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

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.

Abstract

We present a method that takes as input a single dual-pixel image, and simultaneously estimates the image's defocus map—the amount of defocus blur at each pixel—and recovers an all-in-focus image. Our method is inspired from recent works that leverage the dual-pixel sensors available in many consumer cameras to assist with autofocus, and use them for recovery of defocus maps or all-in-focus images. These prior works have solved the two recovery problems independently of each other, and often require large labeled datasets for supervised training. By contrast, we show that it is beneficial to treat these two closely-connected problems simultaneously. To this end, we set up an optimization problem that, by carefully modeling the optics of dual-pixel images, jointly solves both problems. We use data captured with a consumer smartphone camera to demonstrate that, after a one-time calibration step, our approach improves upon prior works for both defocus map estimation and blur removal, despite being entirely unsupervised.

1. Introduction

Modern DSLR and mirrorless cameras feature large-aperture lenses that allow collecting more light, but also introduce *defocus* blur, meaning that objects in images appear blurred by an amount proportional to their distance from the focal plane. A simple way to reduce defocus blur is to *stop down*, i.e., shrink the aperture. However, this also reduces the amount of light reaching the sensor, making the image noisier. Moreover, stopping down is impossible on fixed-aperture cameras, such as those in most smartphones. More sophisticated techniques fall into two categories. First are techniques that add extra hardware (e.g., coded apertures [46], specialized lenses [47, 15]), and thus are impractical to deploy at large scale or across already available cameras. Second are *focus stacking* techniques [76] that capture multiple images at different focus distances, and fuse them into an all-in-focus image. These techniques require long capture times, and thus are applicable only to static scenes.

Ideally, defocus blur removal should be done using data from a single capture. Unfortunately, in conventional cameras, this task is fundamentally ill-posed: a captured image may have no high-frequency content because either the latent all-in-focus image lacks such frequencies, or they are removed by defocus blur. Knowing the *defocus map*, i.e., the spatially-varying amount of defocus blur, can help simplify blur removal. However, determining the defocus map from a single image is closely-related to monocular depth estimation, which is a challenging problem in its own right. Even if the defocus map were known, recovering an all-in-focus image is still an ill-posed problem, as it requires hallucinating the missing high frequency content.

Dual-pixel (DP) sensors are a recent innovation that makes it easier to solve both the defocus map estimation and defocus blur removal problems, with data from a single capture. Camera manufacturers have introduced such sensors to many DSLR and smartphone cameras to improve autofocus [2, 36]. Each pixel on a DP sensor is split into two halves, each capturing light from half of the main lens' aperture, yielding two sub-images per exposure (Fig. 1). These can be thought of as a two-sample lightfield [61], and their sum is equivalent to the image captured by a regular sensor.

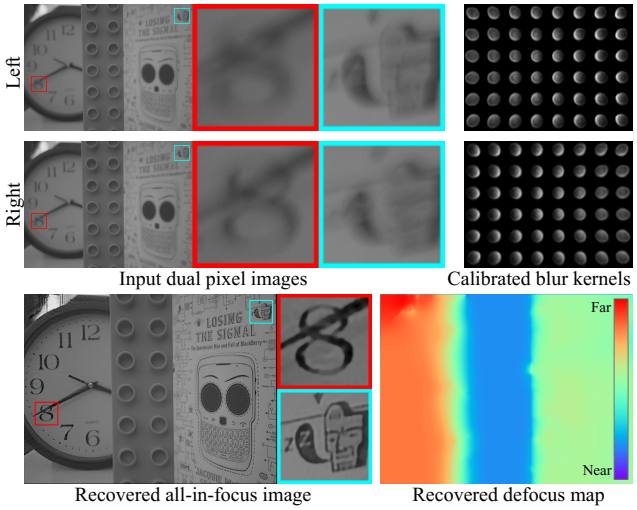


Figure 1: Given left and right dual-pixel (DP) images and corresponding spatially-varying blur kernels, our method jointly estimates an all-in-focus image and defocus map.

*Work primarily done when Shumian Xin was an intern at Google.

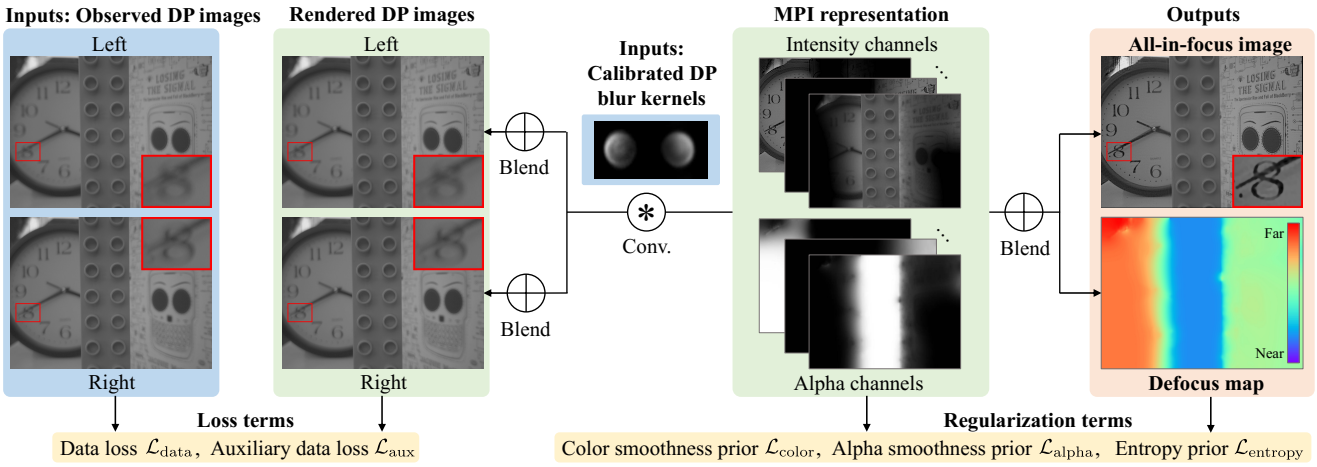


Figure 2: Overview of our proposed method. We use input left and right DP images to fit a multiplane image (MPI) scene representation, consisting of a set of fronto-parallel layers. Each layer is an intensity-alpha image containing the in-focus scene content at the corresponding depth. The MPI can output the all-in-focus image and the defocus map by blending all layers. It can also render out-of-focus images, by convolving each layer with pre-calibrated blur kernels for the left and right DP views, and then blending. We optimize the MPI by minimizing a regularized loss comparing rendered and input images.

The two sub-images have different half-aperture-shaped defocus blur kernels; these are additionally spatially-varying due to optical imperfections such as vignetting or field curvature in lenses, especially for cheap smartphone lenses.

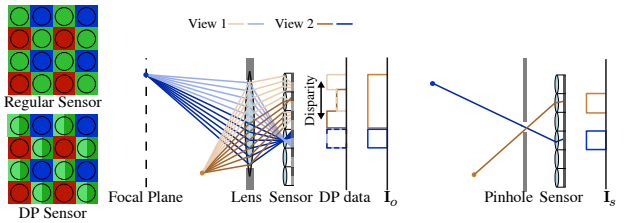
We propose a method to simultaneously recover the defocus map and all-in-focus image from a single DP capture. Specifically, we perform a one-time calibration to determine the spatially-varying blur kernels for the left and right DP images. Then, given a single DP image, we optimize a *multiplane image* (MPI) representation [77, 91] to best explain the observed DP images using the calibrated blur kernels. An MPI is a layered representation that accurately models occlusions, and can be used to render both defocused and all-in-focus images, as well as produce a defocus map. As solving for the MPI from two DP images is under-constrained, we introduce additional priors and show their effectiveness via ablation studies. Further, we show that in the presence of image noise, standard optimization has a bias towards underestimating the amount of defocus blur, and we introduce a bias correction term. Our method does not require large amounts of training data, save for a one-time calibration, and outperforms prior art on both defocus map estimation and blur removal, when tested on images captured using a consumer smartphone camera. We make our implementation and data publicly available [85].

2. Related Work

Depth estimation. Multi-view depth estimation is a well-posed and extensively studied problem [30, 71]. By contrast, single-view, or *monocular*, depth estimation is ill-posed. Early techniques attempting to recover depth from a single image typically relied on additional cues, such as silhouettes, shading, texture, vanishing points, or data-driven supervision [5, 7, 10, 13, 29, 37, 38, 42, 44, 51, 67, 70, 72]. The use of deep neural networks trained on large RGBD datasets [17, 22, 50, 52, 69, 74] significantly improved the performance of data-driven approaches, motivating approaches that use synthetic data [4, 28, 56, 60, 92], self-supervised training [23, 25, 26, 39, 54, 90], or multiple data sources [18, 66]. Despite these advances, producing high-quality depth from a single image remains difficult, due to the inherent ambiguities of monocular depth estimation.

Recent works have shown that DP data can improve monocular depth quality, by resolving some of these ambiguities. Wadhwa *et al*. [82] applied classical stereo matching methods to DP views to compute depth. Punnappurath *et al*. [64] showed that explicitly modeling defocus blur during stereo matching can improve depth quality. However, they assume that the defocus blur is spatially invariant and symmetric between the left and right DP images, which is not true in real smartphone cameras. Depth estimation with DP images has also been used as part of reflection removal algorithms [65]. Garg *et al*. [24] and Zhang *et al*. [87] trained neural networks to output depth from DP images, using a captured dataset of thousands of DP images and ground truth depth maps [3]. The resulting performance improvements come at a significant data collection cost.

Focus or defocus has been used as a cue for monocular depth estimation prior to these DP works. Depth from defocus techniques [19, 63, 78, 84] use two differently-focused images with the same viewpoint, whereas depth from focus techniques use a dense focal stack [27, 33, 76]. Other monocular depth estimation techniques use defocus cues as supervision for training depth estimation networks [75], use a coded aperture to estimate depth from one [46, 81, 89] or two captures [88], or estimate a defocus map using syn-



(a) Sensors (b) Optics for Lens w/ Aperture (c) Optics for Pinhole Lens

Figure 3: A regular sensor and a DP sensor (a) where each green pixel is split into two halves. For a finite aperture lens (b), an in-focus scene point produces overlapping DP images, whereas an out-of-focus point produces shifted DP images. Adding the two DP images yields the image that would have been captured by a regular sensor. (c) shows the corresponding pinhole camera where all scene content is in focus. Ignoring occlusions, images in (b) can be generated from the image in (c) by applying a depth-dependent blur.

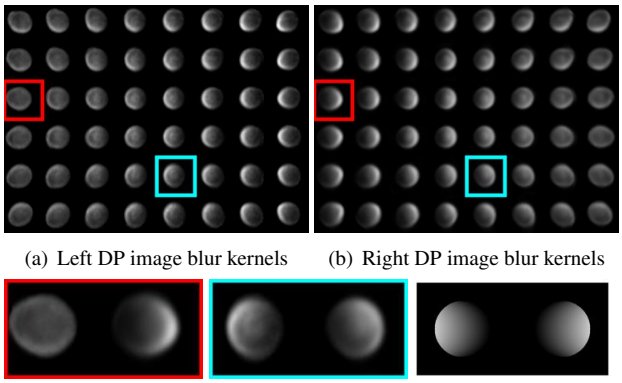
thetic data [45]. Lastly, some binocular stereo approaches also explicitly account for defocus blur [12, 49]; compared to depth estimation from DP images, these approaches assume different focus distances for the two views.

Defocus deblurring. Besides depth estimation, measuring and removing defocus blur is often desirable to produce sharp all-in-focus images. Defocus deblurring techniques usually estimate either a depth map or an equivalent defocus map as a first processing stage [14, 40, 62, 73]. Some techniques modify the camera hardware to facilitate this stage. Examples include inserting patterned occluders in the camera aperture to make defocus scale selection easier [46, 81, 89, 88]; or sweeping through multiple focal settings within the exposure to make defocus blur spatially uniform [59]. Once a defocus map is available, a second deblurring stage employs non-blind deconvolution methods [46, 21, 43, 83, 57, 86] to remove the defocus blur.

Deep learning has been successfully used for defocus deblurring as well. Lee *et al*. [45] train neural networks to regress to defocus maps, that are then used to deblur. Abuolaim and Brown [1] extend this approach to DP data, and train a neural network to directly regress from DP images to all-in-focus images. Their method relies on a dataset of pairs of wide and narrow aperture images captured with a DSLR, and may not generalize to images captured on smartphone cameras, which have very different optical characteristics. Such a dataset is impossible to collect on smartphone cameras with fixed aperture lenses. In contrast to these prior works, our method does not require difficult-to-capture large datasets. Instead, it uses an accurate model of the defocus blur characteristics of DP data, and simultaneously solves for a defocus map and an all-in-focus image.

3. Dual-Pixel Image Formation

We begin by describing image formation for a regular and a dual-pixel (DP) sensor, to relate the defocus map and the all-in-focus image to the captured image. For this, we consider a camera imaging a scene with two points, only one of which is in focus (Fig. 3(b)). Rays emanating from the in-focus point (blue) converge on a single pixel, creating a sharp image. By contrast, rays from the out-of-focus point (brown) fail to converge, creating a blurred image.



(a) Left DP image blur kernels (b) Right DP image blur kernels

(c) (d) (e) Kernels from [64]

Figure 4: Calibrated blur kernels (a) and (b) for the left and right DP images. (c) and (d) show example pairs of left and right kernels marked in red and cyan. Compared to the parametric kernels (e) from [64], calibrated kernels are spatially-varying, not circular, and not left-right symmetric.

If we consider a lens with an infinitesimally-small aperture (i.e., a pinhole camera), only rays that pass through its center strike the sensor, and create a sharp all-in-focus image $\mathbf{I}_s$ (Fig. 3(c)). Under the thin lens model, the blurred image $\mathbf{I}_o$ of the out-of-focus point equals blurring $\mathbf{I}_s$ with a depth-dependent kernel $\boldsymbol{k}_d$, shaped as a d-scaled version of the aperture—typically a circular disc of radius $d = A + {}^{B}/_{Z}$, where Z is the point depth, and A and B are lens-dependent constants [24]. Therefore, the per-pixel signed kernel radius d, termed the *defocus map* $\mathbf{D}$, is a linear function of inverse depth, thus a proxy for the depth map. Given the defocus map $\mathbf{D}$, and ignoring occlusions, the sharp image $\mathbf{I}_s$ can be recovered from the captured image $\mathbf{I}_o$ using non-blind deconvolution. In practice, recovering either the defocus map $\mathbf{D}$ or the sharp image $\mathbf{I}_s$ from a single image $\mathbf{I}_o$ is ill-posed, as multiple $(\mathbf{I}_s, \mathbf{D})$ combinations produce the same image $\mathbf{I}_o$. Even when the defocus map $\mathbf{D}$ is known, determining the sharp image $\mathbf{I}_s$ is still ill-posed, as blurring irreversibly removes image frequencies.

DP sensors make it easier to estimate the defocus map. In DP sensors (Fig. 3(a)), each pixel is split into two halves, each collecting light from the corresponding half of the lens aperture (Fig. 3(b)). Adding the two half-pixel, or DP, images $\mathbf{I}_o^l$ and $\mathbf{I}_o^r$ produces an image equivalent to that captured by a regular sensor, i.e., $\mathbf{I}_o = \mathbf{I}_o^l + \mathbf{I}_o^r$. Furthermore, DP images are identical for an in-focus scene point, and shifted versions of each other for an out-of-focus point. The amount of shift, termed *DP disparity*, is proportional to the

blur size, and thus provides an alternative for defocus map estimation. In addition to facilitating the estimation of the defocus map $\mathbf{D}$, having two DP images instead of a single image provides additional constraints for recovering the underlying sharp image $\mathbf{I}_s$. Utilizing these constraints requires knowing the blur kernel shapes for the two DP images.

Blur kernel calibration. As real lenses have spatially-varying kernels, we calibrate an 8×6 grid of kernels. To do this, we fix the focus distance, capture a regular grid of circular discs on a monitor screen, and solve for blur kernels for left and right images independently using a method similar to Mannan and Langer [55]. When solving for kernels, we assume that they are normalized to sum to one, and calibrate separately for vignetting: we average left and right images from six captures of a white diffuser, using the same focus distance as above, to produce left and right vignetting patterns W_l and W_r. We refer to the supplement for details.

We show the calibrated blur kernels in Fig. 4. We note that these kernels deviate significantly from parametric models derived by extending the thin lens model to DP image formation [64]. In particular, the calibrated kernels are spatially-varying, not circular, and not symmetric.

4. Proposed Method

The inputs to our method are two single-channel DP images, and calibrated left and right blur kernels. We correct for vignetting using W_l and W_r, and denote the two vignetting-corrected DP images as $\mathbf{I}_o^l$ and $\mathbf{I}_o^r$, and their corresponding blur kernels at a certain defocus size d as $\boldsymbol{k}_d^l$ and $\boldsymbol{k}_d^r$, respectively. We assume that blur kernels at a defocus size d' can be obtained by scaling by a factor d'/d [64, 88]. Our goal is to optimize for the multiplane image (MPI) representation that best explains the observed data, and use it to recover the latent all-in-focus image $\hat{\mathbf{I}}_s$ and defocus map $\hat{\mathbf{D}}$. We first introduce the MPI representation, and show how to render defocused images from it. We then formulate an MPI optimization problem, and detail its loss function.

4.1. Multiplane Image (MPI) Representation

We model the scene using the MPI representation, previously used primarily for view synthesis [80, 91]. MPIs discretize the 3D space into N fronto-parallel planes at fixed depths (Fig. 5). We select depths corresponding to linearly-changing defocus blur sizes $[d_1, \ldots, d_N]$. Each MPI plane is an intensity-alpha image of the in-focus scene that consists of an intensity channel $\boldsymbol{c}_i$ and an alpha channel $\boldsymbol{\alpha}_i$.

All-in-focus image compositing. Given an MPI, we composite the sharp image using the *over* operator [53]: we sum all layers weighted by the transmittance of each layer $\boldsymbol{t}_i$,

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

Defocused image rendering. Given the left and right blur kernels $\boldsymbol{k}_{d_i}^{\{l,r\}}$ for each layer, we render defocused images by convolving each layer with its corresponding kernel, then compositing the blurred layers as in Eq. (1):

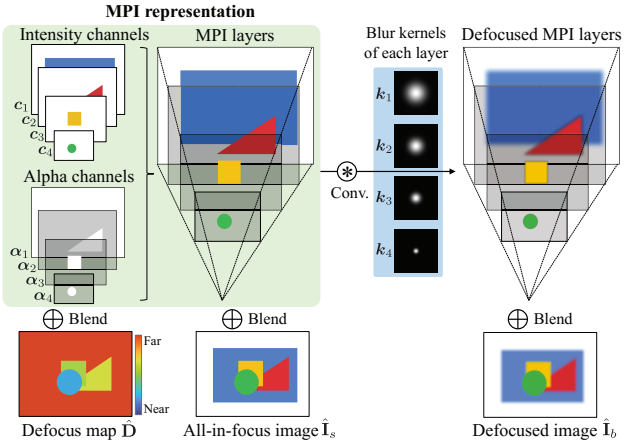


Figure 5: The multiplane image (MPI) representation consists of discrete fronto-parallel planes where each plane contains intensity data and an alpha channel. We use it to recover the defocus map, the all-in-focus image, and render a defocused image according to a given blur kernel.

$$\hat{\mathbf{I}}_b^{\{l,r\}} = \sum_{i=1}^{N} \left[\left(\boldsymbol{k}_{d_i}^{\{l,r\}} * (\boldsymbol{c}_i \boldsymbol{\alpha}_i) \right) \odot \prod_{j=i+1}^{N} \left(\mathbf{1} - \boldsymbol{k}_{d_j}^{\{l,r\}} * \boldsymbol{\alpha}_j \right) \right] , \quad (2)$$

where $*$ denotes convolution. In practice, we scale the calibrated spatially-varying left and right kernels by the defocus size d_i, and apply the scaled spatially-varying blur to each intensity-alpha image $\boldsymbol{c}_i \boldsymbol{\alpha}_i$. We note that we render left and right views from a single MPI, but with different kernels.

4.2. Effect of Gaussian Noise on Defocus Estimation

Using Eq. (2), we can optimize for the MPI that minimizes the L_2-error $||\hat{\mathbf{I}}_b^{\{l,r\}} - \mathbf{I}_o^{\{l,r\}}||_2^2$ between rendered images $\hat{\mathbf{I}}_b^{\{l,r\}}$ and observed DP images $\mathbf{I}_o^{\{l,r\}}$. Here we show that, in the presence of noise, this optimization is biased toward smaller defocus sizes, and we correct for this bias.

Assuming additive white Gaussian noise $\mathbf{N}^{\{l,r\}}$ distributed as $\mathcal{N}(0, \sigma^2)$, we can model DP images as:

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

where $\mathbf{I}_b^{\{l,r\}}$ are the latent noise-free images. For simplicity, we assume for now that all scene content lies on a single fronto-parallel plane with ground truth defocus size $d^\star$. Then, using frequency domain analysis similar to Zhou *et al*. [88], we prove in the supplement that for a defocus size hypothesis d_i, the expected negative log-energy function corresponding to the MAP estimate of the MPI is:

$$E(d_i | \boldsymbol{K}_{d^\star}^{\{l,r\}}, \sigma) = \sum_f C_1(\boldsymbol{K}_{d_i}^{\{l,r\}}, \sigma, \boldsymbol{\Phi}) \left| \boldsymbol{K}_{d^\star}^l \boldsymbol{K}_{d_i}^r - \boldsymbol{K}_{d^\star}^r \boldsymbol{K}_{d_i}^l \right|^2$$
$$+ \sigma^2 \sum_f \left[\frac{|\boldsymbol{K}_{d^\star}^l|^2 + |\boldsymbol{K}_{d^\star}^r|^2 + \sigma^2 |\boldsymbol{\Phi}|^2}{|\boldsymbol{K}_{d_i}^l|^2 + |\boldsymbol{K}_{d_i}^r|^2 + \sigma^2 |\boldsymbol{\Phi}|^2} \right] + C_2(\sigma) , \quad (4)$$

where $\boldsymbol{K}_{d_i}^{\{l,r\}}$ and $\boldsymbol{K}_{d^\star}^{\{l,r\}}$ are the Fourier transforms of kernels $\boldsymbol{k}_{d_i}^{\{l,r\}}$ and $\boldsymbol{k}_{d^\star}^{\{l,r\}}$ respectively, $\boldsymbol{\Phi}$ is the inverse spectral power distribution of natural images, and the summation is over all frequencies. We would expect the loss to be minimized when $d_i = d^\star$. The first term measures the inconsistency between the hypothesized blur kernel d_i and the true kernel $d^\star$, and is indeed minimized when $d_i = d^\star$. However, the second term depends on the noise variance and decreases as $|d_i|$ decreases. This is because, for a normalized blur kernel ($||\boldsymbol{k}_{d_i}^{\{l,r\}}||_1 = 1$), as the defocus kernel size $|d_i|$ decreases, its power spectrum $||\boldsymbol{K}_{d_i}^{\{l,r\}}||_2$ increases. This suggests that white Gaussian noise in input images results in a bias towards smaller blur kernels. To account for this bias, we subtract an approximation of the second term, which we call the *bias correction term*, from the optimization loss:

$$\mathcal{B}\left(d_i|\boldsymbol{K}_{d^\star}^{\{l,r\}},\sigma\right) \approx \sigma^2 \sum_f \frac{\sigma^2|\boldsymbol{\Phi}|^2}{\left|\boldsymbol{K}_{d_i}^{l}\right|^2+\left|\boldsymbol{K}_{d_i}^{r}\right|^2+\sigma^2|\boldsymbol{\Phi}|^2}\,. \quad (5)$$

We ignore the terms containing ground truth $d^\star$, as they are significant only when $d^\star$ is itself small, i.e., the bias favors the true kernels in that case. In an MPI with multiple layers associated with defocus sizes $[d_1, \ldots, d_N]$, we subtract per-layer constants $\mathcal{B}(d_i)$ computed using Eq. (5).

We note that we use a Gaussian noise model to make analysis tractable, but captured images have mixed Poisson-Gaussian noise [31]. In practice, we found it beneficial to additionally denoise the input images using burst denoising [32]. However, there is residual noise even after denoising, and we show in Sec. 5.1 that our bias correction term still improves performance. An interesting future research direction is using a more accurate noise model to derive a better bias estimate and remove the need for any denoising.

4.3. MPI Optimization

We seek to recover an MPI $\{\boldsymbol{c}_i, \boldsymbol{\alpha}_i\}, i \in [1, \ldots, N]$ such that defocused images rendered from it using the calibrated blur kernels are close to the input images. But minimizing only a reconstruction loss is insufficient: this task is ill-posed, as there exists an infinite family of MPIs that all exactly reproduce the input images. As is common in defocus deblurring [46], we regularize our optimization:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{intensity}} + \mathcal{L}_{\text{alpha}} + \mathcal{L}_{\text{entropy}}\,, \quad (6)$$

where $\mathcal{L}_{\text{data}}$ is a bias-corrected data term that encourages rendered images to resemble input images, $\mathcal{L}_{\text{aux}}$ is an auxiliary data term applied to each MPI layer, and the remaining are regularization terms. We discuss all terms below.

Bias-corrected data loss. We consider the Charbonnier [11] loss function $\ell(x) = \sqrt{x^2/\gamma^2 + 1}$, and define a bias-corrected version as $\ell_{\mathcal{B}}(x, \mathcal{B}) = \sqrt{(x^2 - \mathcal{B})/\gamma^2 + 1}$, where we choose the scale parameter $\gamma = 0.1$ [6] . We use this loss function to form a data loss penalizing the difference between left and right input and rendered images as:

$$\mathcal{L}_{\text{data}} = \sum_{x,y} \ell_{\mathcal{B}}\left(\hat{\mathbf{I}}_b^{\{l,r\}}(x,y) - \mathbf{I}_o^{\{l,r\}}(x,y), \mathcal{B}_{\text{all}}^{\{l,r\}}\right)\,, \quad (7)$$

$$\mathcal{B}_{\text{all}}^{\{l,r\}} = \sum_{i=1}^{N}\left[\boldsymbol{k}_{d_i}^{\{l,r\}} * \boldsymbol{\alpha}_i \prod_{j=i+1}^{N}\left(1 - \boldsymbol{k}_{d_j}^{\{l,r\}} * \boldsymbol{\alpha}_j\right)\right]\mathcal{B}(d_i)\,. \quad (8)$$

We compute the total bias correction $\mathcal{B}_{\text{all}}^{\{l,r\}}$ as the sum of all bias correction terms of each layer, weighted by the corresponding defocused transmittance. Eq. (8) is equivalent to Eq. (2) where we replace each MPI layer's intensity channel $\boldsymbol{c}_i$ with a constant bias correction value $\mathcal{B}(d_i)$. To compute $\mathcal{B}(d_i)$ from Eq. (5), we empirically set the variance to $\sigma^2 = 5 \cdot 10^{-5}$, and use a constant inverse spectral power distribution $|\boldsymbol{\Phi}|^2 = 10^2$, following previous work [79].

Auxiliary data loss. In most real-world scenes, a pixel's scene content should be on a single layer. However, because the compositing operator of Eq. (2) forms a weighted sum of all layers, $\mathcal{L}_{\text{data}}$ can be small even when scene content is smeared across multiple layers. To discourage this, we introduce a per-layer auxiliary data loss on each layer's intensity weighted by the layer's blurred transmittance:

$$\mathcal{L}_{\text{aux}} = \sum_{x,y,i}\left(\boldsymbol{k}_{d_i}^{\{l,r\}} * \boldsymbol{t}_i(x,y)\right) \odot \ell_{\mathcal{B}}\left(\boldsymbol{k}_{d_i}^{\{l,r\}} * \boldsymbol{c}_i(x,y) - \mathbf{I}_o^{\{l,r\}}(x,y), \mathcal{B}(d_i)\right)\,, \quad (9)$$

where $\odot$ denotes element-wise multiplication. This auxiliary loss resembles the data synthesis loss of Eq. (7), except that it is applied to each MPI layer separately.

Intensity smoothness. Our first regularization term encourages smoothness for the all-in-focus image and the MPI intensity channels. For an image $\mathbf{I}$ with corresponding edge map $\boldsymbol{E}$, we define an edge-aware smoothness based on total variation $\boldsymbol{V}(\cdot)$, similar to Tucker and Snavely [80]:

$$\boldsymbol{V}_{\boldsymbol{E}}(\mathbf{I}, \boldsymbol{E}) = \ell(\boldsymbol{V}(\mathbf{I})) + (1 - \boldsymbol{E}) \odot \ell(\boldsymbol{V}(\mathbf{I}))\,, \quad (10)$$

where $\ell(\cdot)$ is the Charbonnier loss. We refer to the supplement for details on $\boldsymbol{E}$ and $\boldsymbol{V}(\cdot)$. Our smoothness prior on the all-in-focus image and MPI intensity channels is:

$$\mathcal{L}_{\text{intensity}} = \sum_{x,y} \boldsymbol{V}_{\boldsymbol{E}}\left(\hat{\mathbf{I}}_s, \boldsymbol{E}\left(\hat{\mathbf{I}}_s\right)\right) + \sum_{x,y,i} \boldsymbol{V}_{\boldsymbol{E}}\left(\boldsymbol{t}_i\boldsymbol{c}_i, \boldsymbol{E}\left(\boldsymbol{t}_i\boldsymbol{c}_i\right)\right)\,. \quad (11)$$

Alpha and transmittance smoothness. We use an additional smoothness regularizer on all alpha channels and transmittances (sharpened by computing their square root), by encouraging edge-aware smoothness according to the total variation of the all-in-focus image:

$$\mathcal{L}_{\text{alpha}} = \sum_{x,y,i}\left[\boldsymbol{V}_{\boldsymbol{E}}\left(\sqrt{\boldsymbol{\alpha}_i}, \boldsymbol{E}\left(\hat{\mathbf{I}}_s\right)\right) + \boldsymbol{V}_{\boldsymbol{E}}\left(\sqrt{\boldsymbol{t}_i}, \boldsymbol{E}\left(\hat{\mathbf{I}}_s\right)\right)\right]\,. \quad (12)$$

Alpha and transmittance entropy. The last regularizer is a collision entropy penalty on alpha channels and transmittances. Collision entropy, defined for a vector $\boldsymbol{x}$ as

$S(\boldsymbol{x}) = -\log ||\boldsymbol{x}||_2^2 / ||\boldsymbol{x}||_1^2$, is a special case of Renyi entropy [68], and we empirically found it to be better than Shannon entropy for our problem. Minimizing collision entropy encourages sparsity: $S(\boldsymbol{x})$ is minimum when all but one elements of $\boldsymbol{x}$ are 0, which in our case encourages scene content to concentrate on a single MPI layer, rather than spread across multiple layers. Our entropy loss is:

$$\mathcal{L}_{\text{entropy}} = \sum_{x,y} S\left(\left[\sqrt{\alpha_2}(x,y), \ldots, \sqrt{\alpha_N}(x,y)\right]^{\mathrm{T}}\right) + \sum_{x,y} S\left(\left[\sqrt{t_1}(x,y), \ldots, \sqrt{t_N}(x,y)\right]^{\mathrm{T}}\right). \quad (13)$$

We extract the alpha channels and transmittances of each pixel (x, y) from all MPI layers, compute their square root for sharpening, compute a per-pixel entropy, and average these entropies across all pixels. When computing entropy on alpha channels, we skip the farthest MPI layer, because we assume that all scene content ends at the farthest layer, and thus force this layer to be opaque ($\alpha_1 = 1$).

5. Experiments

We capture a new dataset, and use it to perform qualitative and quantitative comparisons with other state of the art defocus deblurring and defocus map estimation methods. The project website [85] includes an interactive HTML viewer [8] to facilitate comparisons across our full dataset.

Data collection. Even though DP sensors are common, to the best of our knowledge, only two camera manufacturers provide an API to read DP images—Google and Canon. However, Canon's proprietary software applies an unknown scene-dependent transform to DP data. Unlike supervised learning-based methods [1] that can learn to account for this transform, our loss function requires raw sensor data. Hence, we collect data using the Google Pixel 4 smartphone, which allows access to the raw DP data [16].

The Pixel 4 captures DP data only in the green channel. To compute ground truth, we capture a focus stack with 36 slices sampled uniformly in diopter space, where the closest focus distance corresponds to the distance we calibrate for, 13.7 cm, and the farthest to infinity. Following prior work [64], we use the commercial Helicon Focus software [35] to process the stacks and generate ground truth sharp images and defocus maps, and we manually correct holes in the generated defocus maps. Still, there are image regions that are difficult to manually inpaint, e.g., near occlusion boundaries or curved surfaces. We ignore such regions when computing quantitative metrics. We capture a total of 17 scenes, both indoors and outdoors. Similar to Garg *et al.* [24], we centrally crop the DP images to 1008×1344. We refer to the supplement for more details. Our dataset is available at the project website [85].

5.1. Results

We evaluate our method on both defocus deblurring and depth-from-defocus tasks. We use $N = 12$ MPI layers for all scenes in our dataset. We manually determine the kernel sizes of the front and back layers, and evenly distribute layers in diopter space. Each optimization runs for 10,000 iterations with Adam [41], and takes 2 hours on an Nvidia Titan RTX GPU. We gradually decrease the global learning rate from 0.3 to 0.1 with exponential decay. Our JAX [9] implementation is available at the project website [85].

We compare to state-of-the-art methods for defocus deblurring (DPDNet [1], Wiener deconvolution [79, 88]) and defocus map estimation (DP stereo matching [82], supervised learning from DP views [24], DP defocus estimation based on kernel symmetry [64], Wiener deconvolution [79, 88], DMENet [45]). For methods that take a single image as input, we use the average of the left and right DP images. We also provide both the original and vignetting corrected DP images as inputs, and report the best result. We show quantitative results in Tab. 1 and qualitative results in Figs. 6 and 7. For the defocus map, we use the affine-invariant metrics from Garg *et al.* [24]. Our method achieves the best quantitative results on both tasks.

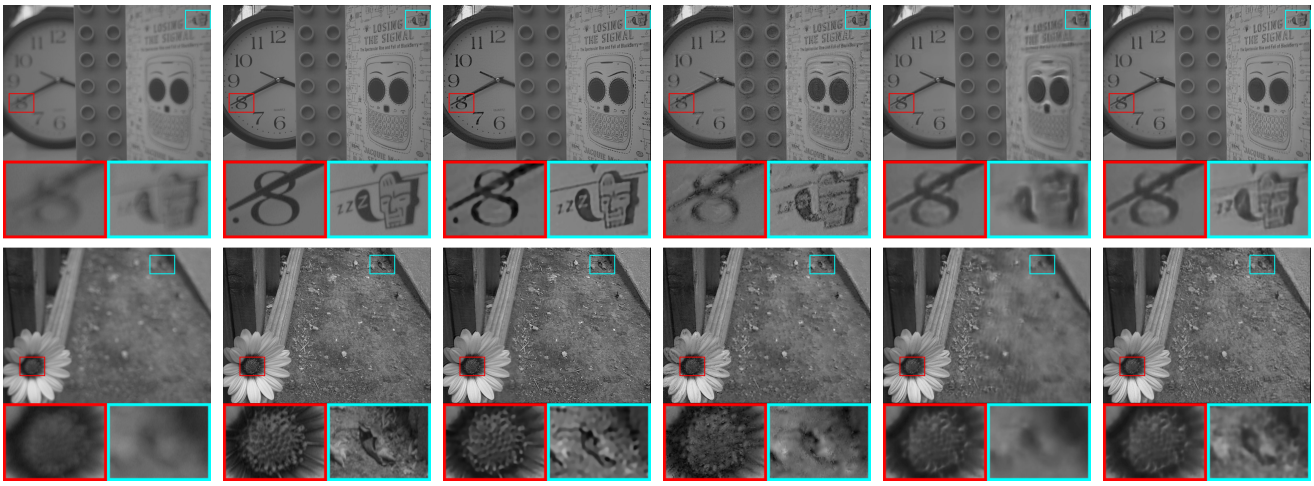
Defocus deblurring results. Despite the large amount of blur in the input DP images, our method produces deblurred results with high-frequency details that are close to the ground truth (Fig. 6). DPDNet makes large errors as it is trained on Canon data and does not generalize. We improve the accuracy of DPDNet by providing vignetting corrected images as input, but its accuracy is still lower than ours.

Defocus map estimation results. Our method produces defocus maps that are closest to the ground truth (Fig. 7), especially on textureless regions, such as the toy and clock in the first scene. Similar to [64], depth accuracy near edges can be improved by guided filtering [34] as shown in Fig. 7(d).

Ablation studies. We investigate the effect of each loss function term by removing them one at a time. Quantitative results are in Tab. 2, and qualitative comparisons in Fig. 8.

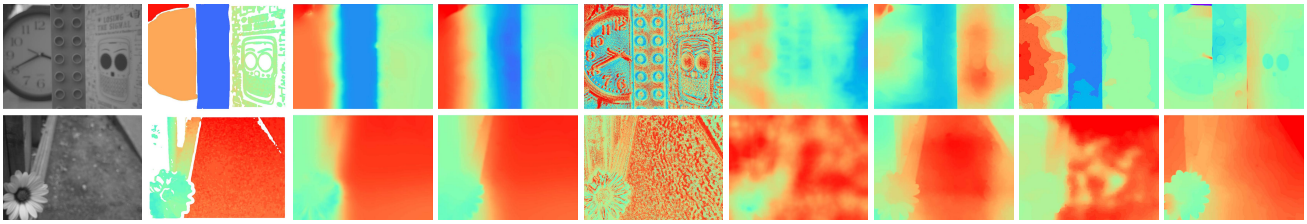
Our full pipeline has the best overall performance in recovering all-in-focus images and defocus maps. $\mathcal{L}_{\text{intensity}}$ and $\mathcal{L}_{\text{alpha}}$ strongly affect the smoothness of all-in-focus images and defocus maps, respectively. Without $\mathcal{L}_{\text{entropy}}$ or $\mathcal{L}_{\text{aux}}$, even though recovered all-in-focus images are reasonable, scene content is smeared across multiple MPI layers, leading to incorrect defocus maps. Finally, without the bias correction term $\mathcal{B}$, defocus maps are biased towards smaller blur radii, especially in textureless areas where noise is more apparent, e.g., the white clock area.

Results on Data from Abuolaim and Brown [1]. Even though Abuolaim and Brown [1] train their model on data from a Canon camera, they also capture Pixel 4 data for qualitative tests. We run our method on their Pixel 4 data, using the calibration from our device, and show that our re-



(a) Input image (b) GT all-in-focus image (c) Ours (d) Wiener deconv. [88] (e) DPDNet [1] (Orig. Input) (f) DPDNet [1]

Figure 6: Qualitative comparisons of various defocus deblurring methods. Input images (a) shown as the average of two DP views, ground truth all-in-focus images (b) computed from focus stacks, recovered all-in-focus images (c) from our method and other methods (d)-(f). We improve the accuracy of DPDNet (e) trained on Canon data by providing vignetting corrected images (f). Our method performs the best in recovering high-frequency details and presents fewer artifacts.



(a) Input image (b) Ground truth (c) Ours (d) Ours w/ GF (e) Wiener [88] (f) DMENet [45] (g) [64] (h) Garg [24] (i) Wadhwa [82]

Figure 7: Qualitative comparisons of defocus map estimation methods. Input images (a) shown as the average of two DP views, ground truth defocus maps (b) from focus stacks with zero confidence pixels in white, our defocus maps (c), and our defocus maps with guided filtering (d), and defocus maps from other methods (f)-(i). Overall, our method produces results that are closest to the ground truth, and correctly handles textureless regions as well.

Method	All-in-focus Image			Defocus Map		
	PSNR ↑	SSIM ↑	MAE ↓	AIWE(1) ↓	AIWE(2) ↓	$1-\lvert\rho_s\rvert$ ↓
Wiener Deconv. [88]	25.806	0.704	0.032	0.156	0.197	0.665
DPDNet [1]	25.591	0.777	0.034	-	-	-
DMENet [45]	-	-	-	0.144	0.183	0.586
Punnappurath *et al*. [64]	-	-	-	0.124	0.161	0.444
Garg *et al*. [24]	-	-	-	0.079	0.102	0.208
Wadhwa *et al*. [82]	-	-	-	0.141	0.177	0.540
Ours	26.692	0.804	0.027	0.047	0.076	0.178
Ours w/ guided filtering	26.692	0.804	0.027	0.059	0.083	0.193

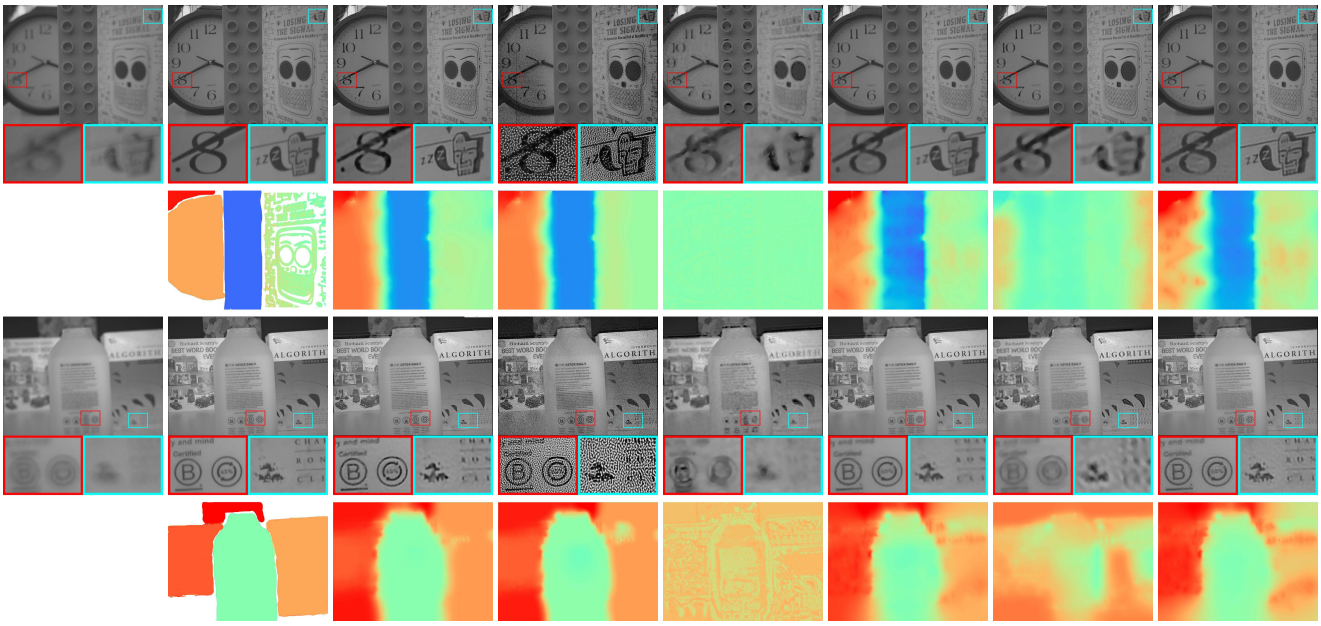
Table 1: Quantitative evaluations of defocus deblurring and defocus map estimation methods on our DP dataset. "-" indicates not applicable. We use the affine-invariant metrics from [24] for defocus map evaluation. Our method achieves the best performance (highlighted in red) in both tasks.

covered all-in-focus image has fewer artifacts (Fig. 9). This demonstrates that our method generalizes well across devices of the same model, even without re-calibration.

6. Discussion and Conclusion

We presented a method that optimizes an MPI scene representation to jointly recover a defocus map and all-in-focus image from a single dual-pixel capture. We showed that image noise introduces a bias in the optimization that, under suitable assumptions, can be quantified and corrected for. We also introduced additional priors to regularize the optimization, and showed their effectiveness via ablation studies. Our method improves upon past work on both defocus map estimation and blur removal, when evaluated on a new dataset we captured with a consumer smartphone camera.

Limitations and future directions. We discuss some limitations of our method, which suggest directions for future research. First, our method does not require a large dataset with ground truth to train on, but still relies on a one-time blur kernel calibration procedure. It would be interesting to explore blind deconvolution techniques [20, 48] that can simultaneously recover the all-in-focus image, defocus map, and unknown blur kernels, thus removing the need for kernel calibration. The development of parametric blur kernel models that can accurately reproduce the features we observed (i.e., spatial variation, lack of symme-



(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 $\mathcal{B}$

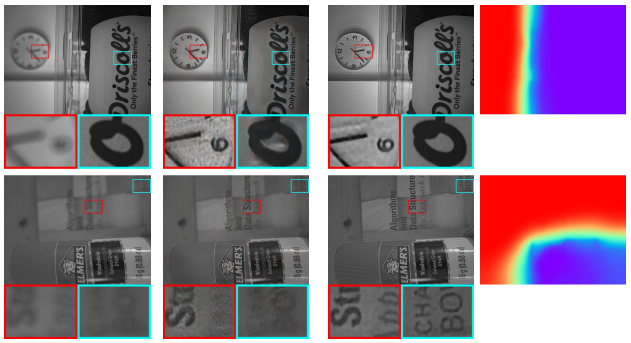
Figure 8: Ablation study. Input images (a), ground truth all-in-focus images, and defocus maps (b) with zero confidence pixels in white, our results (c), and our results with different terms removed one at a time (d)-(h). Removing $\mathcal{L}_{\text{intensity}}$ and $\mathcal{L}_{\text{alpha}}$ strongly affects the smoothness of all-in-focus images and defocus maps respectively. Results without entropy regularization $\mathcal{L}_{\text{entropy}}$, $\mathcal{L}_{\text{aux}}$, or the bias correction $\mathcal{B}$, exhibit more errors in defocus maps on textureless regions (clock).

try, lack of circularity) can facilitate this research direction. Second, the MPI representation discretizes the scene into a set of fronto-parallel depth layers. This can potentially result in discretization artifacts in scenes with continuous depth variation. In practice, we did not find this to be an issue, thanks to the use of the soft-blending operation to synthesize the all-in-focus image and defocus map. Nevertheless, it could be useful to replace the MPI representation with a continuous one, e.g., neural radiance fields [58], to help better model continuously-varying depth. Third, reconstructing an accurate all-in-focus image becomes more difficult as defocus blur increases (e.g., very distant scenes at non-infinity focus) and more high-frequency content is missing from the input image. This is a fundamental limitation shared among all deconvolution techniques. Using powerful data-driven priors to hallucinate the missing high frequency content (e.g., deep-learning-based deconvolution techniques) can help alleviate this limitation. Fourth, the high computational complexity of our technique makes it impractical for real-time operation, especially on resource-constrained devices such as smartphones. Therefore, it is worth exploring optimized implementations.

Acknowledgments. We thank David Salesin and Samuel Hasinoff for helpful feedback. S.X. and I.G. were supported by NSF award 1730147 and a Sloan Research Fellowship.

Method	All-in-focus Image			Defocus Map		
	PSNR ↑	SSIM ↑	MAE ↓	AIWE(1) ↓	AIWE(2) ↓	$1-\lvert\rho_s\rvert$ ↓
Full	26.692	0.804	0.027	0.047	0.076	0.178
No $\mathcal{L}_{\text{intensity}}$	14.882	0.158	0.136	0.047	0.078	0.185
No $\mathcal{L}_{\text{alpha}}$	24.748	0.726	0.037	0.161	0.206	0.795
No $\mathcal{L}_{\text{entropy}}$	27.154	0.819	0.026	0.057	0.085	0.190
No $\mathcal{L}_{\text{aux}}$	26.211	0.768	0.030	0.148	0.190	0.610
No $\mathcal{B}$	26.265	0.790	0.028	0.063	0.092	0.214

Table 2: Quantitative comparisons of ablation studies. We compare the full pipeline with removals of the regularization terms $\mathcal{L}_{\text{alpha}}$, $\mathcal{L}_{\text{intensity}}$ and $\mathcal{L}_{\text{entropy}}$, the auxiliary data loss $\mathcal{L}_{\text{aux}}$, and bias correction term $\mathcal{B}$ respectively. For all ablation experiments, we set the weights on remaining terms to be the same as the ones in the full pipeline. Best and second best results are highlighted in red and orange.



(a) Input from [1] (b) DPDNet [1] (c) Our results

Figure 9: Results on data from [1]. Our method recovers all-in-focus images with fewer artifacts, while using the calibration data from our device.

References

[1] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. *European Conference on Computer Vision*, 2020. 3, 6, 7, 8

[2] Abdullah Abuolaim, Abhijith Punnappurath, and Michael S. Brown. Revisiting autofocus for smartphone cameras. *European Conference on Computer Vision*, 2018. 1

[3] Sameer Ansari, Neal Wadhwa, Rahul Garg, and Jiawen Chen. Wireless software synchronization of multiple distributed cameras. *IEEE International Conference on Computational Photography*, 2019. 2

[4] Amir Atapour-Abarghouei and Toby P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[5] Ruzena Bajcsy and Lawrence Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 1976. 2

[6] Jonathan T. Barron. A general and adaptive robust loss function. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 5

[7] Jonathan T. Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012. 2

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

[9] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 6

[10] Michael Brady and Alan Yuille. An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984. 2

[11] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *IEEE International Conference on Image Processing*, 1994. 5

[12] Ching-Hui Chen, Hui Zhou, and Timo Ahonen. Blur-aware disparity estimation from defocus stereo images. *IEEE/CVF International Conference on Computer Vision*, 2015. 3

[13] Sunghwan Choi, Dongbo Min, Bumsub Ham, Youngjung Kim, Changjae Oh, and Kwanghoon Sohn. Depth analogy: Data-driven approach for single image depth estimation using gradient samples. *IEEE Transactions on Image Processing*, 2015. 2

[14] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *IEEE Transactions on Image Processing*, 2016. 3

[15] Edward R. Dowski and W. Thomas Cathey. Extended depth of field through wave-front coding. *Applied optics*, 1995. 1

[16] Dual pixel capture app. https://github.com/google-research/google-research/tree/master/dual_pixels. 6

[17] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 2014. 2

[18] Jose M. Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Camconvs: camera-aware multi-scale convolutions for single-view depth. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[19] Paolo Favaro. Recovering thin structures via nonlocal-means regularization with application to depth from defocus. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2010. 2

[20] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 2006. 7

[21] D.A. Fish, A.M. Brinicombe, E.R. Pike, and J.G. Walker. Blind deconvolution by means of the Richardson–Lucy algorithm. *Journal of the Optical Society of America A*, 1995. 3

[22] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[23] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *European Conference on Computer Vision*, 2016. 2

[24] 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. 2, 3, 6, 7

[25] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2

[26] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. *IEEE/CVF International Conference on Computer Vision*, 2019. 2

[27] Paul Grossmann. Depth from focus. *Pattern Recognition Letters*, 1987. 2

[28] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. *European Conference on Computer Vision*, 2018. 2

[29] Christian Häne, L'ubor Ladický, and Marc Pollefeys. Direction matters: Depth estimation with a surface normal classifier. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 2

[30] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2

[31] Samuel W. Hasinoff, Frédo Durand, and William T. Freeman. Noise-optimal capture for high dynamic range photography. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2010. 5

[32] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics*, 2016. 5

[33] Caner Hazırbaş, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. *Asian Conference on Computer Vision*, 2018. 2

[34] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *European Conference on Computer Vision*, 2010. 6

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

[36] Charles Herrmann, Richard Strong Bowen, Neal Wadhwa, Rahul Garg, Qiurui He, Jonathan T. Barron, and Ramin Zabih. Learning to autofocus. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[37] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 2005. 2

[38] Berthold K.P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, Massachusetts Institute of Technology, 1970. 2

[39] Huaizu Jiang, Erik G. Learned-Miller, Gustav Larsson, Michael Maire, and Greg Shakhnarovich. Self-supervised depth learning for urban scene understanding. *European Conference on Computer Vision*, 2018. 2

[40] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing*, 2017. 3

[41] Diederick P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 6

[42] Janusz Konrad, Meng Wang, Prakash Ishwar, Chen Wu, and Debargha Mukherjee. Learning-based, automatic 2d-to-3d image and video conversion. *IEEE Transactions on Image Processing*, 2013. 2

[43] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in Neural Information Processing Systems*, 2009. 3

[44] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014. 2

[45] 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*, June 2019. 3, 6, 7

[46] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 2007. 1, 2, 3, 5

[47] Anat Levin, Samuel W. Hasinoff, Paul Green, Frédo Durand, and William T. Freeman. 4D frequency analysis of computational cameras for depth of field extension. *ACM Transactions on Graphics*, 2009. 1

[48] Anat Levin, Yair Weiss, Frédo Durand, and William T. Freeman. Understanding blind deconvolution algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 7

[49] Feng Li, Jian Sun, Jue Wang, and Jingyi Yu. Dual-focus stereo imaging. *Journal of Electronic Imaging*, 2010. 3

[50] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. *IEEE/CVF International Conference on Computer Vision*, 2017. 2

[51] Xiu Li, Hongwei Qin, Yangang Wang, Yongbing Zhang, and Qionghai Dai. DEPT: depth estimation by parameter transfer for single still images. *Asian Conference on Computer Vision*, 2014. 2

[52] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2

[53] Patric Ljung, Jens Krüger, Eduard Groller, Markus Hadwiger, Charles D. Hansen, and Anders Ynnerman. State of the art in transfer functions for direct volume rendering. *Computer Graphics Forum*, 2016. 4

[54] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[55] Fahim Mannan and Michael S. Langer. Blur calibration for depth from defocus. *Conference on Computer and Robot Vision*, 2016. 4

[56] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 2018. 2

[57] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. *European Conference on Computer Vision*, 2014. 3

[58] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *European Conference on Computer Vision*, 2020. 8

[59] Hajime Nagahara, Sujit Kuthirummal, Changyin Zhou, and Shree K. Nayar. Flexible Depth of Field Photography. *European Conference on Computer Vision*, 2008. 3

[60] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[61] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford University, 2005. 1

[62] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 3

[63] Alex Paul Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987. 2

[64] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S. Brown. Modeling defocus-disparity in dual-pixel sensors. *IEEE International Conference on Computational Photography*, 2020. 2, 3, 4, 6, 7

[65] Abhijith Punnappurath and Michael S. Brown. Reflection removal using a dual-pixel sensor. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[66] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[67] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 2

[68] Alfréd Rényi. On measures of entropy and information. *Berkeley Symposium on Mathematical Statistics and Probability*, 1961. 6

[69] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 2

[70] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. *Advances in Neural Information Processing Systems*, 2006. 2

[71] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002. 2

[72] Jianping Shi, Xin Tao, Li Xu, and Jiaya Jia. Break ames room illusion: depth from general single images. *ACM Transactions on Graphics*, 2015. 2

[73] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 3

[74] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *European Conference on Computer Vision*, 2012. 2

[75] Pratul P. Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T. Barron. Aperture supervision for monocular depth estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[76] Supasorn Suwajanakorn, Carlos Hernandez, and Steven M. Seitz. Depth from focus with your mobile phone. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[77] Rick Szeliski and Polina Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 1999. 2

[78] Huixuan Tang, Scott Cohen, Brian Price, Stephen Schiller, and Kiriakos N. Kutulakos. Depth from defocus in the wild. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2

[79] Huixuan Tang and Kiriakos N. Kutulakos. Utilizing optical aberrations for extended-depth-of-field panoramas. *Asian Conference on Computer Vision*, 2012. 5, 6

[80] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 4, 5

[81] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics*, 2007. 2, 3

[82] 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. 2, 6, 7

[83] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 2008. 3

[84] Masahiro Watanabe and Shree K. Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 1998. 2

[85] 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. 2, 6

[86] Jiawei Zhang, Jinshan Pan, Wei-Sheng Lai, Rynson W.H. Lau, and Ming-Hsuan Yang. Learning fully convolutional networks for iterative non-blind deconvolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 3

[87] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du^{2}net: Learning depth estimation from dual-cameras and dual-pixels. *European Conference on Computer Vision*, 2020. 2

[88] Changyin Zhou, Stephen Lin, and Shree K. Nayar. Coded aperture pairs for depth from defocus. *IEEE/CVF International Conference on Computer Vision*, 2009. 2, 3, 4, 6, 7

[89] Changyin Zhou and Shree K. Nayar. What are good apertures for defocus deblurring? *IEEE International Conference on Computational Photography*, 2009. 2, 3

[90] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2

[91] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 2018. 2, 4

[92] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-net: Unsupervised joint learning of depth and flow using cross-task consistency. *European Conference on Computer Vision*, 2018. 2