optimization proposed by Mannan and Langer [6]. Specifically, we solve for a 8×6 grid of kernels corresponding to 1344×1008 central field of view.

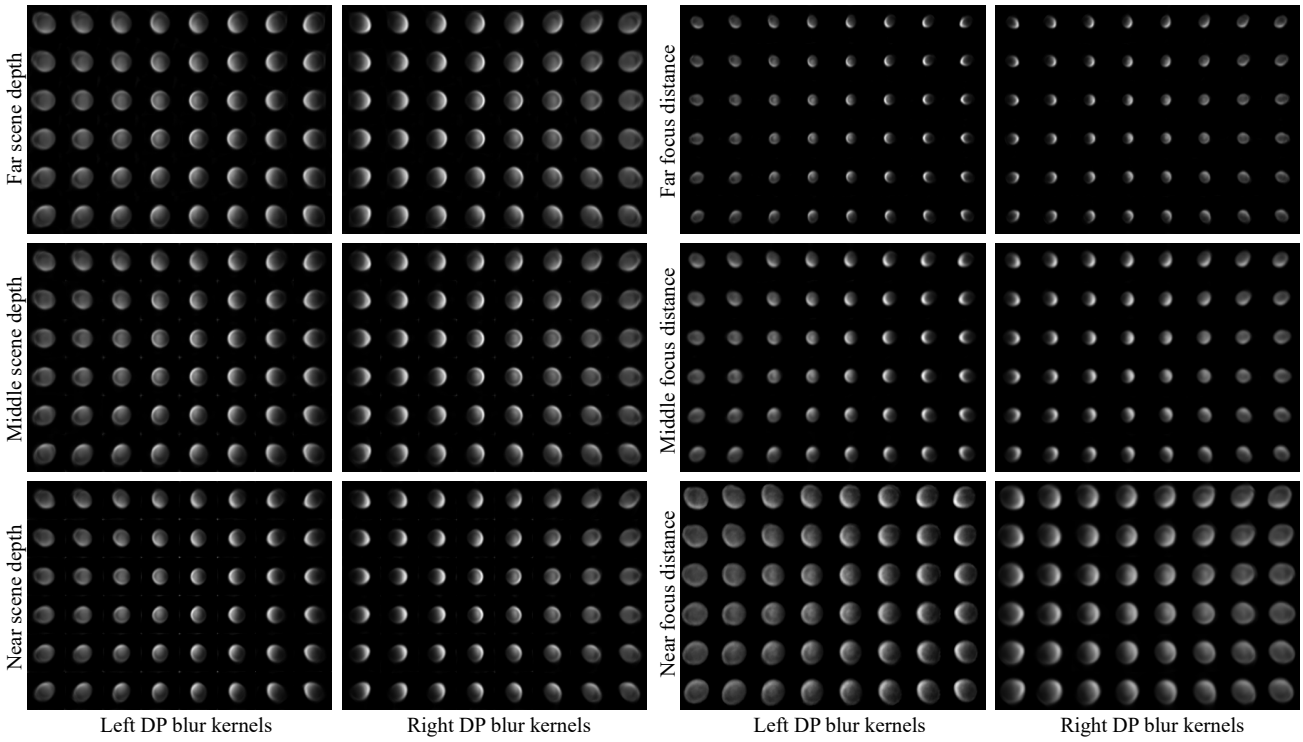


(a) Captured image (b) Calibration pattern (c) Left vignetting (d) Right vignetting

Figure 1: Captured image (a) of the calibration pattern (b) that is used to calibrate the blur kernels. Left DP image (c) and right DP image (d) of a white sheet shot through a diffuser that is used to correct for vignetting.

In addition to the blur kernels, we calibrate for different vignetting in left and right DP images. Specifically, for the same focus distance as above, we capture six images of a white sheet through a diffuser. We then average all left and right images individually to obtain the left and right vignetting patterns W_l and W_r, respectively.

Next, we explore how DP blur kernels change with respect to scene depth and focus distance (Fig. 2). As observed by Tang and Kutulakos [8], we find that kernels behave differently on the opposite sides of the focus plane. Therefore we choose focus settings such that all scene contents are at or behind the focus plane for all of our experiments, including this kernel analysis. We observe that DP blur kernels are approximately resized versions of each other as the scene depth or focus distance changes, similar to the expected behavior for blur kernels in a regular image sensor.



(a) DP blur kernels with respect to scene depth

(b) DP blur kernels with respect to focus distance

Figure 2: DP blur kernels with respect to scene depth and focus distance. We choose focus settings such that all scene contents are at or behind the focus plane, and calibrate for blur kernels either with the same focus settings but at different depths (a), or at the same depth but with various focus distances (b).

3. Additional Method Details

In this section, we provide more technical details about our method. We explain how we render defocus maps from the MPI representation in Sec. 3.1, provide the derivation of the bias correction term in Sec. 3.2, and finally define the total variation function $\boldsymbol{V}(\cdot)$ and the edge map $\boldsymbol{E}$ used in the regularization terms in Sec. 3.3.

3.1. Defocus Map from MPI

We have shown in the main paper that an all-in-focus image can be composited from an MPI representation as:

$$\hat{\mathbf{I}}_s = \sum_{i=1}^{N} \boldsymbol{t}_i \boldsymbol{c}_i = \sum_{i=1}^{N} \left[\boldsymbol{c}_i \boldsymbol{\alpha}_i \prod_{j=i+1}^{N} (1 - \boldsymbol{\alpha}_j) \right] . \tag{1}$$

We can synthesize a continuous-valued defocus map $\hat{\mathbf{D}}$ in a similar way as discussed by Tucker and Snavely [9], by replacing all pixel intensities in Eq. (1) with the defocus blur size d_i of that layer:

$$\hat{\mathbf{D}} = \sum_{i=1}^{N} \left[d_i \boldsymbol{\alpha}_i \prod_{j=i+1}^{N} (1 - \boldsymbol{\alpha}_i) \right] . \tag{2}$$

3.2. Proof of Eq. (4) of the Main Paper

In this section, we provide a detailed derivation of the bias correction term. To be self-contained, we restate our assumed image formation model. Given an MPI representation, its corresponding DP images can be expressed as:

$$\mathbf{I}_o^{\{l,r\}} = \mathbf{I}_b^{\{l,r\}} + \mathbf{N}^{\{l,r\}} , \tag{3}$$

where $\mathbf{I}_b^{\{l,r\}}$ are the latent noise-free left and right defocused images, and $\mathbf{N}^{\{l,r\}}$ is additive white Gaussian noise with entries independent identically distributed with distribution $\mathcal{N}\left(0, \sigma^2\right)$. Our goal is to optimize for an MPI with intensity-alpha layers $(\hat{\boldsymbol{c}}_i, \hat{\boldsymbol{\alpha}}_i)$, with defocus sizes $d_i, i \in [1, \ldots, N]$, such that the L_2 loss $||\hat{\mathbf{I}}_b^{\{l,r\}} - \mathbf{I}_o^{\{l,r\}}||_2^2$ is minimized. We show that, in the presence of image noise, minimizing the above loss biases the estimated defocus map towards smaller blur values. Specifically, we quantify this bias and then correct for it in our optimization.

For simplicity, we assume for now that all scene contents lie on a single fronto-parallel plane with ground truth defocus size $d^\star$, and our scene representation is an MPI with a single opaque layer (i.e., $\hat{\boldsymbol{\alpha}}_i = \mathbf{1}$) with a defocus size hypothesis d_i. Under this assumption, the defocused image rendering equation (Eq. (2) of the main paper)

$$\hat{\mathbf{I}}_b^{\{l,r\}} = \sum_{i=1}^{N} \left[\left(\boldsymbol{k}_{d_i}^{\{l,r\}} * (\boldsymbol{c}_i \boldsymbol{\alpha}_i) \right) \odot \prod_{j=i+1}^{N} \left(\mathbf{1} - \boldsymbol{k}_{d_j}^{\{l,r\}} * \boldsymbol{\alpha}_j \right) \right] \tag{4}$$

reduces to

$$\hat{\mathbf{I}}_b^{\{l,r\}} = \boldsymbol{k}_{d_i}^{\{l,r\}} * \hat{\boldsymbol{c}}_i \, . \tag{5}$$

Similarly, Eq. (3) becomes:

$$\mathbf{I}_o^{\{l,r\}} = \boldsymbol{k}_{d^\star}^{\{l,r\}} * \boldsymbol{c}_i + \mathbf{N}^{\{l,r\}} . \tag{6}$$

We can express the above equations in the frequency domain as follows:

$$\boldsymbol{\mathcal{I}}_o^{\{l,r\}} = \boldsymbol{K}_{d^\star}^{\{l,r\}} \boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^{\{l,r\}} , \tag{7}$$

where $\boldsymbol{\mathcal{I}}_o^{\{l,r\}}, \boldsymbol{K}_{d^\star}^{\{l,r\}}, \boldsymbol{C}_i$, and $\boldsymbol{\mathcal{N}}^{\{l,r\}}$ are the Fourier transforms of $\mathbf{I}_o^{\{l,r\}}, \boldsymbol{k}_{d^\star}^{\{l,r\}}, \boldsymbol{c}_i$, and $\mathbf{N}^{\{l,r\}}$, respectively. Note that the entries of $\boldsymbol{\mathcal{N}}^{\{l,r\}}$ are also independent identically distributed with the same Gaussian distribution $\mathcal{N}\left(0, \sigma^2\right)$ as the entries of $\mathbf{N}^{\{l,r\}}$.

We can obtain a maximum a posteriori (MAP) estimate of $\hat{\boldsymbol{C}}_i$ and d_i by solving the following optimization problem [12]:

$$\begin{aligned} &\arg\max P\left(\boldsymbol{\mathcal{I}}_o^l, \boldsymbol{\mathcal{I}}_o^r | \hat{\boldsymbol{C}}_i, d_i, \sigma\right) P\left(\hat{\boldsymbol{C}}_i, d_i\right) \\ &= \arg\max P\left(\boldsymbol{\mathcal{I}}_o^l, \boldsymbol{\mathcal{I}}_o^r | \hat{\boldsymbol{C}}_i, d_i, \sigma\right) P\left(\hat{\boldsymbol{C}}_i\right) . \end{aligned} \tag{8}$$

According to Eq. (7), we have

$$P\left(\boldsymbol{\mathcal{I}}_o^l, \boldsymbol{\mathcal{I}}_o^r | \hat{\boldsymbol{C}}_i, d_i, \sigma\right) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{v=\{l,r\}} ||\boldsymbol{K}_{d_i}^v \hat{\boldsymbol{C}}_i - \boldsymbol{\mathcal{I}}_o^v||^2 \right) . \tag{9}$$

We also follow Zhou et al. [12] in assuming a prior for the latent all-in-focus image such that:

$$P\left(\hat{\boldsymbol{C}}_i\right) \propto \exp\left(-\frac{1}{2} ||\boldsymbol{\Phi} \hat{\boldsymbol{C}}_i||^2 \right) , \tag{10}$$

where we define $\mathbf{\Phi}$ such that

$$\left|\mathbf{\Phi}\left(f\right)\right|^2 = \frac{1}{\left|\hat{\boldsymbol{C}}_i\left(f\right)\right|^2}, \tag{11}$$

and f is the frequency. As $\hat{\boldsymbol{C}}_i$ is the unknown variable, we approximate Eq. (11) by averaging the power spectrum over a set of natural images $\{\boldsymbol{C}_i\}$:

$$\left|\mathbf{\Phi}\left(f\right)\right|^2 = \frac{1}{\int_{\boldsymbol{C}_i}\left|\boldsymbol{C}_i\left(f\right)\right|^2 \mu\left(\boldsymbol{C}_i\right)}, \tag{12}$$

where $\mu\left(\boldsymbol{C}_i\right)$ represents the probability distribution of $\boldsymbol{C}_i$ in image domain.

Maximizing the log-likelihood of Eq. (8) is equivalent to minimizing the following loss:

$$E\left(d_i|\boldsymbol{\mathcal{I}}_o^l, \boldsymbol{\mathcal{I}}_o^r, \sigma\right) = \min_{\hat{\boldsymbol{C}}_i}\left(\sum_{v=\{l,r\}}||\boldsymbol{K}_{d_i}^v\hat{\boldsymbol{C}}_i - \boldsymbol{\mathcal{I}}_o^v||^2\right) + ||\sigma\mathbf{\Phi}\hat{\boldsymbol{C}}_i||^2. \tag{13}$$

d_i can be estimated as the minimizer of the above energy function. Then given d_i, setting $\partial E/\partial\hat{\boldsymbol{C}}_i = 0$ yields the following solution of $\hat{\boldsymbol{C}}_i$, known as a *generalized Wiener deconvolution with two observations*:

$$\hat{\boldsymbol{C}}_i = \frac{\boldsymbol{\mathcal{I}}_o^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{I}}_o^r\overline{\boldsymbol{K}_{d_i}^r}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2\left|\mathbf{\Phi}\right|^2}, \tag{14}$$

where $\overline{\boldsymbol{K}_{d_i}^{\{l,r\}}}$ is the complex conjugate of $\boldsymbol{K}_{d_i}^{\{l,r\}}$, and $\left|\boldsymbol{K}_{d_i}^{\{l,r\}}\right|^2 = \boldsymbol{K}_{d_i}^{\{l,r\}}\overline{\boldsymbol{K}_{d_i}^{\{l,r\}}}$.

We then evaluate the defocus size hypothesis d_i by computing the minimization loss given the latent ground truth depth $d^\star$, and the noise level σ, that is,

$$E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right) = \mathbb{E}_{\boldsymbol{C}_i,\boldsymbol{\mathcal{I}}_o^l,\boldsymbol{\mathcal{I}}_o^r}E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma, \boldsymbol{C}_i, \boldsymbol{\mathcal{I}}_o^l, \boldsymbol{\mathcal{I}}_o^r\right) \tag{15}$$

$$= \mathbb{E}_{\boldsymbol{C}_i,\boldsymbol{\mathcal{I}}_o^l,\boldsymbol{\mathcal{I}}_o^r}\left[\left(\sum_{v=\{l,r\}}||\boldsymbol{K}_{d_i}^v\hat{\boldsymbol{C}}_i - \boldsymbol{\mathcal{I}}_o^v||^2\right) + ||\sigma\mathbf{\Phi}\hat{\boldsymbol{C}}_i||^2\right], \tag{16}$$

where $\mathbb{E}\left(\cdot\right)$ is the expectation. Substituting $\hat{\boldsymbol{C}}_i$ with Eq. (14) gives us:

$$\begin{aligned}&E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right)\\ =&\mathbb{E}_{\boldsymbol{C}_i,\boldsymbol{\mathcal{I}}_o^l,\boldsymbol{\mathcal{I}}_o^r}\left[\left(\sum_{v=\{l,r\}}||\boldsymbol{K}_{d_i}^v\frac{\boldsymbol{\mathcal{I}}_o^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{I}}_o^r\overline{\boldsymbol{K}_{d_i}^r}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2\left|\mathbf{\Phi}\right|^2} - \boldsymbol{\mathcal{I}}_o^v||^2\right) + ||\sigma\mathbf{\Phi}\frac{\boldsymbol{\mathcal{I}}_o^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{I}}_o^r\overline{\boldsymbol{K}_{d_i}^r}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2\left|\mathbf{\Phi}\right|^2}||^2\right].\end{aligned} \tag{17}$$

Then substituting $\boldsymbol{\mathcal{I}}_o^v$ with Eq. (7), we get:

$$\begin{aligned}&E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right)\\ =&\mathbb{E}_{\boldsymbol{C}_i,\boldsymbol{\mathcal{N}}^l,\boldsymbol{\mathcal{N}}^r}\left[\left(\sum_{v=\{l,r\}}||\boldsymbol{K}_{d_i}^v\frac{\left(\boldsymbol{K}_{d^\star}^l\boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^l\right)\overline{\boldsymbol{K}_{d_i}^l} + \left(\boldsymbol{K}_{d^\star}^r\boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^r\right)\overline{\boldsymbol{K}_{d_i}^r}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2\left|\mathbf{\Phi}\right|^2} - \left(\boldsymbol{K}_{d^\star}^v\boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^v\right)||^2\right) +\right.\\ &\left.||\sigma\mathbf{\Phi}\frac{\left(\boldsymbol{K}_{d^\star}^l\boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^l\right)\overline{\boldsymbol{K}_{d_i}^l} + \left(\boldsymbol{K}_{d^\star}^r\boldsymbol{C}_i + \boldsymbol{\mathcal{N}}^r\right)\overline{\boldsymbol{K}_{d_i}^r}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2\left|\mathbf{\Phi}\right|^2}||^2\right].\end{aligned} \tag{18}$$

We now define $B = \left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2 |\boldsymbol{\Phi}|^2$. We can rearrange the above equation as:

$$
\begin{aligned}
&E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right) \\
=&\mathbb{E}_{\boldsymbol{C}_i, \boldsymbol{\mathcal{N}}^l, \boldsymbol{\mathcal{N}}^r}\left[\left(\sum_{v=\{l,r\}} || \frac{\boldsymbol{C}_i\left[\boldsymbol{K}_{d_i}^v\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right) - \boldsymbol{K}_{d^\star}^v B\right]}{B} + \frac{\boldsymbol{K}_{d_i}^v\left(\boldsymbol{\mathcal{N}}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{N}}^r\overline{\boldsymbol{K}_{d_i}^r}\right)}{B} - \boldsymbol{\mathcal{N}}^v||^2\right) + \right. \\
&\left. ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{C}_i\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right)}{B} + \sigma\boldsymbol{\Phi}\frac{\boldsymbol{\mathcal{N}}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{N}}^r\overline{\boldsymbol{K}_{d_i}^r}}{B}||^2\right].
\end{aligned} \tag{19}
$$

Given that the entries of $\boldsymbol{\mathcal{N}}^{\{l,r\}}$ are independent identically distributed with distribution $\mathcal{N}\left(0, \sigma^2\right)$, we have $\mathbb{E}\left(\boldsymbol{\mathcal{N}}^v\right) = \mathbf{0}, \mathbb{E}\left(\boldsymbol{\mathcal{N}}^{v\,2}\right) = \boldsymbol{\sigma}^2$ and $\mathbb{E}\left(\boldsymbol{\mathcal{N}}^l\boldsymbol{\mathcal{N}}^r\right) = \mathbf{0}$, and we can simplify the above equation as:

$$
\begin{aligned}
&E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right) \\
=&\mathbb{E}_{\boldsymbol{C}_i, \boldsymbol{\mathcal{N}}^l, \boldsymbol{\mathcal{N}}^r}\left[\left(\sum_{v=\{l,r\}} ||\frac{\boldsymbol{C}_i\left[\boldsymbol{K}_{d_i}^v\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right) - \boldsymbol{K}_{d^\star}^v B\right]}{B}||^2 + ||\frac{\boldsymbol{K}_{d_i}^v\left(\boldsymbol{\mathcal{N}}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{N}}^r\overline{\boldsymbol{K}_{d_i}^r}\right)}{B} - \boldsymbol{\mathcal{N}}^v||^2\right) + \right. \\
&\left. ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{C}_i\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right)}{B}||^2 + ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{\mathcal{N}}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{\mathcal{N}}^r\overline{\boldsymbol{K}_{d_i}^r}}{B}||^2\right]
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
=&\mathbb{E}_{\boldsymbol{C}_i}\left\{\left[\sum_{v=\{l,r\}} ||\frac{\boldsymbol{C}_i\left[\boldsymbol{K}_{d_i}^v\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right) - \boldsymbol{K}_{d^\star}^v B\right]}{B}||^2 + \sigma^2\left(||\frac{\boldsymbol{K}_{d_i}^{v\,2} + \sigma^2|\boldsymbol{\Phi}|^2}{B}||^2 + ||\frac{\boldsymbol{K}_{d_i}^l\boldsymbol{K}_{d_i}^r}{B}||^2\right)\right] + \right. \\
&\left. ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{C}_i\left(\boldsymbol{K}_{d^\star}^l\overline{\boldsymbol{K}_{d_i}^l} + \boldsymbol{K}_{d^\star}^r\overline{\boldsymbol{K}_{d_i}^r}\right)}{B}||^2 + \sigma^2\left(||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{K}_{d_i}^l}{B}||^2 + ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{K}_{d_i}^r}{B}||^2\right)\right\}.
\end{aligned} \tag{21}
$$

Recall that, in Eq. (12), we defined $\boldsymbol{\Phi}(f)$ such that $\frac{1}{|\boldsymbol{\Phi}(f)|^2} = \int_{\boldsymbol{C}_i}|\boldsymbol{C}_i(f)|^2\,\mu(\boldsymbol{C}_i)$. Then we can further simplify $E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right)$ as:

$$
\begin{aligned}
&E\left(d_i|\boldsymbol{K}_{d^\star}^l, \boldsymbol{K}_{d^\star}^r, \sigma\right) \\
=&\sum_f\left[\frac{\frac{1}{|\boldsymbol{\Phi}|^2}\left|\boldsymbol{K}_{d^\star}^l\boldsymbol{K}_{d_i}^r - \boldsymbol{K}_{d^\star}^r\boldsymbol{K}_{d_i}^l\right|^2}{B}\right] + \sum_f\left[\frac{\frac{1}{|\boldsymbol{\Phi}|^2}\sigma^2|\boldsymbol{\Phi}|^2\left(\left|\boldsymbol{K}_{d^\star}^l\right|^2 + \left|\boldsymbol{K}_{d^\star}^r\right|^2\right)}{B}\right] + \\
&\sum_f\left[\sigma^2\left(||\frac{\boldsymbol{K}_{d_i}^{l\,2} + \sigma^2|\boldsymbol{\Phi}|^2}{B}||^2 + ||\frac{\boldsymbol{K}_{d_i}^{r\,2} + \sigma^2|\boldsymbol{\Phi}|^2}{B}||^2 + 2||\frac{\boldsymbol{K}_{d_i}^l\boldsymbol{K}_{d_i}^r}{B}||^2 + ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{K}_{d_i}^l}{B}||^2 + ||\sigma\boldsymbol{\Phi}\frac{\boldsymbol{K}_{d_i}^r}{B}||^2\right)\right]
\end{aligned} \tag{22}
$$

$$
=\sum_f\left[\frac{\frac{1}{|\boldsymbol{\Phi}|^2}\left|\boldsymbol{K}_{d^\star}^l\boldsymbol{K}_{d_i}^r - \boldsymbol{K}_{d^\star}^r\boldsymbol{K}_{d_i}^l\right|^2}{B}\right] + \sigma^2\sum_f\left[\frac{\left|\boldsymbol{K}_{d^\star}^l\right|^2 + \left|\boldsymbol{K}_{d^\star}^r\right|^2}{B} + \frac{\sigma^2|\boldsymbol{\Phi}|^2}{B} + 1\right] \tag{23}
$$

$$
=\sum_f\left[\frac{\frac{1}{|\boldsymbol{\Phi}|^2}\left|\boldsymbol{K}_{d^\star}^l\boldsymbol{K}_{d_i}^r - \boldsymbol{K}_{d^\star}^r\boldsymbol{K}_{d_i}^l\right|^2}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2|\boldsymbol{\Phi}|^2}\right] + \sigma^2\sum_f\left[\frac{\left|\boldsymbol{K}_{d^\star}^l\right|^2 + \left|\boldsymbol{K}_{d^\star}^r\right|^2 + \sigma^2|\boldsymbol{\Phi}|^2}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2|\boldsymbol{\Phi}|^2} + 1\right]. \tag{24}
$$

If we define $C_1\left(\boldsymbol{K}_{d_i}^{\{l,r\}}, \sigma, \boldsymbol{\Phi}\right) = \frac{\frac{1}{|\boldsymbol{\Phi}|^2}}{\left|\boldsymbol{K}_{d_i}^l\right|^2 + \left|\boldsymbol{K}_{d_i}^r\right|^2 + \sigma^2|\boldsymbol{\Phi}|^2}$, and $C_2(\sigma) = \sigma^2\sum_f 1$, then Eq. (24) boils down to Eq. (4) of the main paper.

3.3. Edge-aware Total Variation Function

We first define a pixel-wise total variation function of a single-layer image $\mathbf{I}$ that in used in both the intensity smoothness prior $\mathcal{L}_{\text{intensity}}$ and the alpha and transmittance smoothness prior $\mathcal{L}_{\text{alpha}}$:

$$\boldsymbol{V}(\mathbf{I}) = \sqrt{\mathbf{I}^2 * g - (\mathbf{I} * g)^2}, \tag{25}$$

where g is a two-dimensional Gaussian blur kernel:

$$g = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}. \tag{26}$$

Each "pixel" in $\boldsymbol{V}(\mathbf{I})(x, y)$ is equivalent to, for the 3×3 window surrounding pixel (x, y) in $\mathbf{I}$, computing the sample standard deviation (weighted by a Gaussian kernel) of the pixel intensities in that window. This follows easily from two facts: 1) as g sums to 1 by construction, $\mathbf{I} * g$ produces an image whose pixel intensities can be viewed as expectations of their surrounding 3×3 input patch; and 2) the standard deviation $\sqrt{\mathbb{E}\left[(X - \mathbb{E}[X])^2\right]}$ can be written equivalently as $\sqrt{\mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2}$.

As done in prior work [9], we would like to encourage edge-aware smoothness in addition to minimizing total variation, so a bilateral edge mask is computed using this total variation:

$$\boldsymbol{E}(\mathbf{I}) = \mathbf{1} - \exp\left(-\frac{\mathbf{I}^2 * g - (\mathbf{I} * g)^2}{2\beta^2}\right). \tag{27}$$

In this equation, β is set to $1/32$ (assuming pixel intensities are in $[0, 1]$). A joint total variation function that takes into account both the original and the edge-aware total variation is then defined as:

$$\boldsymbol{V}_{\boldsymbol{E}}(\mathbf{I}, \boldsymbol{E}) = \ell(\boldsymbol{V}(\mathbf{I})) + (1 - \boldsymbol{E}) \odot \ell(\boldsymbol{V}(\mathbf{I})). \tag{28}$$

4. Additional Implementation Details and Experimental Results

We first discuss more implementation details about our method, then show qualitative results on more data in our collected Google Pixel 4 dataset in Fig. 4-6. We also provide an interactive HTML viewer [2] at the project website [11] to facilitate the comparisons.

Data normalization. Before running the optimization, we first compute an intensity scaling factor $s = 0.5/\text{mean}\left(\mathbf{I}_o^{\{l,r\}}\right)$, and normalize the inputs $\bar{\mathbf{I}}_o^{\{l,r\}} = s\mathbf{I}_o^{\{l,r\}}$ to account for global intensity variations. After optimization, we undo the normalization by dividing the all-in-focus image by s.

Scaling factors of each loss term. Recall that our optimization loss is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{data}} + \lambda_2 \mathcal{L}_{\text{aux}} + \lambda_3 \mathcal{L}_{\text{intensity}} + \lambda_4 \mathcal{L}_{\text{alpha}} + \lambda_5 \mathcal{L}_{\text{entropy}}. \tag{29}$$

$\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{aux}}$ have the same weight: $\lambda_1 = \lambda_2 = 2.5 \cdot 10^4$. For most scenes, $\lambda_3 = 30$, $\lambda_4 = 7.5 \cdot 10^4$, and $\lambda_5 = 12$. We set higher weights on the regularization terms $\mathcal{L}_{\text{intensity}}$, $\mathcal{L}_{\text{alpha}}$, and $\mathcal{L}_{\text{entropy}}$ for scenes with less texture, e.g., data from Abuolaim and Brown [1].

Kernel size of each MPI layer. We manually determine the kernel sizes of the front and back layers, and evenly distribute MPI layers in disparity space. As mentioned in Sec. 2, we choose focus settings such that all scene contents are at or behind the focus plane. Therefore the kernel size of the front MPI layer is usually set to a small positive number, e.g., in the range of 1×1 to 3×3, to mimic a 2D delta function, while the kernel size corresponding to the back MPI layer is set to a large enough value, e.g., in the range of 57×57 to 61×61, to represent blur kernels at infinity.

Quantitative Metrics. We use the commercial software, Helicon Focus [4] to compute the ground truth all-in-focus images and the defocus maps using focus stacking. There may be a small *shift* between the ground truth all-in-focus image and the all-in-focus image from the deblurring algorithms we evaluate. This is because one can apply an arbitrary transform to the blur kernels and an inverse transform to the recovered all-in-focus image to yield the same blurred image. We determine this shift for each algorithm by using OpenCV to align the ground-truth all-in-focus image with the all-in-focus image from the algorithm via an affine transform for a single specific scene, and then using that transform to align all images before

computing the metrics for all-in-focus images. We also crop a small border of 8 pixels before computing the metrics as it may contain invalid pixels after alignment.

Additional results for bias correction. Fig. 3 shows additional results for our ablation study. Specifically, it shows that without bias correction term $\mathcal{B}$, the estimated defocus size is smaller on average as predicted by our analysis.

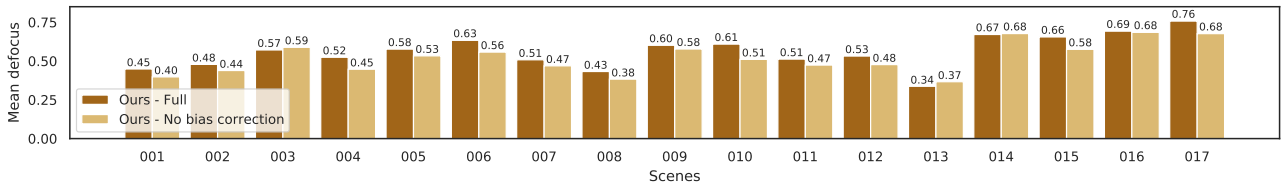


Figure 3: Mean of the predicted defocus map for our full pipeline vs an ablation where bias correction term is not applied. Defocus is measured as the relative scaling applied to the calibrated kernels. Without bias correction, the mean defocus is lower in 14 of the 17 scenes, i.e., the prediction is biased towards smaller defocus size.

References

[1] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. *European Conference on Computer Vision*, 2020. 6, 8

[2] Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. Reversible jump metropolis light transport using inverse mappings. *ACM Transactions on Graphics*, 2017. 1, 6

[3] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. *IEEE/CVF International Conference on Computer Vision*, 2019. 8

[4] Helicon focus. https://www.heliconsoft.com/. 6

[5] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 8

[6] Fahim Mannan and Michael S. Langer. Blur calibration for depth from defocus. *Conference on Computer and Robot Vision*, 2016. 1

[7] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S. Brown. Modeling defocus-disparity in dual-pixel sensors. *IEEE International Conference on Computational Photography*, 2020. 8

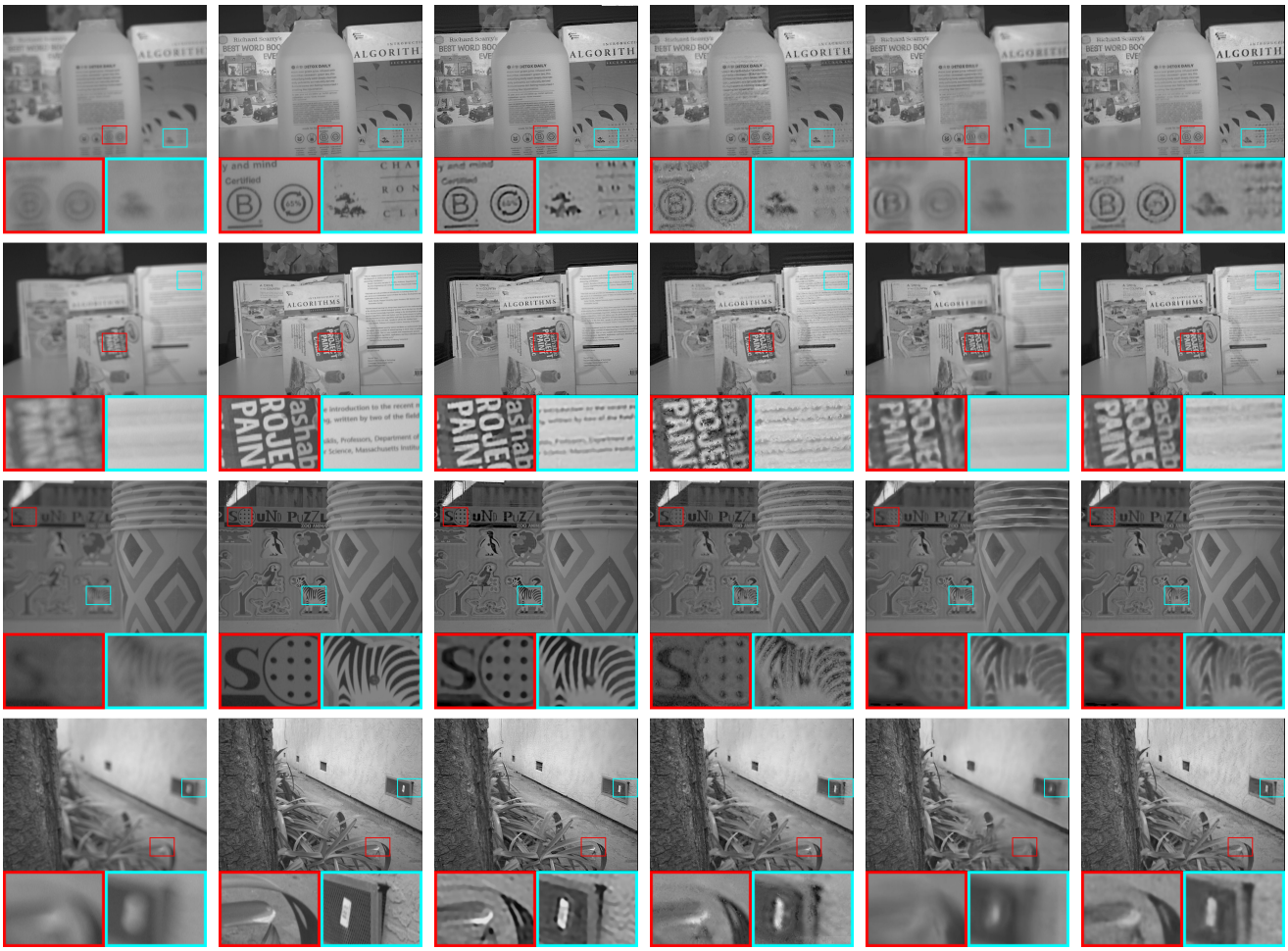
[8] Huixuan Tang and Kiriakos N. Kutulakos. Utilizing optical aberrations for extended-depth-of-field panoramas. *Asian Conference on Computer Vision*, 2012. 2

[9] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3, 6

[10] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, 2018. 8

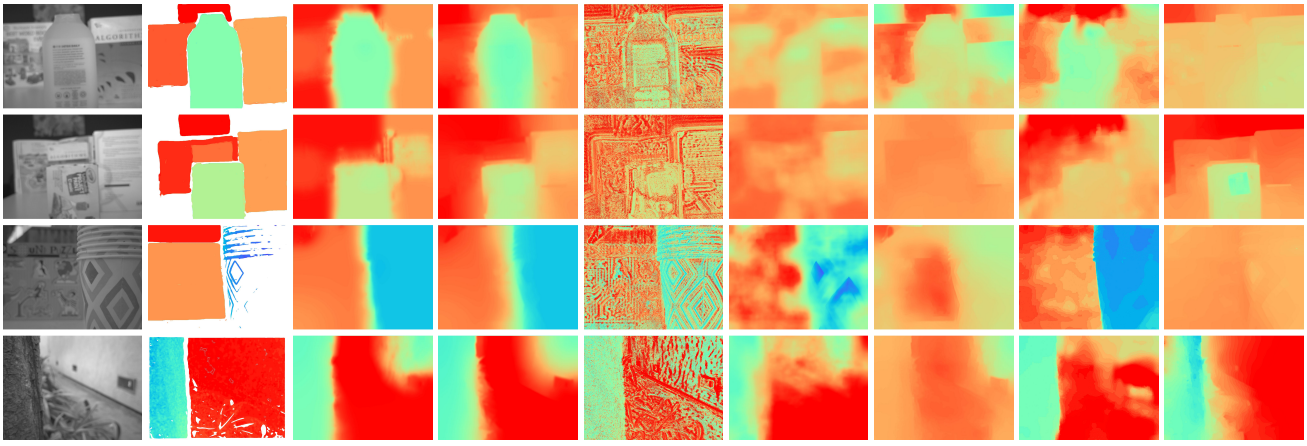
[11] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T. Barron, Pratul P. Srinivasan, Jianwen Chen, Ioannis Gkioulekas, and Rahul Garg. Project website, 2021. https://imaging.cs.cmu.edu/dual_pixels. 1, 6

[12] Changyin Zhou, Stephen Lin, and Shree K. Nayar. Coded aperture pairs for depth from defocus. *IEEE/CVF International Conference on Computer Vision*, 2009. 3, 8



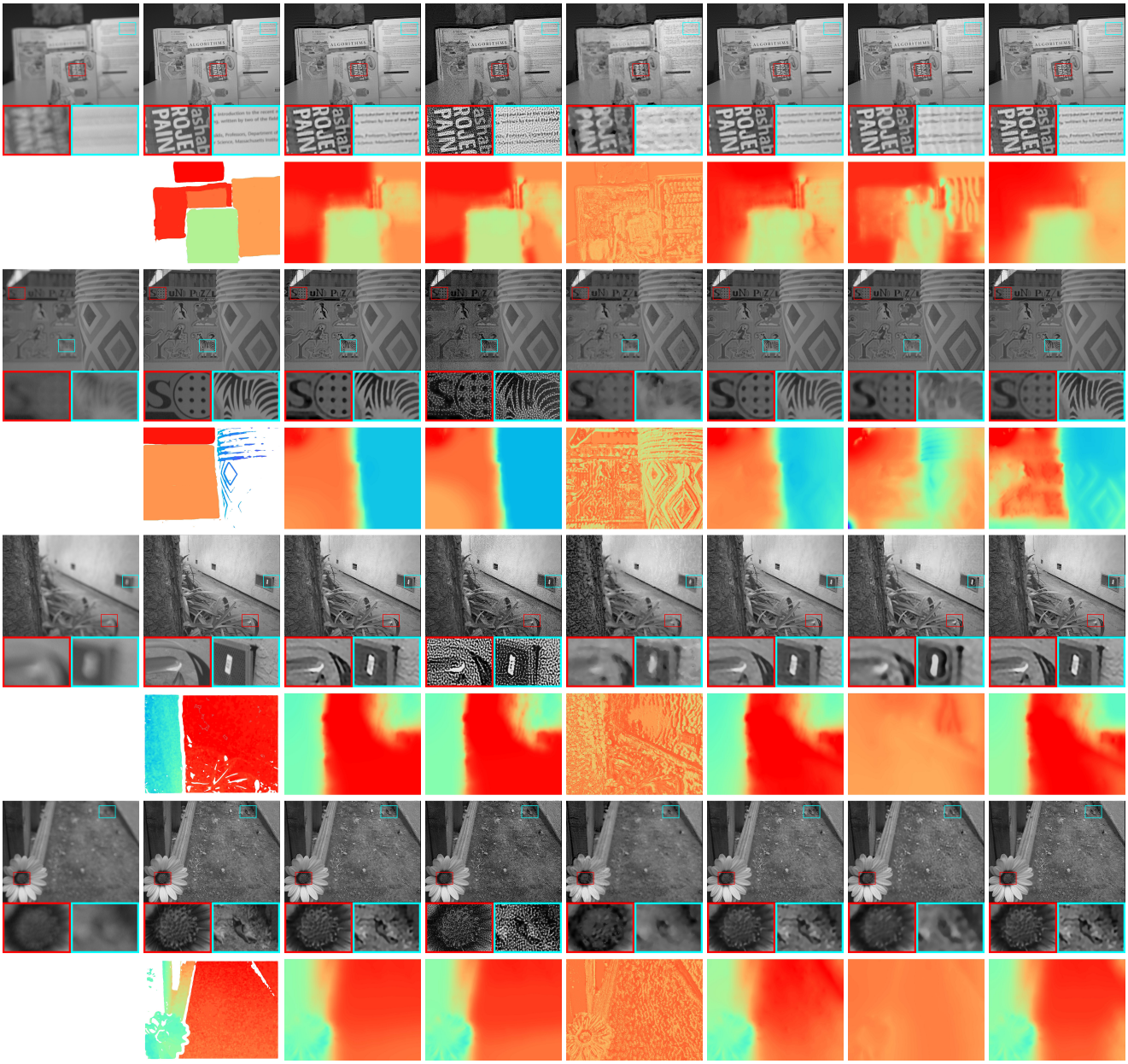
(a) Input image (b) GT all-in-focus image (c) Ours (d) Wiener deconv. [12] (e) DPDNet [1] (f) DPDNet w/ calib [1]

Figure 4: More qualitative comparisons of various defocus deblurring methods.



(a) Input image (b) Ground Truth (c) Ours (d) Ours w/ GF (e) Wiener [12] (f) DMENet [5] (g) [7] (h) Garg [3] (i) Wadhwa [10]

Figure 5: More qualitative comparisons of defocus map estimation methods.



(a) Input image (b) Ground truth (c) Ours full (d) No $\mathcal{L}_{\text{intensity}}$ (e) No $\mathcal{L}_{\text{alpha}}$ (f) No $\mathcal{L}_{\text{entropy}}$ (g) No $\mathcal{L}_{\text{aux}}$ (h) No $\mathcal{B}$

Figure 6: More qualitative results on ablation study.