Lens Parameter Estimation for Realistic Depth of Field Modeling

Dominique Piché-Meunier[1,2], Yannick Hold-Geoffroy[2], Jianming Zhang[2], Jean-François Lalonde[1]
[1]Université Laval, [2]Adobe
https://lvsn.github.io/inversedof/

Abstract

We present a method to estimate the depth of field effect from a single image. Most existing methods related to this task provide either a per-pixel estimation of blur and/or depth. Instead, we go further and propose to use a lens-based representation that models the depth of field using two parameters: the blur factor and focus disparity. Those two parameters, along with the signed defocus representation, result in a more intuitive and linear representation which we solve using a novel weighting network. Furthermore, our method explicitly enforces consistency between the estimated defocus blur, the lens parameters, and the depth map. Finally, we train our deep-learning-based model on a mix of real images with synthetic depth of field and fully synthetic images. These improvements result in a more robust and accurate method, as demonstrated by our state-of-the-art results. In particular, our lens parametrization enables several applications, such as 3D staging for AR environments and seamless object compositing.

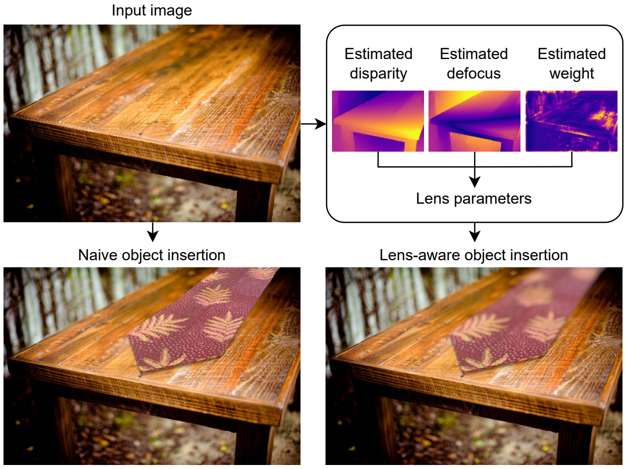

Figure 1: Naively inserting 3D objects in images results in unreasonable compositions in the presence of depth of field blur (bottom-left). Our technique (top-right) estimates physically-based lens parameters that enable lens-aware insertion (bottom-right), where objects appear more realistic.

1. Introduction

Modern cameras are equipped with sophisticated compound lenses whose task is to focus light on the sensor. The lens aperture (or f-number) determines, along with the exposure time, the amount of light captured in a photograph. Adjusting the aperture however has another side effect: a larger aperture (lower f-number) induces a depth-dependent blur, leaving only parts of the image in focus. This depth of field (DoF) effect is often sought by photographers to guide the viewers attention to the subject (e.g., macro shots), or create an artistic look (e.g., tilt-shift photography). Even when not exacerbated by an experienced photographer, some amount of depth-dependent defocus blur is bound to be present in images due to non-zero apertures.

The vast majority of computer vision algorithms ignore this DoF blur and make the simplifying (pinhole) assumption that the image is entirely in focus. While this may be acceptable in some cases, this may create unwanted effects. Consider for example the case of virtual 3D object insertion (fig. 1). It is of paramount importance that the inserted objects be rendered by a virtual camera and lens sharing the same properties as the real ones (fig. 1, bottom-right), as not doing so results in an improbable result (fig. 1, bottom-left).

Blur estimation is a longstanding problem in computer vision, and many techniques estimating defocus blur specifically [43, 30, 23, 52, 28] have been proposed in the literature. However, these non-parametric approaches produce per-pixel estimates, which do not allow advanced image editing tasks such as 3D virtual object insertion as in fig. 1.

In this paper, we propose to go beyond per-pixel defocus blur and argue that reasoning about the camera lens parameters is a more principled approach to understanding the DoF effect in images. We propose what is, to the best of our knowledge, the first method for estimating the camera lens parameters that control the depth of field from a single image. Specifically, we estimate two key lens parameters: the focus disparity and the blur factor (scaled aperture). Our approach first relies on pixel-wise depth and disparity estimation networks. Then, we use a signed disparity blur model, allowing for a more intuitive and linear representation that can recover the desired lens parameters via linear least-squares. Our work goes one step further and proposes a weighting network to guide the attention of the method during the least-squares fitting to increase the accuracy and robustness. We obtain a corrected defocus map by making it explicitly consistent with the estimated lens parameters, contrary to currently available methods which can technically produce any defocus at any depth, yielding globally inconsistent results. Finally, we use the recovered lens parameters to enable novel applications such as 3D virtual object insertion, where objects are automatically blurred by the rendering engine without having to rely on any 2D post-processing. In short, we make the following contributions:

- We present the first method for estimating the lens parameters that control the DoF from a single image: the focus disparity and blur factor. Our method produces globally consistent defocus maps and employs a weighting module to focus its attention on relevant parts of the image;
- We present a comparative analysis of lens parameter estimation from per-pixel maps, favorably comparing our proposed approach to several strong baselines;
- We demonstrate state-of-the-art performance on pixel-wise disparity and defocus blur estimation on images with strong DoF effects compared to recent techniques;
- Recovering lens parameters from a single image enables, for the first time, the realistic insertion of 3D objects in shallow DoF images.

2. Related work

Depth estimation has received a lot of attention in the community over many decades. Early methods performed depth estimation from a single image by solving a graphical model [38, 39, 58] or through retrieval [18]. Later, deep learning based methods were proposed to extract priors optimized for this task directly from data. In their seminal paper, Eigen et al. [7] propose a multi-resolution two-stage method that refines an initially coarse depth estimation. [9] further improves robustness with discrete depth and ordinal regression. Later methods exploit large-scale unannotated datasets for self-supervision, notably utilizing geometric consistency [12], optical flow [51], visual odometry [45], or a combination of masks and multi-scale sampling [13]. Of note, MiDaS [35] proposes to take advantage of multiple datasets with seemingly incompatible annotations by employing a scale-invariant loss. The method was later extended to leverage visual transformers [34]. However, most monocular depth estimation methods ignore lens effects. As a result, they typically fail on images exhibiting strong blur.

Depth from defocus methods recover scene depth using the lens blur effect. Typically, those methods require multiple images, either leveraging a varying focus [42, 47, 8], a structured light system [11], or aperture coding [54]. Some methods estimate the depth from defocus from a single image, using texture cues and defocus [41] extracted with handcrafted features, while [2] proposes to directly estimate the per-pixel depth through defocus using a deep neural network.

Defocus map estimation has first been tackled using handcrafted features in several ways [43, 57, 56, 17, 40, 50, 6, 25], which yielded limited robustness. Defocus maps are often used for deblurring, hence several methods propose to perform both simultaneously [53, 36, 28]. It is also used for other tasks, including image quality assessment [29], defocus magnification [3], and image refocusing [37].

Defocus map estimation has recently significantly improved by taking advantage of deep learning, starting with the work of Park *et al.* [30] who combines both handcrafted multi-scale features with a deep learning model. Later, end-to-end methods were proposed, such as DMENet [23], which is trained on synthetic images and leverages a domain adaptation scheme to bridge the domain gap on real images. Concurrently, [14] proposes to estimate the Point Spread Function (PSF) per pixel, also enabling defocus map estimation. More recently, DID-ANet [28] advances a deblurring method guided by a defocus map estimated as an auxiliary task. DBENet [16] estimates the defocus map by discarding edges due to discontinuities and focusing on texture edges. [36] exploit a light field dataset to train an image restoration network. Joint-Depth-Defocus (JDD) [52] propose jointly estimating depth and defocus from a single image using a physical consistency constraint. We leverage their proposed loss but proceed further by estimating the lens parameters.

Complementary to defocus estimation, work has also been done in defocus synthesis [48, 10, 22]. In particular, BokehMe [31] was recently proposed to generate a plausible DoF effect from a single in-focus image. Here, we make use of the latter as a processing step to obtain training data.

Camera calibration Instead of estimating the dense defocus map, we argue that reasoning on the depth of field as a global lens effect enables several applications such as virtual 3D scene setup and object insertion. Pertuz et al. [32] propose using focus sampling and PSF fitting on a focus sweep of several images to model the camera lens with a single parameter. Inspired by this work, we tackle this task from a single image, using an end-to-end differentiable approach.

3. Method

3.1. Image formation model

In a lens-based optical system, only the scene points at the focal plane (at depth z_f) appear perfectly sharp on the imaging sensor. According to the thin lens approximation and as illustrated in fig. 2, rays incoming from points at any other depth will converge either in front of, or behind the sensor. A point at depth z will therefore project as a circle of

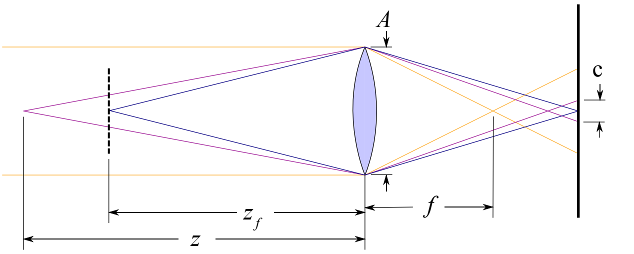

Figure 2: Thin lens approximation used to calculate the size of the circle of confusion c on the sensor as a function of the depth z. This relation depends on the focus depth z_f, the size the aperture A and the focal length f.

diameter c on the sensor, dubbed the "circle of confusion". The relationship between these quantities, the lens aperture A and the focal length f has been thoroughly studied in the literature (e.g., [44, 33, 4]) and can be summarized as

$$c = Af \left| \frac{z - z_f}{z(z_f - f)} \right| \approx Af \left| \frac{z - z_f}{z z_f} \right| . \tag{1}$$

The approximation comes from the hypothesis that $z_f \gg f$. The relationship in eq. (1) provides the amount of blur, as measured by the circle of confusion, for every pixel in the image with known depth z.

This non-linear relationship between camera parameters (A, f, z_f) and pixel quantities (c, z) (fig. 3a) can be simplified by replacing the depth by disparity $d = {}^1\!/_z$:

$$c \approx Af \left| d - d_f \right| , \tag{2}$$

making the relationship linear (fig. 3b). Eq. (2) can be further simplified by using the *signed defocus* c_s, where c_s is negative if $d < d_f$, and positive otherwise. After substituting in eq. (1), the diameter of the circle of confusion is now

$$c_s \approx \kappa \left(d - d_f \right) , \tag{3}$$

where $d_f = {}^1\!/_{z_f}$ is the disparity at the focus plane, or the focus disparity, and $\kappa = Af$ is the blur factor (scaled aperture) (fig. 3c). Note that resolving the scale ambiguity would require estimating the focal length f, which is beyond the scope of this work.

3.2. Approach

An overview of our method to estimate the focus disparity d_f and the blur factor κ from a single image is illustrated in fig. 4. We begin by relying on two networks: 1) a defocus network which estimates the signed defocus map $\hat{\mathbf{C}}_s$, see eq. (3); and 2) a disparity network which outputs the estimated disparity map $\hat{\mathbf{D}}$ from the input image. These two networks are trained using a combination of an L1 loss ℓ_1 and a multi-scale scale-invariant gradient matching loss ℓ_{msg}

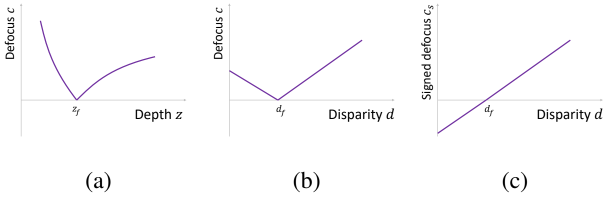

Figure 3: Typical defocus curves for the parametrizations defined by (a) eq. (1), (b) eq. (2) and (c) eq. (3).

(evaluated at four different scales) [24]:

$$\ell_{\text{defocus}} = \lambda_1 \ell_1(\hat{\mathbf{C}}_s, \mathbf{C}_s) + \lambda_2 \ell_{\text{msg}}(\hat{\mathbf{C}}_s, \mathbf{C}_s) , \tag{4}$$

$$\ell_{\text{disp}} = \lambda_3 \ell_1(\hat{\mathbf{D}}, \mathbf{D}) + \lambda_4 \ell_{\text{msg}}(\hat{\mathbf{D}}, \mathbf{D}) , \tag{5}$$

with $\mathbf{D}$ and $\mathbf{C}_s$ denoting the ground truth disparity and signed defocus maps, respectively. In addition, a physical consistency loss [52] helps ensure that the estimated defocus $\hat{\mathbf{C}}_s$ and disparity $\hat{\mathbf{D}}$ are consistent which each other, following eq. (3). Using the ground truth camera parameters, we compute a signed defocus map from the estimated disparity map $\tilde{\mathbf{C}}_s = \kappa(\hat{\mathbf{D}} - d_f)$, as well as a disparity map from the estimated defocus map $\tilde{\mathbf{D}} = {}^{\hat{\mathbf{C}}_s}\!/_{\kappa} + d_f$. Physical consistency between the two networks is then enforced by minimizing

$$\ell_{\text{pc}} = \lambda_5 \ell_1(\tilde{\mathbf{D}}, \mathbf{D}) + \lambda_6 \ell_1(\tilde{\mathbf{C}}_s, \mathbf{C}_s) . \tag{6}$$

The networks are trained using $\ell = \ell_{\text{defocus}} + \ell_{\text{disp}} + \ell_{\text{pc}}$.

The camera parameters $(\tilde{d_f}, \tilde{\kappa})$ are recovered from the outputs of the two previous networks according to eq. (3):

$$(\tilde{d_f}, \tilde{\kappa}) = \underset{d_f, \kappa}{\arg\min} \sum_{i=1}^{N} \left(\hat{\mathbf{C}}_s(i) - \kappa(\hat{\mathbf{D}}(i) - d_f) \right)^2 , \tag{7}$$

which can be solved via linear least squares. Here, $i \in [1, N]$ iterates over the N image pixels. We observed that solving this equation directly is sensitive to outliers and errors in the maps, reducing the accuracy. Intuitively, defocus blur might be more noticeable on heavily textured parts of the image, and we can hypothesize that the estimated defocus map will be more accurate in these regions. They should thus be given more weight in the least-square parameter estimation.

For this purpose, we train an additional weight network to estimate a per-pixel weight that can be used to refine the global linear fit from eq. (7). We train this weighting network while keeping the disparity and defocus networks frozen. It estimates a per-pixel weight map $\hat{\mathbf{W}}$ s.t.

$$(\hat{d}_f, \hat{\kappa}) = \underset{d_f, \kappa}{\arg\min} \sum_{i=1}^{N} \left(\hat{\mathbf{W}}(i) \left(\hat{\mathbf{C}}_s(i) - \kappa(\hat{\mathbf{D}}(i) - d_f) \right) \right)^2 . \tag{8}$$

Since we do not have ground truth for the per-pixel weights, we define the parameter loss:

$$\ell_{\text{param}} = \lambda_7 \ell_1(\hat{\kappa}, \kappa) + \lambda_8 \ell_1(\hat{d}_f, d_f) . \tag{9}$$

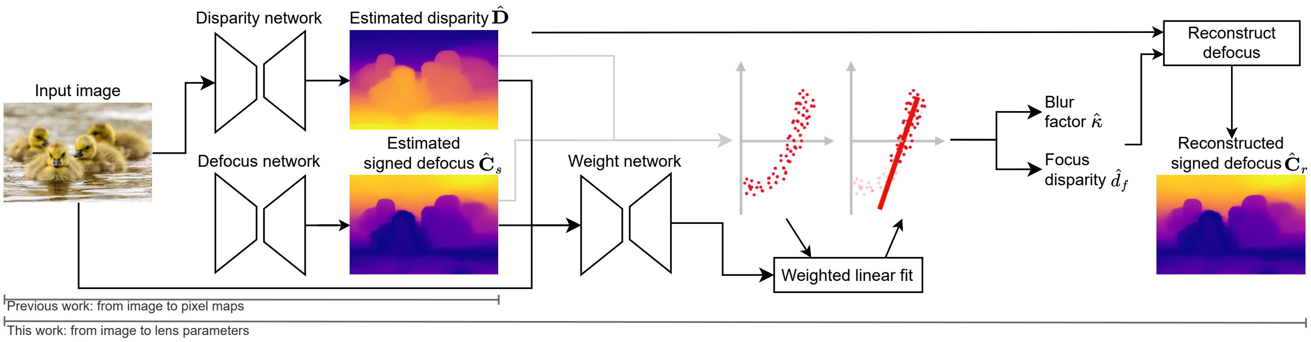
Disparity network
Estimated disparity $\hat{\mathbf{D}}$
Input image
Defocus network
Estimated signed defocus $\hat{\mathbf{C}}_s$
Weight network
Weighted linear fit
Blur factor $\hat{\kappa}$
Focus disparity $\hat{d}_f$
Reconstruct defocus
Reconstructed signed defocus $\hat{\mathbf{C}}_r$
Previous work: from image to pixel maps
This work: from image to lens parameters

Figure 4: Overview of our approach. From a single input image, two networks first estimate pixel-wise disparity and defocus values $\hat{\mathbf{D}}$ and $\hat{\mathbf{C}}_s$ respectively. From these two estimates, we then perform a weighted least squares linear fit, where the weights are given by a third network, to recover the lens parameters (blur factor $\hat{\kappa}$ and focus disparity $\hat{d}_f$) that model the depth of field. We finally use these parameters, along with the disparity map, to get a more accurate reconstructed defocus map estimation.

We also reconstruct a signed defocus map from the estimated disparity and lens parameters $\hat{\mathbf{C}}_r = \hat{\kappa}\left(\hat{\mathbf{D}} - \hat{d}_f\right)$, and consider a loss between this weighted reconstructed defocus map and the ground truth signed defocus:

$$\ell_{\text{recon}} = \lambda_9 \ell_1(\hat{\mathbf{C}}_r, \mathbf{C}_s) + \lambda_{10} \ell_{\text{msg}}(\hat{\mathbf{C}}_r, \mathbf{C}_s)\,, \tag{10}$$

We train the weight network using $\ell_{\text{weight}} = \ell_{\text{param}} + \ell_{\text{recon}}$.

3.3. Architecture and training details

We use the same architecture for all three networks, consisting of a Pyramid Vision Transfomer v2 [46] encoder, and a modified SegFormer [49] decoder head (refer to the supplementary material for more details). For the linear regressions, we find the least-squares solution for the camera parameters by computing the pseudo-inverse. To avoid numerical instability issues and reduce the memory footprint of this step, we fit 100 random subsets of defocus and disparity maps and consider the mean of the obtained estimated parameters.

We use an input resolution of 640 × 640. To make the defocus estimation independent of the input resolution, the defocus network estimates the size of the circle of confusion in relative pixels (pixels normalized by the input width). Finally, because the scale of the disparity is absorbed in the blur factor κ, we normalize the disparity between 0 and 1.

We use the AdamW [27] optimizer to train the network, with a learning rate of 10^{-6}, exponential decay rates of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. We set $\lambda_{1\ldots6} = 1$, $\lambda_7 = 100$, $\lambda_8 = 10$, $\lambda_{9,10} = 1000$. We use a batch size of 8, and train the disparity and defocus modules by alternating between 1000 and 500 iterations resp. on the BokehMe (proprietary) and SynthWorld datasets (see sec. 4)

4. Datasets

To palliate the lack of paired images, depth, and defocus maps, we use a mix of synthetic and semi-synthetic datasets.

Synthetic Synthworld To get synthetic defocused images with exact defocus blur, we generate renders from 3D scenes, using the Blender Cycles physically-based renderer [5]. We use 67 realistic 3D scenes from Evermotion [1], from which we use 56 for training, 19 for validation, and 10 for testing. For each scene, we manually identify several possible camera positions, and randomly sample deviations from these to obtain camera intrinsics and extrinsics. We render 200 images per scene, yielding a total of 18k images. Fig. 5a shows samples from this dataset. Please refer to the supplementary material for more details about the rendering process. This results in sets of 12k, 4k and 2k images for training, validation, and testing, respectively.

Semi-synthetic BokehMe We obtain our second dataset by applying a synthetic defocus blur on real all-in-focus RGBD images using the BokehMe [31] defocus synthesis method. This approach uses a classical physics-based renderer to generate blur, and fixes the artifacts near depth discontinuities using a neural renderer. We apply this method on 294k RGBD images of DIML [20, 19, 21]. We additionally apply this method to 100k proprietary images we licensed, from each we generate 10 defocused images with randomly selected defocus parameters. The BokehMe (DIML) dataset is split into sets of 240k, 53k, and 1k images for training, validation, and testing, respectively, while the BokehMe (proprietary) is split 868k/155k/10k. Fig. 5b and fig. 5c show samples from these datasets.

Other datasets We use DED [28], a light-field-based dataset, to evaluate our methods. We also consider SYNDOF [23], a semi-synthetic dataset generated from RGBD images. Both these datasets tend to have artifacts due to their depth processing. Please refer to the supp. material for visual comparison and additional description of the datasets.

Training and Evaluation Our method and experiments employ the following datasets. First, we train our disparity and defocus modules on the Synthworld dataset and BokehMe applied to proprietary captured data. Our weight module is trained on BokehMe applied to the DIML dataset.

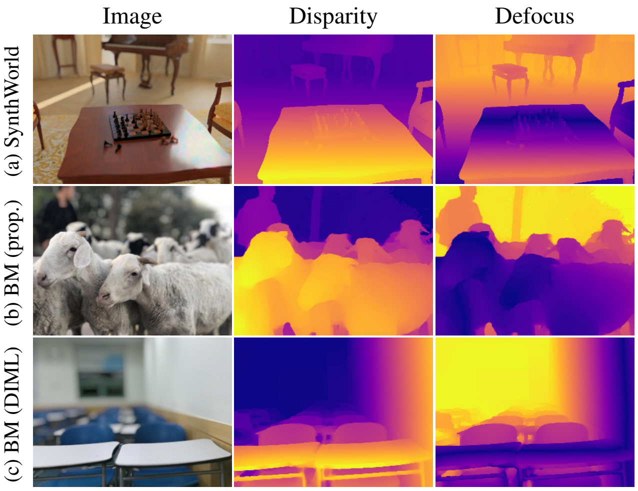

Figure 5: Example images from our datasets: a synthetic Synthworld, and b, c semi-synthetic BokehMe resp. on proprietary and DIML RGBD images. In both cases, accurate ground truth disparity and defocus images are available to train our models.

To evaluate our defocus estimation, we adopt SYNDOF [23] (8k images), DED [28] (1k images), Synthworld, and BokehMe (proprietary). We evaluate our parameter estimation on the BokehMe (DIML) dataset. We will release the code to generate our dataset from RGBD images using BokehMe as well as the necessary metadata to re-generate our BokehMe (DIML) datasets upon acceptance.

5. Defocus and disparity maps

While our primary goal is to recover camera parameters, our method relies on two per-pixel maps to perform its estimation—the defocus and disparity. In the following, we showcase the behavior of our estimated maps.

5.1. Methodology

Baselines We show our estimated defocus and disparity maps alongside the only method that provides depth and defocus simultaneously, Joint-Depth-Defocus (JDD) [52], which we re-implemented since their code is unavailable. To benchmark our defocus map estimation, we consider DMENet [23], DBENet [16] and DID-ANet [28]. For depth estimation, we evaluate the seminal MiDaS, both its convolution v2 version [35] and the transformer v3 (DPT) extension [34]. We convert all depth to disparity for comparison purposes. To evaluate our proposed architecture (see sec. 3.3), we first train the disparity estimation network on the same training set as MiDaS, which contains only sharp, all-in-focus images. We dub this pretrained network “Ours (pretrained)”. We then finetune it and jointly train our defocus estimation network on Synthworld and BokehMe exclusively, which we refer to as “Ours” later.

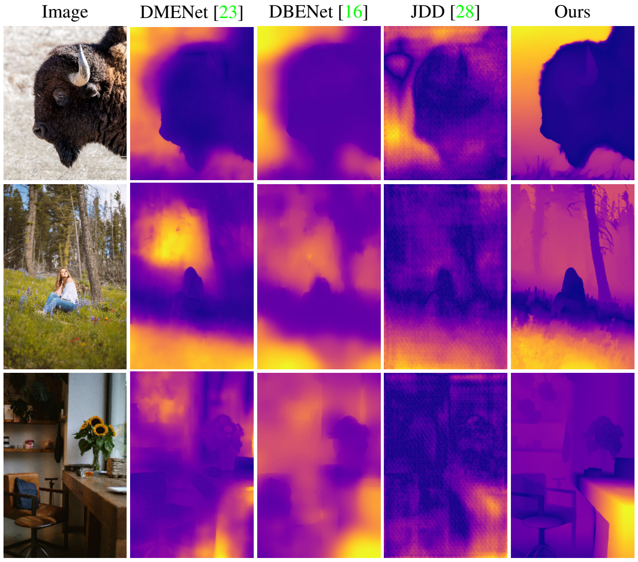

Figure 6: Qualitative comparison with existing methods for single image defocus map estimation. Our method produces significantly better defocus maps than previous methods. The estimated map is color-coded from in-focus (dark-blue) to strongly blurry out-of-focus regions (yellow).

5.2. Defocus map estimation

We first present defocus map estimation results in Tab. 1, comparing four different methods against ours. Our method performs significantly better on the real images with synthetic blur from BokehMe and SynthWorld. While DMENet [23] is an older method, we observe that DBENet [16] and DID-ANet [28] obtain worse results on our tested datasets.

We show qualitative defocus map estimation results in fig. 6. To avoid bias, input images are downloaded from the Internet and do not belong to any defocus datasets used to train the methods. Our method produces both sharper and seemingly more accurate results than existing methods. We further compare qualitatively against the ground truth of our Synthworld and BokehMe datasets (see sec. 4) in fig. 8. Our method also exhibits state-of-the-art results in this case.

5.3. Disparity map estimation

We quantitatively compare the accuracy of the evaluated models on disparity estimation on both Synthworld and BokehMe test sets in tab. 2. As expected, methods trained exclusively on all-in-focus images (top part of tab. 2) do not perform as well on images with out-of-focus blur.

We show a qualitative evaluation on disparity estimation in fig. 7. The same data precautions were taken as in fig. 6.

	SynthWorld (px)	BokehMe (px)	Syndof [23] (σ)	DED [28] (sc. px)
DMENet [23]	12.98	11.13	0.739	0.143
DBENet [16]	12.97	11.79	1.090	0.136
DID-ANet [28]	13.51	13.18	1.14	**(0.127)**[1]
JDD [52]	15.85	19.05	2.827	0.791
Ours ($\hat{\mathbf{C}}_s$)	**5.83**	2.22	0.741	0.133
Ours ($\hat{\mathbf{C}}_r$)	6.36	**2.13**	**0.726**	0.141

Table 1: Root mean square error (RMSE) on the defocus estimations[2]. Results are color-coded as **best** and second best.

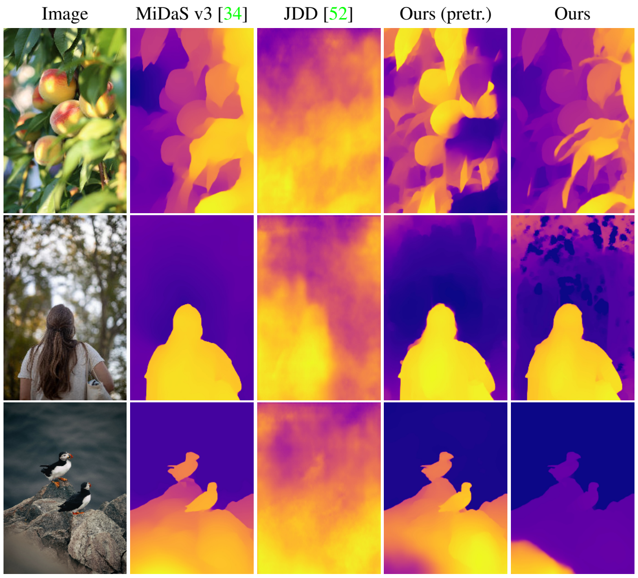

Figure 7: Qualitative comparison with existing methods for single image disparity map estimation. All images are downloaded from the Internet and do not belong to any training dataset. Our prediction provides higher disparity map quality, and it also captures more accurate details in the out-of-focus region than the other methods.

Note how methods trained exclusively on all-in-focus images (MiDaS v3 and ours (pretrained)) have issues with overly blurry regions (e.g., canopy, 2nd row, rock in the bottom left, 3rd row). Finetuning on rich datasets helps in resolving the ambiguity (ours). Similarly to the defocus map estimation, our architecture provides sharper details. Fig. 8 also shows disparity estimation results compared to the ground truth on our Synthworld and BokehMe datasets (see sec. 4).

[1] We use the DED training set for testing since ground truth defocus are unavailable on the test set. DID-ANet has seen this dataset during training.

[2] Since the evaluated methods use different defocus representations (e.g. σ vs diameter), we scale the defocus maps by the best factor (per-dataset).

	Synthworld	BokehMe
MiDaSv2 (conv.) [35]	0.186	0.190
MiDaSv3 (DPT) [34]	0.147	0.151
Ours (pretrained)	0.152	0.140
JDD [52]	0.303	0.324
Ours	**0.087**	**0.043**

Table 2: Root mean square error (RMSE) for the disparity estimatons on the Synthworld and BokehMe test sets. Results are color-coded as **best** and second best. Methods are trained (top) exclusively on all-in-focus images [35], and (bottom) on our Synthworld and BokehMe training sets.

	Synthworld		BokehMe	
	$\hat{d}_f$	$\hat{\kappa}$	$\hat{d}_f$	$\hat{\kappa}$
DMENet [23]*	0.150	17.12	0.161	24.66
DBENet [16]*	0.182	15.99	0.175	26.48
DID-ANet [28]*	0.178	15.45	0.202	28.05
JDD [52]	0.641	44.44	0.307	51.6
Ours (linear)	**0.043**	4.59	**0.033**	**3.95**
Ours (weight)	**0.043**	**4.55**	0.034	4.00

* use "ours (pretrained)" for disparity estimation.

Table 3: Median error on lens parameter. On the first four rows, we obtain the camera parameters by doing a least square fit on the estimated disparity and absolute value defocus maps, following eq. 2. (**best** and second best)

	BokehMe–DIML		
	$\hat{d}_f$	$\hat{\kappa}$	$\hat{\mathbf{C}}_r$
0) Linear fit	0.038	3.62	4.44
1) Lin. fit near focus disp	0.038	10.15	13.83
2) Median in-focus disp	0.040	7.15	6.49
3) RANSAC linear fit	0.044	4.60	4.85
4) Parameter network	0.068	11.5	4.42
5) Gradient weights	**0.037**	3.62	4.42
5) Weight network	**0.037**	**3.49**	**4.17**

Table 4: Median error on lens parameter estimation using multiple methods, color-coded **best** and second best.

6. Camera parameters estimation

We now present results on the novel task of recovering camera lens parameters, enabling us to obtain a global representation of the depth of field effect in a given scene. Since we are the first to tackle this task from a single image, there exists no previous method we can directly compare against. We therefore present two comparative analyses.

First, we compare to strong baselines on the Synthworld and BokehMe datasets by combining existing defocus map estimation techniques (DMENet [23], DBENet [16], DID-ANet [28]) with the "ours (pretrained)" disparity estimation network. All these methods employ our proposed linear least-squares fit to recover the parameters (eq. (7)). We also

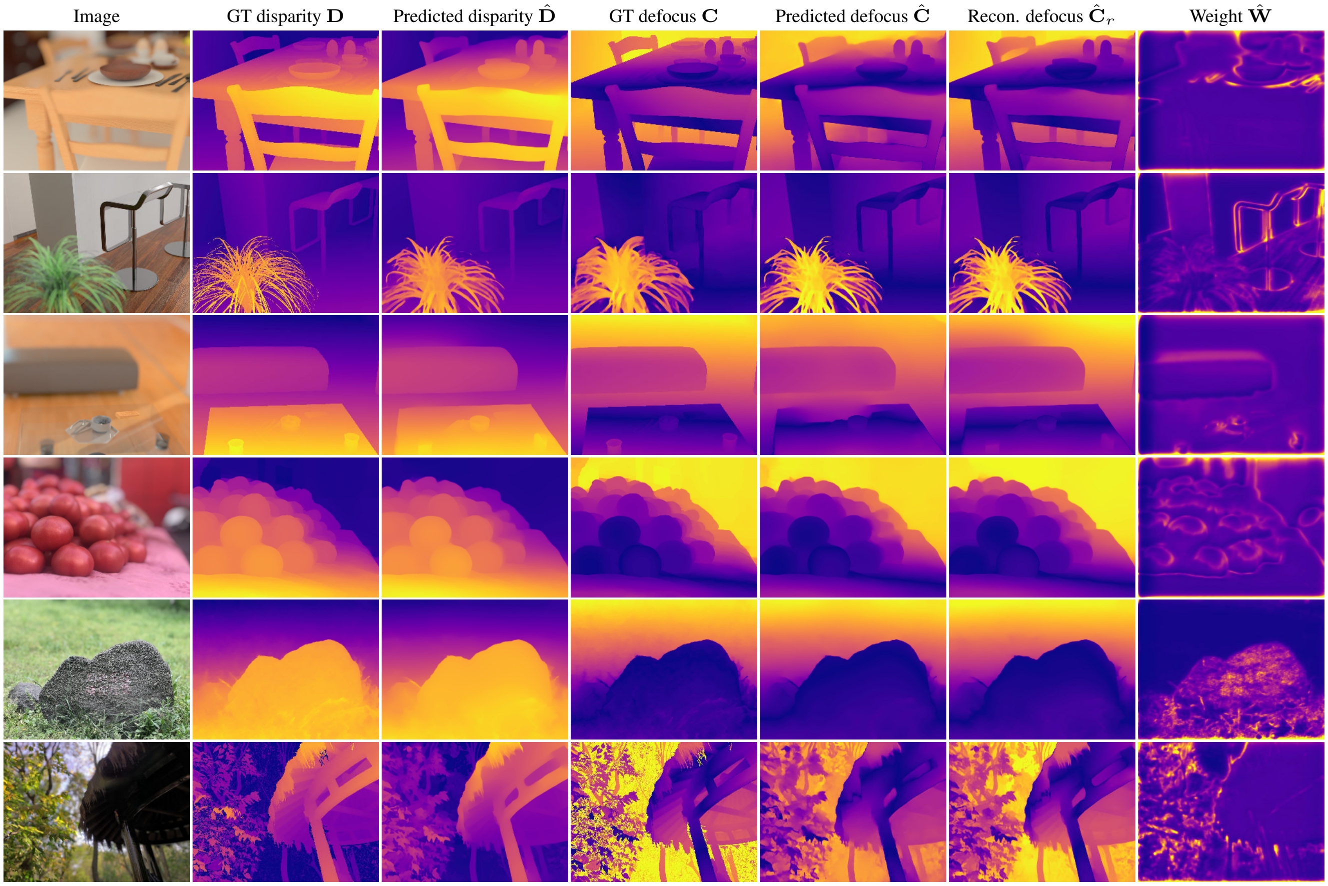

Figure 8: Qualitative results on our Synthworld (top 3 rows) and BokehMe (remainder) datasets. Left to right: input image, ground truth/estimated disparity maps, ground truth/predicted/reconstructed absolute value defocus maps, weight maps.

compare our method with linear and weight fits. As shown in tab. 3, our proposed combination of architecture, training datasets, and formulation allows for the most accurate estimation of lens parameters. We also show that our method performs the same within statistical significance with and without our weight network on these two datasets.

Second, we compare different strategies for recovering the camera parameters using our defocus and disparity estimation networks. In particular, we consider five methods: 1) do a second linear fit (eq. (7)) only on pixels with disparities within ± 10% of the initially estimated focus disparity, 2) estimate the focus disparity as the median disparity of the in-focus pixels (i.e., with estimated coc $<$ 1px), 3) use the RANSAC algorithm, 4) train a ConvNeXt-tiny [26] network with the same loss as our weight network to directly regress the parameters, 5) use the image gradients as weights, and 6) our proposed weight network. To evaluate the robustness to out-of-domain images—when the estimated maps have more outliers and mistakes than on Synthworld and BokehMe (prop.)—we perform this evaluation on a new dataset never seen in training for the per-pixel map estimation networks, BokehMe (DIML), the results of which are shown in tab. 4. Our weight network yields the best performance across estimated parameters and defocus map reconstruction.

To further showcase the robustness of our method to in-the-wild images (despite the circle of confusion approximation not modeling ideally recent sensors), we show reconstructed defocus maps on images from a wide variety of sensors in fig. 9. Interestingly, the estimated weight maps indicate that the network focuses on geometric edges and textures, where the defocus is most noticeable while discarding uniform textureless regions.

7. Applications

3D objects compositing Our method is the first to enable plausible 3D virtual object insertion from a single image with strong depth of field effect. We convert our estimated focus disparity $\hat{\kappa}$ and blur factor d_f to Blender's parametrization (focus distance and f-number). We place in the 3D scene a flat ground plane acting as a shadow catcher, and we scale its disparity so that it matches our disparity map estimation. We use [15] to get a focal length estimation, which allows us

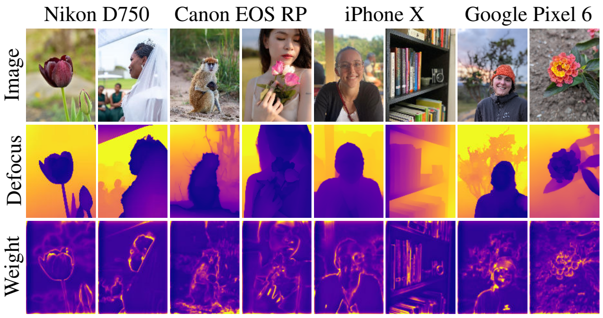

Figure 9: Reconstructed maps on in-the-wild images.

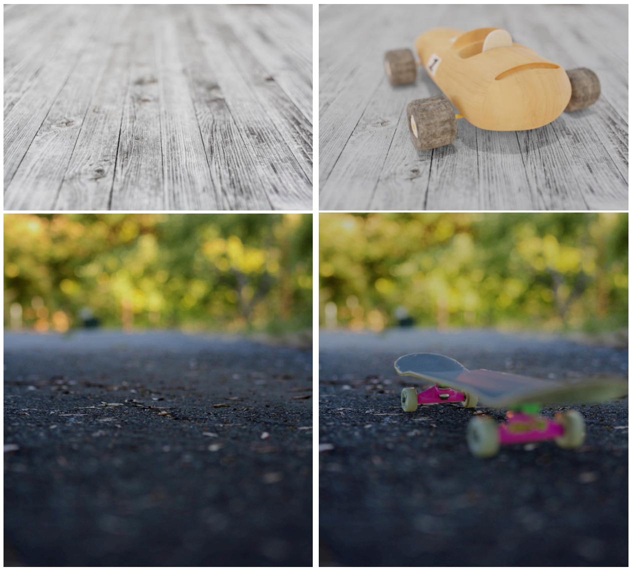

Figure 10: 3D object insertion (right) in (left) shallow DoF images. Objects are rendered using the Blender Cycles renderer, using our estimated lens parameters $(\hat{\kappa}, \hat{d}_f)$.

to recover the metric camera aperture (and f-number) using eq. (3). We showcase the results in fig. 10. Both the toy car and the skateboard plausibly match the blur effect present in the image, giving a credible insertion. Note that no 2D image processing (e.g., pixel-wise blur) is necessary here: since the virtual camera possesses the same properties as the real camera, the depth of field effect is obtained directly from the physics-based rendering engine. This can only be done if realistic lens parameters are recovered.

2D objects compositing Our approach also allows the realistic compositing of 2D objects in shallow depth of field images, which we illustrate in fig. 11. Here, we simply blur each pixel of the inserted object according to the corresponding pixel in the estimated defocus map $\hat{\mathbf{C}}_s$.

Defocus magnification Using our estimated defocus map, we can magnify the defocus effect by providing our estimated disparity and focus disparity to the BokehMe defocus generation method [31]. This is illustrated in fig. 12 where blurry regions are accentuated for artistic purposes.

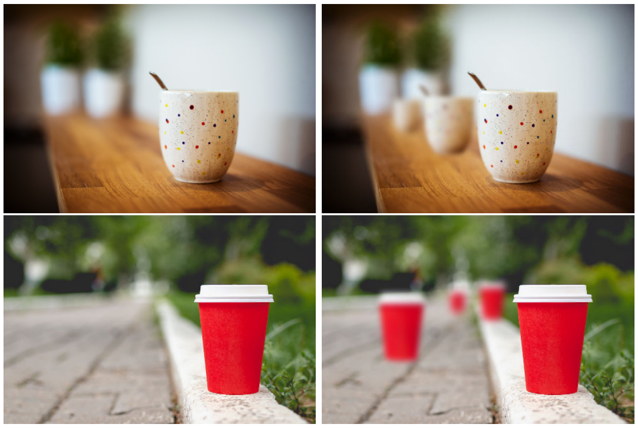

Figure 11: 2D object compositing (right) in (left) shallow DoF images. Inserted objects are blurred according to our estimated defocus map $\hat{\mathbf{C}}_s$.

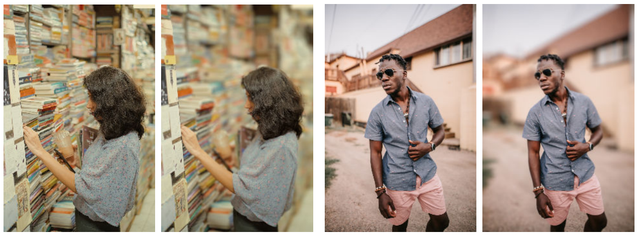

Figure 12: From images with depth of field effect (1st and 3rd), we leverage the estimated defocus map $\hat{\mathbf{C}}_s$ to magnify defocus (2nd and 4th), thereby enhancing the artistic effect.

8. Discussion

Limitations Our method yields an estimate of the *scaled* aperture: recovering metric aperture (or f-number) requires knowledge of the focal length f as well, which could potentially be solved by considering focal length estimation methods [15, 55]. Since we impose a physical model on the estimation, our approach does not handle images that have undergone manual editing, for example if physically-incorrect blur is added by an artist.

Conclusion We present a method that estimates the camera lens parameters to model depth of field from a single image. Our method is the first to enable 3D virtual object compositing and scene staging for AR. Our method advances several insights including the use of a linear lens parametrization for this task, an attention-like weight network, and explicitly enforcing the consistency between the defocus and disparity to achieve global lens parameter estimates. We hope our contributions pave the way to improve future camera lens calibration methods and inspire methods for staging virtual environments from real images.

References

[1] Evermotion - 3d models store, 3d assets, scenes, PBR materials, cg news and tutorials. https://evermotion.org/. 4

[2] Saeed Anwar, Zeeshan Hayder, and Fatih Porikli. Depth estimation and blur removal from a single out-of-focus image. In *Brit. Mach. Vis. Conf.*, volume 1, page 2, 2017. 2

[3] Soonmin Bae and Frédo Durand. Defocus magnification. In *Comput. Graph. Forum*, volume 26, pages 571–579, 2007. 2

[4] Anton Christoffersson. Real-time depth of field with realistic bokeh: with a focus on computer games, 2020. 3

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4

[6] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *IEEE Trans. Image Process.*, 25(4):1660–1673, 2016. 2

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. 2014. 2

[8] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):406–417, 2005. 2

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[10] Konstantinos Georgiadis, Albert Saà-Garriga, Mehmet Kerim Yucel, Anastasios Drosou, and Bruno Manganelli. Adaptive mask-based pyramid network for realistic bokeh rendering. *arXiv preprint arXiv:2210.16078*, 2022. 2

[11] Bernd Girod and Stephen Scherock. Depth from defocus of structured light. In *Optics, Illumination, and Image Sensing for Machine Vision IV*, volume 1194, pages 209–215. SPIE, 1990. 2

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[13] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Int. Conf. Comput. Vis.*, 2019. 2

[14] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[15] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 7, 8

[16] Ali Karaali, Naomi Harte, and Claudio R. Jung. Deep Multi-Scale Feature Learning for Defocus Blur Estimation. *IEEE Trans. Image Process.*, 31:1097–1106, 2022. 2, 5, 6

[17] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Trans. Image Process.*, 27(3):1126–1137, 2017. 2

[18] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, 2014. 2

[19] Sunok Kim, Dongbo Min, Bumsub Ham, Seungryong Kim, and Kwanghoon Sohn. Deep stereo confidence prediction for depth estimation. In *2017 ieee international conference on image processing (icip)*, pages 992–996. IEEE, 2017. 4

[20] Youngjung Kim, Bumsub Ham, Changjae Oh, and Kwanghoon Sohn. Structure selective depth superresolution for rgb-d cameras. *IEEE Transactions on Image Processing*, 25(11):5227–5238, 2016. 4

[21] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8):4131–4144, 2018. 4

[22] Brian Lee, Fei Lei, Huaijin Chen, and Alexis Baudron. Bokeh-loss gan: Multi-stage adversarial training for realistic edge-aware bokeh. *arXiv preprint arXiv:2208.12343*, 2022. 2

[23] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep Defocus Map Estimation Using Domain Adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 1, 2, 4, 5, 6

[24] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[25] Shaojun Liu, Fei Zhou, and Qingmin Liao. Defocus map estimation from a single image based on two-parameter defocus model. *IEEE Trans. Image Process.*, 25(12):5943–5956, 2016. 2

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 7

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2017. 4

[28] Haoyu Ma, Shaojun Liu, Qingmin Liao, Juncheng Zhang, and Jing-Hao Xue. Defocus Image Deblurring Network With Defocus Map Estimation as Auxiliary Task. *IEEE Trans. Image Process.*, pages 216–226, 2022. 1, 2, 4, 5, 6

[29] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F Hughes, and Frédo Durand. Defocus video matting. *ACM Trans. Graph.*, 24(3):567–576, 2005. 2

[30] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 1, 2

[31] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When neural rendering meets classical rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2, 4, 8

[32] Said Pertuz, Miguel Angel Garcia, and Domenec Puig. Efficient focus sampling through depth-of-field calibration. *Int. J. Comput. Vis.*, 112(3):342–353, 2015. 2

[33] Michael Potmesil and Indranil Chakravarty. A lens and aperture camera model for synthetic image generation. *ACM SIGGRAPH Computer Graphics*, 15(3):297–305, 1981. 3

[34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Int. Conf. Comput. Vis.*, 2021. 2, 5, 6

[35] 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 Trans. Pattern Anal. Mach. Intell.*, 2020. 2, 5, 6

[36] Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. AIFNet: All-in-Focus Image Restoration Network Using a Light Field-Based Dataset. *IEEE Trans. Comp. Imag.*, 7:675–688, 2021. 2

[37] Parikshit Sakurikar, Ishit Mehta, Vineeth N Balasubramanian, and PJ Narayanan. Refocusgan: Scene refocusing using a single image. In *Eur. Conf. Comput. Vis.*, 2018. 2

[38] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In *Adv. Neural Inform. Process. Syst.*, 2005. 2

[39] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, 2008. 2

[40] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2

[41] Vivek Srikakulapu, Himanshu Kumar, Sumana Gupta, and KS Venkatesh. Depth estimation from single image using defocus and texture cues. In *Nat. Conf. Comp. Vis. Pat. Recog. Img. Proc. Graph.*, 2015. 2

[42] Murali Subbarao and Gopal Surya. Depth from defocus: A spatial domain approach. *Int. J. Comput. Vis.*, 13(3):271–294, 1994. 2

[43] Yu-Wing Tai and Michael S Brown. Single image defocus map estimation using local contrast prior. In *IEEE Int. Conf. Image Process.*, 2009. 1, 2

[44] Chang Tang, Chunping Hou, and Zhanjie Song. Depth recovery and refinement from a single image using defocus cues. *Journal of Modern Optics*, 62(6):441–448, 2015. 3

[45] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[46] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8:415–424, Sept. 2022. 4

[47] Masahiro Watanabe and Shree K Nayar. Rational filters for passive depth from defocus. *Int. J. Comput. Vis.*, 27(3):203–225, 1998. 2

[48] Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. Dof-nerf: Depth-of-field meets neural radiance fields. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1718–1729, 2022. 2

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Adv. Neural Inform. Process. Syst.*, 2021. 4

[50] Guodong Xu, Yuhui Quan, and Hui Ji. Estimating defocus blur via rank of local patches. In *Int. Conf. Comput. Vis.*, 2017. 2

[51] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[52] Anmei Zhang and Jian Sun. Joint Depth and Defocus Estimation From a Single Image Using Physical Consistency. *IEEE Trans. Image Process.*, 30:3419–3433, 2021. 1, 2, 3, 5, 6

[53] Xinxin Zhang, Ronggang Wang, Xiubao Jiang, Wenmin Wang, and Wen Gao. Spatially variant defocus blur map estimation and deblurring from a single image. *Vis. Comm. Img. Rep.*, 35:257–264, 2016. 2

[54] Changyin Zhou, Stephen Lin, and Shree Nayar. Coded aperture pairs for depth from defocus. In *Int. Conf. Comput. Vis.*, 2009. 2

[55] Rui Zhu, Xingyi Yang, Yannick Hold-Geoffroy, Federico Perazzi, Jonathan Eisenmann, Kalyan Sunkavalli, and Manmohan Chandraker. Single view metrology in the wild. In *Eur. Conf. Comput. Vis.*, 2020. 8

[56] Xiang Zhu, Scott Cohen, Stephen Schiller, and Peyman Milanfar. Estimating spatially varying defocus blur from a single image. *IEEE Trans. Image Process.*, 22(12):4879–4891, 2013. 2

[57] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011. 2

[58] Wei Zhuo, Mathieu Salzmann, Xuming He, and Miaomiao Liu. Indoor scene structure analysis for single image depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2