

Camera-Independent Single Image Depth Estimation from Defocus Blur

Lahiru Wijayasingha

Computer Science Department
University of Virginia, USA
lnw8px@virginia.edu

Homa Alemzadeh

Electrical and Computer Engineering Department
University of Virginia, USA
ha4d@virginia.edu

John A. Stankovic

Computer Science Department
University of Virginia, USA
stankovic@cs.virginia.edu

Abstract

Monocular depth estimation is an important step in many downstream tasks in machine vision. We address the topic of estimating monocular depth from defocus blur which can yield more accurate results than the semantic based depth estimation methods. The existing monocular depth from defocus techniques are sensitive to the particular camera that the images are taken from. We show how several camera-related parameters affect the defocus blur using optical physics equations and how they make the defocus blur depend on these parameters. The simple correction procedure we propose can alleviate this problem which does not require any retraining of the original model. We created a synthetic dataset which can be used to test the camera independent performance of depth from defocus blur models. We evaluate our model on both synthetic and real datasets (DDFF12 and NYU depth V2) obtained with different cameras and show that our methods are significantly more robust to the changes of cameras. Code: https://github.com/sleekEagle/defocus_camind.git

1. Introduction

Computer vision based depth estimation has many applications such as augmented and virtual reality (AR and VR) [27], autonomous robotics [8], background subtraction, and changing the focus of an image after it was taken [35] [19] [5]. Techniques such as structure from motion, structure from shading, shape from structured light, shape from defocus blur, depth from focus, multi-view stereo and Time-of-Flight (ToF) sensors can be used to estimate the depth of a scene [10,36]. Active methods such as structured light and ToF sensors need specialized hardware and are power hungry. Stereo techniques measure depth by relying on multiple cameras to take several pictures of the scene. Techniques such as structure-from-motion and depth-from-focus [38] require several images of a static scene to estimate its structure. Also, the assumption about static scene

does not hold when the scene is changing over time. Furthermore, structure from motion can only recover the depth of a scene up to a scale and cannot measure the absolute depth.

Single image defocus blur based depth estimation is a fairly under-explored topic in the literature [35] which utilizes the phenomena that certain objects in a photo appear more blurred than the others depending on the distance to those objects from the camera. Therefore, measuring the amount of defocus blur at a point of an image can provide a way to recover the depth to the respective point in the real 3D world. As we will show in Section 3, this method is effective for close range depth measurements (typically under 2 to 3 meters). This makes defocus blur-based depth estimation techniques ideal for measuring depth under many situations including in microscopic scenes [2,40] and measuring depth to hands and nearby objects for a wearable camera.

Single image depth from defocus blur methods are not robust to changes of cameras. As we will show in our experiments, the performance of existing methods degrades significantly when they are trained on images taken from one camera and evaluated on images taken from another camera (even when they both image the same scene). This is due to the fact that different cameras will produce defocus blurs with different characteristics.

In this paper we describe a novel technique to estimate depth from defocus blur in a camera-independent manner. We exploit the optical physics equations that describe the relationships between various camera parameters and the amount of defocus blur. Our method can be used to train a deep learning model in a supervised manner on a dataset containing defocus blurred images taken from a single or multiple camera/s and respective ground truth depth maps. This trained model can be used to predict depth using images taken with a wide range of other cameras with a slight modification to the model (depending on the particular camera parameters of the new camera) and without the need for retraining. We also describe a novel method to estimate the camera parameters of a given camera with an easy to use calibration process. This will be particularly useful when

the parameters for a certain camera cannot be obtained (certain manufacturers do not provide all the parameters in the data-sheets and/or the values are only provided as approximations).

Our main **contributions** are as follows:

- We show that depth from defocus technique can measure depth more accurately than the state-of-the-art techniques.
- We show that existing depth from defocus methods are not robust to changes of cameras the images are acquired with.
- This paper is the first to device a relationship between defocus blur and the blur created due to pixel binning.
- We present a novel depth from defocus blur method which is robust to images taken from a wide range of cameras, given camera parameters that describe a particular camera.
- We present a novel calibration technique to estimate the camera parameters based on several images taken from a given camera.
- Our methods have less estimation error than the state-of-the-art when performing depth from blur in a camera-independent manner. The error reduction is around 3cm under the DDF12 dataset, 7cm under the NYU depth v2 dataset and around 5cm for the synthetic dataset we created.

2. Related Work

2.1. Depth from RGB images

Estimating depth maps from images can use various characteristics of images such as semantics, stereo matching, blur or differences in blur over a stack of images [6, 15, 34]. Although stereo matching based and blur based depth measurements are seen as completely separate methods, Schechner and Kiryati [28] showed that both of them can be understood under the same mathematical formulation. A depth map can be estimated for the given image based on the domain knowledge on the structure of the objects in the image embedded in the estimation model [3, 16, 20, 25, 41]. Methods such as ZoeDepth [3] and VPD [41] have pushed the state-of-the art to be very accurate in measuring depth. However, a problem with these methods is that the estimated depth is only an approximation based on the structure of the objects. This makes these models sensitive to domain changes [10]. Also, techniques that can recover 3D structure from RGB images such as structure from motion can only estimate relative depths in a given scene [37].

2.2. Depth from defocus blur

The amount of defocus blur can be used to measure the depth of a scene from images. Since these methods rely

more on blur which is a local feature of the image to estimate the depth, they are more robust to domain changes [10]. Shape/depth from focus methods aim to measure depth to a scene given a stack of images of different focus levels. A measure of the sharpness of each pixel of the images over the stack is calculated. The depth of a point is taken as the focus distance with the sharpest pixel. Various methods such as the Laplacian or sum-modified-Laplacian, gray-level variance and gradient magnitude squared were traditionally used to measure the sharpness [21, 32]. Modern methods utilize deep learning to automatically learn the sharpness measure from focal stacks [11, 35, 38]. But deep learning based techniques require a large amount of data to train [19].

Depth from focus methods that use a focal stack of the same scene has several drawbacks. First they assume the scene is static during the time needed to acquire several images with different focus (focal stack). Second, an accurate registration of the images in the focal stack is needed due to focal breathing (slight change of the field-of-view of the camera due to changes of focal distance) or small movement of the camera and/or the scene [17]. Therefore, more investigation on depth estimation with a single image is necessary. Depth from defocus/blur rely on measuring the exact blur on a single image to estimate the depth and cannot use the relative variation of sharpness/blurriness of a focal stack. Due to this, depth from blur can be used to estimate the depth from a single blurred image [1, 10, 19, 30, 39]. Certain works are also concerned about removing the blur at the same time as estimating depth [1, 19, 26].

Estimating depth from the amount of the blur of a single image is ill-posed. This is due to having two possible depth values for a given blur [10]. Researchers have take two different paths to solve this. One solution is hardware based. One example for this is changing the shape of the aperture (coded aperture) of the camera to a shape that can help avoid the ambiguity. Ikoma et al. [12] used deep learning to learn the optimal shape for an aperture and came up with a prototype camera to measure depth from blur. Another example is to use a lightfield camera which takes many pictures with closely spaced micro lenses placed inside the camera [22]. The second approach is to use the domain knowledge (e.g. the shape and sizes of objects in the scene) of the scene to remove the ambiguity. Our research falls into this category. Gur and Wolf created a model which can generate the depth map of a scene given the blurred image and the All-in-Focus (AiF) image [10]. They also makes certain assumptions about the shape of the blur circle. Usage of both AiF image and blurred images in making prediction makes this model less useful in certain situations because both of these images are not usually available from regular cameras. Many methods in the literature first estimate the blur of a given blurred image and secondly estimate the

depth from the blur. Physical consistency between the estimated blur and depth has been used as a form of domain knowledge by Zhang et al. [39]. Lu et al. create two separate models to estimate the blur and the amount of focus (sharpness) of a given blurred image. They claim that this method provides better estimates of depth due to the capability of estimating both blur and the sharpness of an image [18]. But since sharpness is just the inverse of blur, a question remains that by estimating blur aren't we also estimating the (inverse of) sharpness. Ban et al. [2] extend depth from blur to microscopic images. Certain works focus just on blur estimation from a blurred image. Tai and Brown use hand crafted features of an image to estimate the blur map [33] while Zhuo and Sim assume that the edges in the images are step edges [42]. Cun et al. estimated the blur of a given image to separate the blurred and focused areas from a blurred image [7]. While all of the above methods assume that the blur is a single parameter (e.g. Gaussian or disk shape) Liu et al. expand our understanding by introducing a two parameter model. This model is also helpful in removing errors due to pattern edges [17].

2.3. Camera dependency of depth from blur

Certain characteristics of blur depend on the camera that is being used to acquire the images. The blurred image of a point has the same shape (but scaled) as the lens aperture. For example, if the aperture is circular, the blur of a point is also circular theoretically. But in practice this is a Gaussian due to diffraction [30]. In this research we assume all the apertures are circular in shape.

The size of the blur of a point depends on many other parameters of the camera. The f-number, focal length, pixel size of the image sensor (if the camera is digital), camera output scale and focal distance all affect the size of the blur [10] as shown in section 3. Depth from defocus blur techniques estimate the blur of a given image as an intermediate step when estimating the depth. This makes these models sensitive to the variations due to camera parameters. We show evidence supporting this in our evaluation section. But no papers in the literature address this problem. Gur and Wolf [10] use camera parameters in their model to recreate a blurred image. It was not used directly to predict depth and they do not test their model under different cameras. Although Maximov et.al [19] evaluate their model on several simulated datasets generated with different camera parameters they do not explicitly address or propose a solution to this problem.

3. Approach

This section starts with a theoretical introduction to estimating depth from defocus blur and establishes the challenges faced by this technique and our solution.

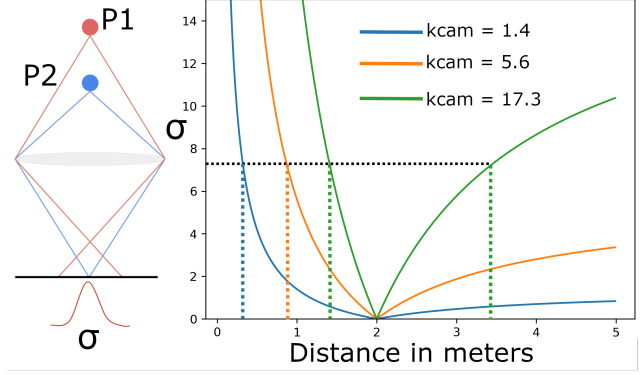


Figure 1. Left:Image formation in a simple camera system. Right:Blur vs. distance

3.1. Theory and Techniques

When imaging a scene with a camera, the points that are not in focus appear blurred and the points that are perfectly in focus appear sharp in the image. This phenomenon is called defocus blurring. To illustrate this, in the left side of the Figure 1, the point $P2$ that is in focus appears as a point in the image plane of the camera. A point $P1$ that is not in focus appears as a blur in the image plane where the pixel intensity is the highest at the center and gradually falls off as we move away. This can be modelled with a 2D Gaussian function as denoted in equation 1 with σ as the standard deviation, x and y are image coordinates.

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{1}{2} \frac{x^2+y^2}{\sigma^2}} \quad (1)$$

σ depends on the distance to the point $P1$ from the camera center and several other camera dependent factors as shown in equation 2.

$$\frac{|s_1 - s_2|}{s_2} \cdot \frac{1}{(s_1 - f)} \cdot \frac{f^2}{N} \cdot \frac{1}{p} \cdot \frac{out_{pix}}{sensor_{pix}} = k_r \cdot \sigma \quad (2)$$

In equation 2, s_2 is the depth (distance to the) f is the focal length of the camera, N is the f-number, p is the pixel width, out_{pix} is the number of pixels in the final image, $sensor_{pix}$ is the number of pixels in the image sensor, s_1 is the focus distance, k_r is a constant that depends on the camera [31]. Many cameras allow user to change s_1 thereby focusing the camera at different distances. we define a camera dependent parameter k_{cam} as shown in equation 3.

$$\frac{|s_1 - s_2|}{s_2} \cdot k_{cam} = \sigma \quad (3)$$

$$\text{where } k_{cam} = \frac{1}{(s_1 - f)} \cdot \frac{f^2}{N} \cdot \frac{1}{p} \cdot \frac{out_{pix}}{sensor_{pix}} \cdot \frac{1}{k_r}$$

$G(x, y)$ is the response of the camera system to a point target and is called the point spread function (PSF). We can obtain the defocus blurred image $B(x, y)$ by convolving the

the perfectly focused image $F(x, y)$ with PSF $G(x, y)$ as show in equation 4.

$$B(x, y) = G(x, y) * F(x, y) \quad (4)$$

Equation 4 explains the blur solely due to defocus blurring. An additional blurring can occur due to various other reasons such as filtering in the camera hardware (e.g. to reduce noise), pixel binning, color filter mosaics, analog/digital image processing, analog to digital conversion, etc. [13]. This additional blurring can also be modelled as a convolution with another Gaussian function $Q(x, y)$ having a standard deviation γ which we assume to be constant for a given camera setting. The final image can be obtained by

$$I(x, y) = Q(x, y) * G(x, y) * F(x, y) \quad (5)$$

All we can observe is the final image I . We show that the combined blurring (from defocus and due to other reasons described above) can also be modelled with a Gaussian PSF (refer the Appendix) and we can estimate the standard deviation of this PSF (λ) at each pixel in I . After estimating λ for a given image and when γ is known we can obtain σ as shown in the equation 6.

$$\sigma = \sqrt{\lambda^2 - \gamma^2} \quad (6)$$

Substituting equation 6 into equation 3 we can obtain equation 7.

$$\frac{|s_1 - s_2|}{s_2} \cdot k_{cam} = \sqrt{\lambda^2 - \gamma^2} \quad (7)$$

The right side of the Figure 1 shows the variation of σ with different distances (s_2) and under different cameras. For a given camera (hence for a given k_{cam}), we can estimate the σ from a given image and then estimate s_2 . For certain sections of the curve (e.g. curve of $k_{cam} = 17.3$ at the shown value of σ), estimating s_2 is ambiguous since there will be two s_2 values for a given σ . This limitation can be mitigated by using a learning based model to estimate s_2 . Another observation is that the value of σ depends on k_{cam} . This poses the main problem that we are addressing in this paper. If a model was trained to estimate depth using data from a camera with one k_{cam} , this model will fail to predict the depth accurately for images taken with a camera having a different k_{cam} . Furthermore, the sensitivity of σ to the distance diminishes as the distance increases. Hence the effectiveness of the defocus blur based depth measurement techniques will also lessen with increasing distance. This limits the effectiveness of depth from defocus blur techniques to close range; as a rule of thumb, to distances less than 2m.

Table 1 shows some camera models and their k_{cam} values based on the particular lens/settings used. Please refer to the Appendix for a more detailed calculation and for k_{cam} values for more cameras/settings.

Camera/device	Lens	f (mm)	N	K_{cam}
Cannon EOS Rebel T7	EF-S	18	4	15.54
	EF 50mm	50	5.6	105.58
	EF 70-300mm	70	1.2	406.16
Nikon D7500	Nikon AF-S	50	5.6	172.35
	AF-S DX NIKKOR	18	1.8	240.40
		18	3.5	15.76
		18	5.6	9.85
		55	3.5	149.98
55	5.6	93.73		
Sony Alpha 7 IV	FE PZ 16-35mm	16	4	8.92
		35	4	43.13
		16	22	1.62
	FE 70-200 mm	70	2.8	250.94
		200	2.8	2196.48
		70	22	31.93
Google Pixel 7 Pro	wide	25	1.85	50.39
	telephoto	120	2.55	1577.82

Table 1. k_{cam} of some popular cameras

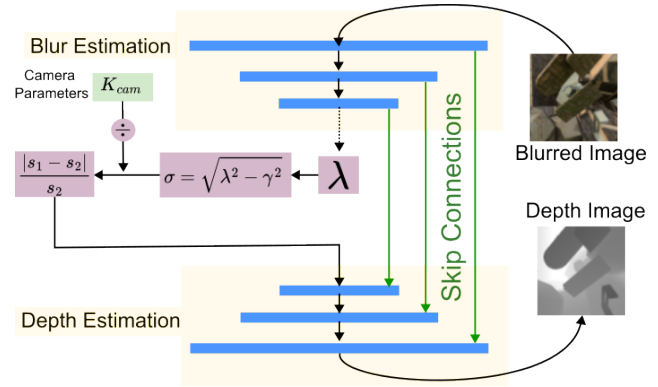


Figure 2. Our Model

3.2. Our solution

The operation of our model is shown in Figure 2. Both Blur Estimation and Depth estimation sections are CNN based neural networks inspired by the defocusnet [19]. Given a defocus blurred image, the blur estimation model estimates the PSF standard deviation λ at each pixel of the image. Then we calculate the standard deviation of the PSF solely due to defocus blurring σ according to equation 6. Next we divide the obtained σ with k_{cam} to obtain $\frac{|s_2 - s_1|}{s_2}$ which does not depend on either k_{cam} or γ . Then $\frac{|s_2 - s_1|}{s_2}$ is sent to the depth estimation model to estimate s_2 . Note that we require a learning based model such as a CNN to estimate s_2 due to the ambiguity explained in the previous section and to impart semantic domain knowledge of the image into the estimation process via the skip connections.

When we train our model, we calculate two types of losses; blur estimation loss and the depth estimation loss. Blur estimation loss (L_b) is calculated at the prediction of λ . Ground truth L_b can be obtained with equation 2 with known camera parameters at the training time. Depth prediction loss (L_d) is calculated comparing the predicted and

ground truth depth maps. The final loss is obtained by,

$$L_{total} = L_d + b_weight \cdot L_b \quad (8)$$

where b_weight is a parameter used to scale L_b .

3.3. Defocus Blur Calibration

Assume we train our model with images from a certain camera and need to use this already trained model to estimate depth using images from another camera with a different k_{cam} and γ . In this section we present our novel method that can be used to estimate these parameters for a given camera. We call this method the "Defocus Blur Calibration". Note that defocus blur calibration is different from but requires the conventional camera calibration where camera intrinsics and distortion coefficients are estimated. The steps for defocus blur calibration are as follows.

1. Fix the focal distance of the camera at s_1 (we used $s_1 = 2m$ in our experiments) and calibrate the camera (in a conventional sense) with a calibration pattern [23]. We have used an asymmetric circular pattern as can be seen in Figure 3. Maintain a rough distance of around $\frac{s_1}{2}$ from the camera to the calibration pattern. After this calibration, we can estimate the distance to a given point on the calibration pattern that is visible in a given image.
2. Capture two images of a circular calibration pattern (preferably the same pattern that was used in step 1) while maintaining a distance of $\frac{s_1}{2}$ (we used $1m$) from the camera to the pattern. The first image is obtained with the camera focused on the pattern ($s_1 = 1m$) and the second image is obtained while maintaining $s_1 = 2m$. Since the first image is focused on the calibration pattern, the circles on the pattern will appear sharp as shown in the upper part of Figure 3. The second image will look blurred as shown in the bottom half of the Figure 3. According to Figure 3, the images of circle edges on the focused images have a steeper slope (A Gaussian with a lower std). The slight blurring in these images are solely due to pixel binning. The edges on the blurred images have a more gradual slope. Also the slope becomes even more gradual as k_{cam} is increased.
3. Estimate the std of the Gaussian function of the circle edges from the focused image. We horizontally slice the image of the circle as seen in Figure 3 and obtain the distribution of pixel intensities. These are flat-top Gaussian functions. The flat top nature is due to the intensity being constant inside the circle. It falls gradually at the edges of the circles. We scale these intensity values into the range from zero to one. We consider all the values less than a threshold (we used

0.95) as belonging to the falling edges. We then integrate the resulting distribution (one dimensional Gaussian). According to equation 9, we can estimate γ after obtaining the integral J for a constant y .

$$J = \int_{-\infty}^{\infty} G(x)dx = \int_{-\infty}^{\infty} e^{-\frac{1}{2} \frac{x^2+y^2}{\gamma^2}} dx = \gamma\sqrt{2\pi} \quad (9)$$

4. Estimate the std of the Gaussian of the falling edges (λ) of the circles from defocus blurred images similarly to step 3. Note the the blurring of the defocused images are due to both defocus blurring and pixel binning. We can estimate the std of the Gaussian of the falling edges due to defocus blurring with equation 6.
5. Estimate the distance to each circle center from camera using the defocus blurred images using the camera intrinsic matrix generated with calibration in step1. This is a well-established procedure that is available in most of the computer vision libraries. We can write a separate version of equation 7 for each circle in the calibration pattern. With λ and γ already estimated, we can estimate the k_{cam} for the given camera using equation 7. Here we have assumed that the distance from the camera to each circle center is approximately equal to the distance to the edges of the circle. This can be justified because the distance to the circles from the camera (around 1m) is much larger than the diameter of the circles (around 4cm in our case).
6. To improve the accuracy of the estimate, we can repeat steps from 2 to 5 several times. See the evaluation section for further details on the experiments.

We can estimate the k_{cam} for a given camera with the above steps. The estimated k_{cam} can be used as shown in Figure 2 to predict depth with the images taken from this new camera.

4. Experiments

4.1. Datasets

Defocusnet dataset. We use the synthetic dataset generated by Maximov et al. [19] to train one of our models. This dataset was created with a virtual camera having several K_{cam} values of 0.15,0.33,0.78,1.59 and 2.41. This dataset has 500 focal stacks, each with 5 images with different focal distances.

Synthetic Blender dataset. We create a new synthetic dataset by expanding the defocusnet dataset [19]. This new dataset has various textures (to make them realistic) mapped to the 3D objects that was not present in the original dataset. We use several simulated cameras with K_{cam} of 0.08, 0.15,

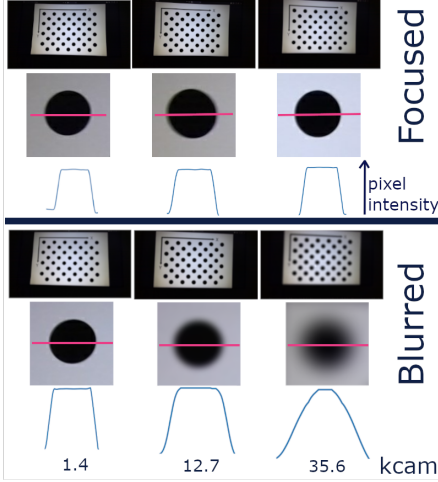


Figure 3. Defocus Blur Calibration

0.23 and 0.33. This dataset has a focal distance of 1.5m and contains 400 defocus blurred images. We use the script provided by Maximov et al. (modified) to create our dataset. The images we generate in this dataset are 256 x 256 pixels.

Both the Synthetic blender and the defocusnet datasets also have a perfectly focused image per each defocus blurred image.

DDFF12 dataset. We also use the DDFF12 dataset provided by Hazirbas et al. [11] which contains 720 images created with a lightfield camera. We use the two real world datasets (DDFF12 and the NYU dataset described next) so that we can show that our models can work under real world images and deal with the domain gap between real and synthetic images [37]. After pre-processing the images as mentioned by Hazirbas et al. we obtained the blurred images which are focused at various distances.

NYU depth v2 dataset. The NYU depth v2 dataset [29] contains 1449 pairs of aligned RGB and depth image pairs. Following previous papers [19] [9] we create the training and testing splits. We create artificially defocus blurred images from this dataset using the method described by Carvalho et al. [4]. We have fixed certain drawbacks in their Matlab script in order to produce more realistic defocus blurred images as further discussed in the appendix. We used K_{cam} values for training and testing as shown in Table 3. Images of 480 x 480 pixels were used for training and 480 x 640 were used for testing.

4.2. Experimental Setup

We use PyTorch [24] to implement the neural networks. We use the Adam optimizer [14] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of 10^{-4} . Mean Squared Error was used as the loss function for both the blur and the depth to train all the models. We evaluate our depth predictions with the metrics absolute relative error (REL), mean-squared error (MSE), Root-Mean-Squared error (RMSE)

Method	K_{cam}			
	0.08	0.14	0.23	0.33
in-focus	0.099	0.081	0.082	0.100
No K_{cam}	0.062	0.050	0.056	0.085
GT K_{cam}	0.045	0.037	0.052	0.061

Table 2. Performance on Blender dataset (MSE)

and average log10 error. We also report threshold accuracy δ_n which is the percentage of pixels which satisfy the condition $\max(d_i/\hat{d}_i, \hat{d}_i/d_i) < 1.25^n$. We train our models on the defocusnet dataset for 400 epochs and a batch size of 20. Our NYU depth models were train for 800 epochs with a batch size of 8.

4.3. Performance

Table 2 shows the performance of the model trained on the defocusnet [19] dataset and evaluated on our Blender dataset with different k_{cam} values of simulated cameras. All three methods; in-focus, No K_{cam} and GT K_{cam} , use the same deep learning architecture to predict depth. The only exception is that the GT K_{cam} model performs the K_{cam} correction as shown in Figure 2. We use the K_{cam} values that were used to generate the data and these can be called the Ground Truth K_{cam} values (GT K_{cam}). The In-focus model was both trained and tested on perfectly focused images. The No K_{cam} model does not consider the effect of K_{cam} during either training or testing (similar to defocusnet [19] model). This means the No K_{cam} model does not divide the output of the blur estimation model with K_{cam} as shown in Figure 2. The GT K_{cam} model on the other hand considers the effect of K_{cam} and behaves as shown in Figure 2 during both training and testing. According to Table 2 the performance of both the No K_{cam} and the GT K_{cam} models are better (by around 0.025) than that of the in-focus method which shows that considering defocus blur is valuable when estimating depth. Our models perform better when considering the effect of K_{cam} (GT K_{cam}) compared to when not considering it (No K_{cam}) by around 0.015 in MSE. This shows that we can transfer the knowledge learned with the trained model into a new domain (images taken with a different K_{cam}) just with one parameter K_{cam} .

Table 3 shows the performance on the defocus blurred NYU depth dataset. Here we use a single trained model to evaluate the performance under various settings. The model was trained on data refocused with a K_{cam} of 8.79 and 35.61 and tested on the rest under the distance range of 0 to 2 m. The VPD model was trained and tested on in-focus images with no defocus blurring. Our GT and est K_{cam} methods outperform the state-of-the-art depth estimation model (VPD) on the NYU depth v2 dataset [41] by around 0.04 in RMSE. This converts to a reduction of error of around 4cm in the depth estimation. This proves again the importance

K_{cam}	method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL \downarrow	RMSE \downarrow	log10 \downarrow
in-focus	VPD [41]	0.953	0.992	0.999	0.052	0.154	0.027
8.79	GT K_{cam}	0.976	0.997	0.999	0.046	0.082	0.019
8.79	No K_{cam}	0.912	0.975	0.998	0.095	0.161	0.037
35.61	GT K_{cam}	0.976	0.997	0.999	0.046	0.082	0.019
35.61	No K_{cam}	0.962	0.995	0.999	0.054	0.101	0.023
12.69	GT K_{cam}	0.969	0.999	0.999	0.068	0.123	0.088
12.69	est K_{cam}	0.970	0.999	0.999	0.068	0.122	0.030
12.69	No K_{cam}	0.853	0.963	0.999	0.127	0.193	0.050
22.67	GT K_{cam}	0.980	0.998	0.999	0.068	0.117	0.028
22.67	est K_{cam}	0.980	0.998	0.999	0.069	0.118	0.028
22.67	No K_{cam}	0.896	0.994	0.999	0.105	0.165	0.043

Table 3. Performance on NYU dataset

of defocus blurring in depth estimation. We evaluate our models under three methods which depend on the nature of the K_{cam} values used. The method column, "GT K_{cam} " means we have used the K_{cam} values that were used to defocus blur the particular dataset which can be considered as Ground Truth K_{cam} values. "est K_{cam} " represents the K_{cam} values that were estimated with the defocus calibration method described in section 3.3. No K_{cam} models do not consider the effect of K_{cam} . We describe further details of the estimation process in the subsequent sections.

Table 4 shows the performance of our model under the DDF12 dataset [11]. All the models were first trained on the defocusnet dataset [19] (the same model we used to evaluate the Blender dataset as shown in Table 2). The In-focus model was trained and tested on well focused images. The No K_{cam} and est K_{cam} models were trained and tested on defocus blurred images. Since we do not have ground truth K_{cam} for the DDF12 dataset, we performed a linear search of the K_{cam} which predicts the best depth using the ground truth depth maps provided in the training set. The results in Table 4 are the performance of our model under the test set using the K_{cam} value found above. Both the No K_{cam} and est K_{cam} models perform better than the in-focus model for depth prediction. Also using the appropriate K_{cam} to transfer the model to the new domain of images significantly improves the performance compared to the no K_{cam} model which does not perform a correction that depends on the camera.

4.4. Defocus Blur Calibration Performance

We expand the discussion on defocus blur calibration in this section. These experiments were performed on the refocused NYU depth v2 dataset. We have created refocused data with K_{cam} values of 1.39, 5.61, 8.79, 12.69, 22.67, 25.61. Note that we have used the additional K_{cam} values (1.39 and 5.61) that were not used to evaluate the performance of depth estimation in Table 2. We obtain several photos of the asymmetric circular pattern shown in Figure 3 with the Microsoft Kinect camera and refocus them with the above mentioned K_{cam} values. We used from 19 to

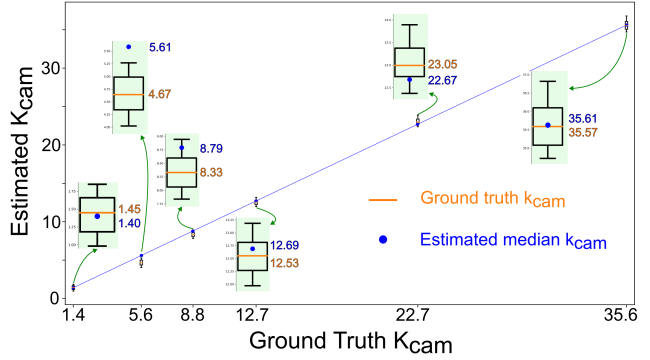


Figure 4. Estimation of K_{cam} values of different cameras

20 different image pairs (an in-focus image and a defocus-blurred image) for each K_{cam} value. Then we perform the defocus blur calibration procedure described in section 3.2. Estimated K_{cam} values vs. the actual values (ground truth K_{cam}) are shown in Figure 4. The relationship between the ground truth and estimated K_{cam} values are very linear as expected. We estimate one K_{cam} value per one circle from an in-focus and defocus blurred image pair. Since there are 44 circles in the pattern, for 20 image pairs we obtained 880 estimated K_{cam} values. The results in Figure 4 were obtained after removing outliers and calculating the median from the estimated K_{cam} values. We show box plots with interquartile range, median, minimum and maximum values of these estimations along with the ground truth K_{cam} values.

4.4.1 Sensitivity of depth estimation performance to K_{cam}

In this section we explore how the variation in estimated K_{cam} values affect the depth estimation performance. We use the same model that we used to obtain the results in Table 3 that was trained on data from K_{cam} values of 8.79 and 35.61 and evaluate them on data from K_{cam} values of 12.69 and 22.67. As can be seen in Figure 5, we use a range of numbers centered on the actual K_{cam} values for the respective datasets and obtain the RMSE error of depth estimation. It can be seen that the error response of the model to the variation of K_{cam} used has a clear minimum. The error increases if the values used in the place of K_{cam} deviates from the actual value. For example, the error of the response of $K_{cam}=23.67$ increases by around 16% if the K_{cam} used deviates positively from the GT values by 18%. Figure 8 shows some examples of predicted depth maps when the model was provided with an unseen virtually blurred image from a camera with $K_{cam} = 22.67$. Agreeing with the Figure 5, the predictions get distorted faster when the K_{cam} used lowers than the ground truth K_{cam} and distorts slower when it increases.

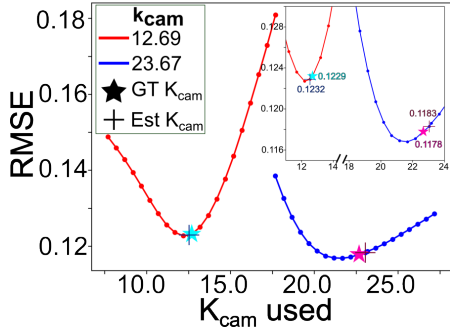


Figure 5. K_{cam} Estimation Error

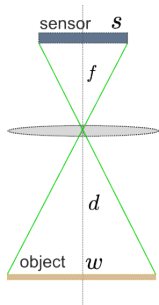


Figure 6. FOV of a camera

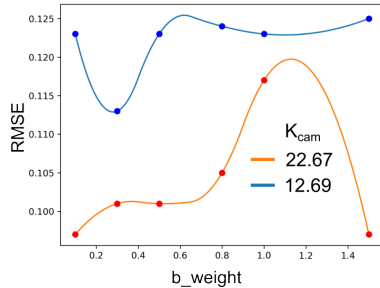


Figure 7. RMSE Vs b_weight

Method	MSE
in-focus	0.0640
No K_{cam}	0.0139
est K_{cam}	0.0096

Table 4. Performance on DDF12 dataset

K_{cam}	Image size	RMSE
12.69	Original size	0.123
	Resized	0.438
22.67	Original size	0.117
	Resized	0.399

Table 5. Effect of FOV

4.5. Effect of the blur weight

We change the scaling parameter b_weight from equation 8 while training several models on data from defocus blurred NYU depth v2 dataset with K_{cam} values of 8.79 and 35.61. The performance on the two evaluation datasets (with K_{cam} values of 12.69 and 22.67) are shown in Figure 7.

4.6. Effect of the Field of View

Field of view of a camera can be defined in several ways. One way is to define it as the size of an object at a given distance from the camera that would completely fill the image sensor. In Figure 6, s is the length of the sensor, f is the focal length of the lens, d is the distance to the object and w is the length of the object. The size w of an object that would completely fill the sensor will be given by $w = \frac{s}{f} \cdot d$. It can be seen that w is inversely proportional to f . Cameras with a smaller focal length have a larger Field of View and vice-versa. In all the experiments that we performed including the NYU dataset, we have assumed that the cameras have a fixed FOV even when the k_{cam} (and therefore f)

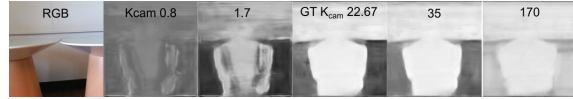


Figure 8. Examples of from the camera with $K_{cam} = 22.67$ predicted using various K_{cam} values

changes. While this is helpful to analyze the performance of blur based depth estimation methods, it is important to investigate the effect of the FOV change on the performance of the models. We created a dataset by scaling down the images in the NYU depth dataset by a factor of 0.6 and then refocusing with a k_{cam} (respective f is 30mm) of 12.69. The model has been trained with data having k_{cam} s of 8.79 and 35.61 (they had focal lengths (f) of 20mm and 50mm). We scaled down the images of $f = 30mm$ with respect to $f = 50mm$ which is 0.6. Note that the images have the same amount of blur as the images of original size; only the size of the objects visible have changed. From Table 5, it can be seen that the performance drops significantly by more than three folds when we perform the resizing. The reason for this can be understood from Figure 2. The depth estimation section receives two inputs. One is the blur and the second is the image features in the form of skip connections. Although we account for the change of blur through division by the respective k_{cam} , we do not modify image features to reflect the change of FOV. This is a limitation of our work and needs to be addressed in the future.

5. Conclusions

We show that estimating depth from defocus blur is significantly superior to conventional semantic based depth prediction provided that the camera is suitable for it. But this technique is sensitive to the camera. Our novel approach performs a simple correction to an already trained depth prediction model using camera parameters of a given camera. We show that this correction can alleviate the sensitivity of the model to the camera. Our novel defocus blur calibration technique can estimate the camera parameters using several images taken by a given camera. We show that our approach beats the state-of-the-art for several datasets. Finally we show some limitations of our work and suggest future improvements.

Methods such as ours which utilize a camera can come with certain privacy and security concerns. Our technique specifically can be used by enemy drones to measure distance to a target person in close quarters and cause significant harm which raises several ethical concerns. As researchers it is important to address these concerns alongside the algorithmic improvements to the state-of-the-art.

Acknowledgment: This work is supported by the award 70NANB21H029 from the U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

References

- [1] Saeed Anwar, Zeeshan Hayder, and Fatih Porikli. Depth Estimation and Blur Removal from a Single Out-of-focus Image. In *BMVC*, volume 1, page 2, 2017. 2
- [2] Yuxi Ban, Mingzhe Liu, Peng Wu, Bo Yang, Shan Liu, Lirong Yin, and Wenfeng Zheng. Depth estimation method for monocular camera defocus images in microscopic scenes. *Electronics*, 11(13):2012, 2022. 1, 3
- [3] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2
- [4] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. Deep Depth from Defocus: how can defocus blur improve 3D estimation using dense neural networks? In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, page 0, 2018. 6
- [5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelisa Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8001–8008, 2019. 1
- [6] Yu-Lin Chang, Wei-Yin Chen, Jing-Ying Chang, Yi-Min Tsai, Chia-Lin Lee, and Liang-Gee Chen. Priority depth fusion for the 2D to 3D conversion system. In *Three-Dimensional Image Capture and Applications 2008*, volume 6805, pages 320–327. SPIE, 2008. 2
- [7] Xiaodong Cun and Chi-Man Pun. Defocus blur detection via depth distillation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 747–763. Springer, 2020. 3
- [8] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16940–16961, 2022. 1
- [9] 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*, 27, 2014. 6
- [10] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7683–7692, 2019. 1, 2, 3
- [11] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 525–541. Springer, 2019. 2, 6, 7
- [12] Hayato Ikoma, Cindy M Nguyen, Christopher A Metzler, Yifan Peng, and Gordon Wetzstein. Depth from defocus with learned optics for imaging and occlusion-aware depth estimation. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2021. 2
- [13] Jeff Meyer and Alex Summersby. Image sensors explained. 4
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [15] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1558–1567, 2019. 2
- [16] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810, 2022. 2
- [17] Shaojun Liu, Fei Zhou, and Qingmin Liao. Defocus map estimation from a single image based on two-parameter defocus model. *IEEE Transactions on Image Processing*, 25(12):5943–5956, 2016. 2, 3
- [18] Yawen Lu, Garrett Milliron, John Slagter, and Guoyu Lu. Self-supervised single-image depth estimation from focus and defocus clues. *IEEE Robotics and Automation Letters*, 6(4):6281–6288, 2021. 3
- [19] Maxim Maximov, Kevin Galim, and Laura Leal-Taixé. Focus on defocus: bridging the synthetic to real domain gap for depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1071–1080, 2020. 1, 2, 3, 4, 5, 6, 7
- [20] Alican Mertan, Damien Jade Duff, and Gozde Unal. Single image depth estimation: An overview. *Digital Signal Processing*, page 103441, 2022. 2
- [21] S K Nayar and Y Nakagawa. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, 1994. 2
- [22] 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. 2
- [23] OpenCV. Camera Calibration, 2023. 5
- [24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6
- [25] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022. 2
- [26] S Pertuz, D Puig, M A Garcia, and A Fusiello. Generation of All-in-Focus Images by Noise-Robust Selective Fusion of Limited Depth-of-Field Images. *IEEE Transactions on Image Processing*, 22(3):1242–1251, 2013. 2
- [27] Jiamin Ping, Yue Liu, and Dongdong Weng. Comparison in depth perception between virtual reality and augmented reality systems. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)*, pages 1124–1125. IEEE, 2019. 1

- [28] Yoav Y Schechner and Nahum Kiryati. Depth from Defocus vs. Stereo: How Different Really Are They? *International Journal of Computer Vision*, 39(2):141–162, 2000. 2
- [29] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. *ECCV (5)*, 7576:746–760, 2012. 6
- [30] Murali Subbarao and Gopal Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994. 2, 3
- [31] Murali Subbarao and Gopal Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994. 3
- [32] M Subbarao and J . K. Tyan. Selecting the optimal focus measure for autofocusing and depth-from-focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):864–870, 1998. 2
- [33] Yu-Wing Tai and Michael S Brown. Single image defocus map estimation using local contrast prior. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1797–1800. IEEE, 2009. 3
- [34] Ning-Hsu Wang, Bolivar Solarte, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 360sd-net: 360 stereo depth estimation with learnable cost volume. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 582–588. IEEE, 2020. 2
- [35] Ning-Hsu Wang, Ren Wang, Yu-Lun Liu, Yu-Hao Huang, Yu-Lin Chang, Chia-Ping Chen, and Kevin Jou. Bridging unsupervised and supervised depth from focus via all-in-focus supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12621–12631, 2021. 1, 2
- [36] Masahiro Watanabe, Shree K Nayar, and Minori N Noguchi. Real-time computation of depth from defocus. In *Proc.SPIE*, volume 2599, pages 14–25, 1 1996. 1
- [37] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 611–620, 2020. 2, 6
- [38] Fengting Yang, Xiaolei Huang, and Zihan Zhou. Deep Depth from Focus with Differential Focus Volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2022. 1, 2
- [39] Anmei Zhang and Jian Sun. Joint Depth and Defocus Estimation From a Single Image Using Physical Consistency. *IEEE Transactions on Image Processing*, 30:3419–3433, 2021. 2, 3
- [40] Xiaolei Zhang, Huaying Wang, Wenjian Wang, Shaokai Yang, Jieyu Wang, Jialiang Lei, Zijian Zhang, and Zhao Dong. Particle field positioning with a commercial microscope based on a developed CNN and the depth-from-defocus method. *Optics and Lasers in Engineering*, 153:106989, 2022. 1
- [41] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *arXiv preprint arXiv:2303.02153*, 2023. 2, 6, 7
- [42] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011. 3