DeepLens: Shallow Depth Of Field From A Single Image

LIJUN WANG*, Dalian University of Technology
XIAOHUI SHEN, ByteDance AI Lab
JIANMING ZHANG, OLIVER WANG, and ZHE LIN, Adobe Research
CHIH-YAO HSIEH and SARAH KONG, Adobe Systems
HUCHUAN LU, Dalian University of Technology

Fig. 1. Each high resolution, shallow DoF image shown here was synthetically generated from an all-in-focus input image with a single forward pass through a neural network system. Users can freely interact with the system by modifying the focal plane and aperture size. Please view zoomed-in on screen, and use Acrobat Reader to watch the refocusing animations. *Photo credits: Franco Vannini, Jim Liestman, wombatarama, and European Space Agency.*

We aim to generate high resolution shallow depth-of-field (DoF) images from a single all-in-focus image with controllable focal distance and aperture size. To achieve this, we propose a novel neural network model comprised of a depth prediction module, a lens blur module, and a guided upsampling module. All modules are differentiable and are learned from data. To train our depth prediction module, we collect a dataset of 2462 RGB-D images captured by mobile phones with a dual-lens camera, and use existing segmentation datasets to improve border prediction. We further leverage a synthetic dataset with known depth to supervise the lens blur and guided upsampling modules. The effectiveness of our system and training strategies are verified in the experiments. Our method can generate high-quality shallow DoF images at high resolution, and produces significantly fewer artifacts than the baselines and existing solutions for single image shallow DoF synthesis. Compared with the iPhone portrait mode, which is a state-of-the-art shallow DoF solution based on a dual-lens depth camera, our method generates comparable results, while allowing for greater flexibility to choose focal points and aperture size, and is not limited to one capture setup.

CCS Concepts: • **Computing methodologies** → **Rendering**; *Machine learning approaches*; *Neural networks*;

Additional Key Words and Phrases: Shallow Depth of Field; Neural Network

ACM Reference Format:
Lijun Wang, Xiaohui Shen, Jianming Zhang, Oliver Wang, Zhe Lin, Chih-Yao Hsieh, Sarah Kong, and Huchuan Lu. 2018. DeepLens: Shallow Depth Of Field From A Single Image. *ACM Trans. Graph.* 37, 6, Article 245 (November 2018), 11 pages. https://doi.org/10.1145/3272127.3275013

*Work done while interining at Adobe Research.

Authors' addresses: Lijun Wang, Dalian University of Technology, wlj@mail.dlut.edu.cn; Xiaohui Shen, ByteDance AI Lab, shenxiaohui@bytedance.com; Jianming Zhang, jianmzha@adobe.com; Oliver Wang, owang@adobe.com; Zhe Lin, zlin@adobe.com, Adobe Research; Chih-Yao Hsieh, hsieh@adobe.com; Sarah Kong, sakong@adobe.com, Adobe Systems; Huchuan Lu, Dalian University of Technology, lhchuan@dlut.edu.cn.

1 INTRODUCTION

The shallow depth-of-field (DoF) effect is an important technique in photography which draws the viewers' attention to the region of focus by blurring out the rest of the image. Typically, these images are captured with expensive, single-lens reflex (SLR) cameras and large-aperture lenses. This makes their acquisition less accessible to casual photographers. In addition, once captured, these images can not be easily edited to refocus onto other regions, or to change the amount of defocus.

In this work, we propose to train a neural network system which generates synthetic shallow DoF effects on all-in-focus photos captured by ordinary cameras or mobile devices. Our network is composed of a depth prediction module for single image depth estimation, a lens blur module for predicting spatially-varying blur kernels, and a guided upsampling module for generating high-resolution shallow DoF images.

The whole network is fully differentiable and therefore allows end-to-end training. However, the required training data (all-in-focus and shallow DoF image pairs) are hard to capture for diverse scenes with non-static content. One potential solution for data collection is to generate ground truth shallow DoF images of real scenes with existing rendering methods that leverage depth maps. However, as image-based methods suffer from various artifacts (*c.f.* Fig. 3), the rendered ground truth contains artifacts. An alternative solution is to adopt artifact-free synthetic data, but then the challenge is to generalize this data to real world images. Based on these discussions, we propose to train the network in a piece-wise manner. Specifically, since depth prediction requires semantic understanding of the scene, we train the depth prediction module on real images to allow for generalization. In comparison, rendering DoF effects based on

the predicted depth mainly involves low-level operations which are consistent across real and synthetic data. To alleviate artifacts, the lens blur and guided upsampling modules are therefore jointly trained on high-quality synthetic images.

To train our depth prediction module, we collect a new dataset consisting of 2462 RGB-D images of diverse content from real-world environments. The RGB-D images are acquired by the dual-lens camera on an iPhone. Compared with existing RGB-D datasets [Geiger et al. 2012; Silberman et al. 2012; Srinivasan et al. 2017], our proposed dataset contains more diverse contents and is more focused on everyday scenes and common photography subjects. We further leverage existing salient object datasets [Cheng et al. 2015; Wang et al. 2017] to augment the depth training with an auxiliary foreground segmentation task, which improves the quality at object boundaries, and makes the depth prediction module better generalize to diverse content and image types.

The lens blur module together with the guided upsampling module generates high resolution shallow DoF images based on predicted depth maps. To jointly train the two modules, we introduce a synthetic dataset consisting of scenes with known depth and occlusion information, which can be used to render shallow DoF supervision without artifacts. We further randomly distort the known depth during training to make our lens blur module and guided upsampling module robust to inaccurate depth maps. After training, the guided upsampling module can be recurrently applied on the output of the lens blur module to generate high-resolution results of arbitrary resolutions (we show results up to 2k in this paper).

We conduct user studies to show that our results significantly outperform existing single image-based solutions, and are comparable to DoF images rendered by the iPhone portrait mode, which requires depth inputs from a dual-lens camera. Our method can be used with any camera, and allows users to control the focal point and blur amount at interactive rates (*e.g.*, 0.7s for 2048 resolution). We further show that our method can be applied in more general scenarios such as historic photos and even paintings.

In summary, we present the following contributions:

(1) A learning-based DoF rendering method that can be trained to produce fewer artifacts than existing hand-designed lens blur methods, while being more robust to inaccurate depth maps. This is trained with a loss designed to concentrate the training on depth discontinuity boundaries.
(2) A memory-efficient network architecture that performs the lens blur operation in a compressed feature space and recurrently applies guided upsampling to generate results at high resolution.
(3) A carefully-designed training scheme using a combination of real and synthetic data for the general single image shallow DoF synthesis.

2 RELATED WORK

Depth reconstruction is a classic problem in computer vision with a large body of literature. Here, we review learning-based single image depth estimation methods, as well as related works that address depth in the context of lens blur effects. We will also cover related works in lens blur rendering and defocus estimation.

Single Image Depth Estimation. CNNs have proven to be powerful tools to learn strong scene priors for single image depth prediction, usually by training to reconstruct the output of depth cameras [Eigen et al. 2014; Laina et al. 2016; Liu et al. 2015]. However, depth cameras often have limitations in terms of range, image quality, portability, and resolution. As a result, models trained on raw depth data often only work in limited scenarios (*e.g.*, indoor rooms [Silberman et al. 2012], street scenes [Geiger et al. 2012], landmarks [Li and Snavely 2018]). In [Srinivasan et al. 2018], aperture stacks are used as supervision for depth prediction, which demonstrates its effectiveness on flower images and indoor scenes. However, only using aperture supervision can suffer from depth ambiguities, especially with diverse scenes, and we found in our experiments that we were not able to generate high quality results by using aperture supervision alone .

Multi-Images Depth Estimation. Using correspondences across multiple images is the most common way to acquire depth information, and several methods have used this in the context of shallow DoF effects. For example, depth can be created from focal stacks on mobile phones [Suwajanakorn et al. 2015], stereo camera pairs [Barron et al. 2015], small baseline burst image capture [Ha et al. 2016; Joshi and Zitnick 2014], or multiple frames of video [Klose et al. 2015]. These methods all perform a two stage process, where the pre-computed depth and an in-focus image is used to synthesize shallow DoF results. As opposed to this, we compute depth from a single image, and introduce a differential shallow DoF renderer, which is trained to be robust to errors in estimated depth.

Lens Blur Effects. Lens blur effects that operate on a single RGB-D image cannot generate an exact result, as defocus blur requires seeing *behind* objects. Existing methods that approximate the lens blur can be generally classified into object space and image space. Object space methods [Haeberli and Akeley 1990; Lee et al. 2010] explore ray tracing and real camera models, and are more effective in rendering realistic shallow DoF images, but most of these methods are time-consuming and entail 3D scene representation which is hard to achieve. Image space methods [Kraus and Strengert 2007; Lee et al. 2009; Yang et al. 2016] operate on a single image, and cast shallow DoF effects rendering as a post-processing step with gathering and spreading operations. These methods are more efficient and can be converted to a fixed but differentiable layer, as has been done in the compositional aperture rendering [Srinivasan et al. 2018]. However, this approach uses fixed convolutional kernels (discs) at discrete depths and blends them to generate a final image. This can lead to results that suffer from intensity leakage at depth discontinuities, and cannot handle foreground occluders in the defocused regions well. As opposed to these methods, our approach uses a lens blur module that is motivated by real lens optics, and can produce effects such as an out-of-focus foreground, which is more challenging to generate.

Current high end mobile phones, such as the iPhone and Google Pixel 2, have a "portrait mode" application, which simulates shallow DoF images. These are based off of depth measurements from multiple views or dual-pixels, or from a simpler foreground segmentation mask. Moreover, users are unable to control the focal point or the blur amount in the current portrait mode application.
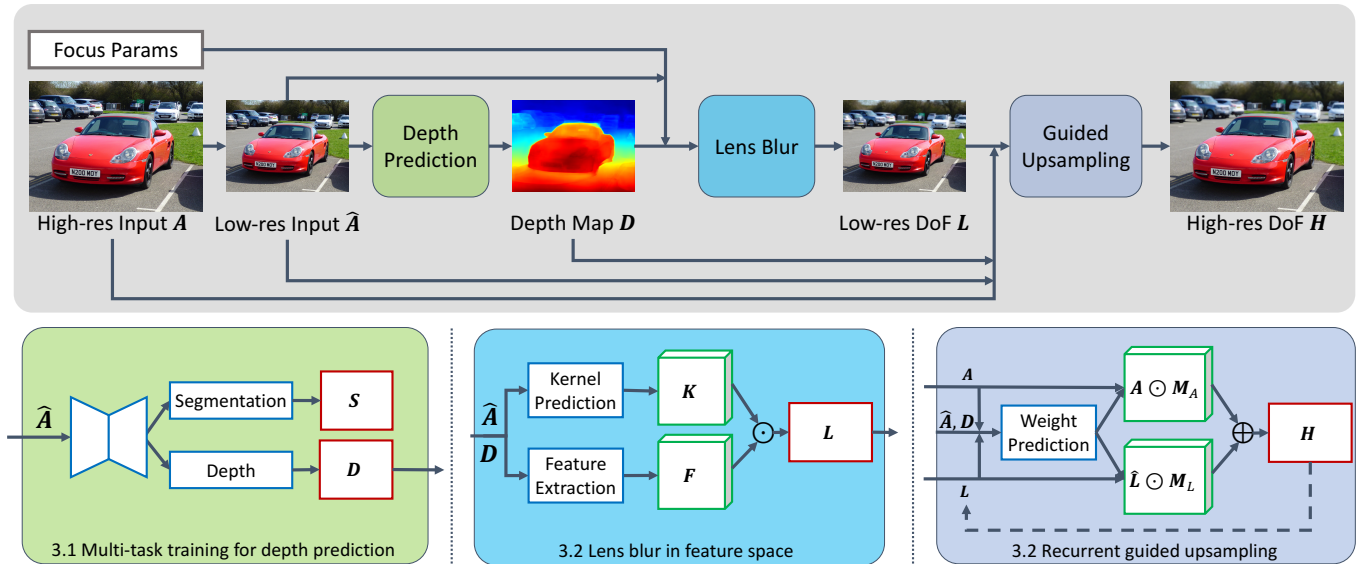
Fig. 2. **Method Overview**. At a high level (top), we predict a low resolution depth map D from the all-in-focus input $\hat{A}$. D, $\hat{A}$, and the focus parameters are used to generate a low resolution shallow DoF image L, which is then upsampled into a final high resolution output H. In more detail (bottom), the depth prediction module takes $\hat{A}$ and jointly predicts depth D and segmentation S maps, both of which have supervision. The lens blur module computes spatially varying kernels K which are applied to a reduced dimension feature map F to produce L. The guided upsampling module recurrently super-resolves the DoF image by a factor of $\times 2$ until it reaches the original resolution. Red blocks indicate supervision, blue blocks are learnable modules, and green blocks are intermediate feature maps. See the supplementary materials for more details of the network architecture. *Photo credits: grassrootsgroundswell.*

In contrast, our method allows users to modify the shallow DoF parameters interactively, and can generate results with comparable quality *without* a dedicated hardware depth module.

3 METHOD

While it is possible to cast the shallow DoF rendering problem as a black-box input-to-output regression problem [Isola et al. 2017], we found this approach did not generate satisfactory results. Instead, we propose an approach motivated by physical models that consists of three modules: **depth prediction**, **lens blur**, and **guided upsampling**. Fig. 2 shows an overview of our approach. We now describe each step in the model, but please refer to the supplementary materials for detailed network architectures.

3.1 Depth Prediction

We use a fully convolutional network architecture for depth prediction, trained on our newly collected RGB-D dataset (see Section 4.1). To further increase the generalization ability of our model across different image types, we augment the training with an additional foreground segmentation task, using [Cheng et al. 2015; Wang et al. 2017] The auxiliary segmentation task also enforces more accurate object boundary delineation in depth prediction, which is essential to shallow DoF rendering.

The depth prediction module is shown in Fig. 2. It consists of an encoder and a decoder followed by multi-task heads. The encoder extracts features using the first 14 residual blocks (from Conv1 to Res4f) of the ResNet-50 [He et al. 2016] network pretrained on ImageNet [Deng et al. 2009], with atrous convolution layers on the

last block to preserve feature resolution. It is then followed by a four-level pyramid pooling layer [Zhao et al. 2017] with bin sizes of 1×1, 2×2, 4×4 and 8×8, respectively. The decoder then upsamples the feature maps with three $\times 2$ bilinear upsampling layers, with skip connections from the encoder at each level. At the end of the decoder, the network is split into two small multi-task heads, each consisting of three convolutional layers. These heads generate depth prediction and foreground segmentation respectively.

Shallow DoF rendering only relies on depth ordinal information, making the reconstruction of the *absolute* depth values unnecessary. We therefore invert the depth to compute the disparity which is further normalized to the range of $[0, 1]$ by the minimum and maximum values. Both the ground truth and the network prediction are normalized in the same manner. We use "depth prediction" to refer to the generation of these normalized inverted depth maps in the rest of the text.

The network is then trained in a multi-task manner using the following loss function:

$$J_d(\theta_d) = \|D - D^g\|_1 + \gamma \times \|S - S^g\|_1, \tag{1}$$

where D and S denote the predicted depth map and foreground segmentation map, respectively; D^g and S^g are the corresponding ground truth; γ represents a trade-off parameter to balance the two tasks. We note that D^g and S^g come from different datasets, and each mini-batch during training contains images from both datasets. During the test phase, we only maintain the depth prediction stream, while the foreground segmentation stream is discarded.

| Input | Compositional[2018] | [Yang et al. 2016] | Ours |

Fig. 3. Comparison of lens blur methods. Our approach produces a more accurate focus/defocus boundary, while the compositional approach, and the ray tracing method of Yang et al. [2016] over-blurs high contrast regions. *Photo credits: mrchrishill and Kris Arnold.*

3.2 Lens Blur Rendering

Given the predicted depth map along with the original image, existing image space or object space rendering methods could be utilized to approximate lens blur effects [Lee et al. 2010; Srinivasan et al. 2018; Yang et al. 2016]. However, these methods often produce artifacts around object boundaries. Therefore, to obtain higher quality results, we use a differentiable neural network to approximate the spatially varying lens blur kernel, and train this network on a new artifact-free synthetic dataset (see Section 4.2). We compare the results of our lens blur approach with the ones generated by other lens blur rendering methods in Fig. 3, where all methods are given the same input image and depth map. We can see that our learned lens blur module produces more accurate results with much fewer artifacts.

One challenge with this approach is that it is prohibitive to directly predict and apply the learned spatially varying lens blur kernel on a large image due to memory consumption. Consider an input image of $H \times W$ resolution and a maximum blur kernel size of $k \times k$. This would require computing an $H \times W \times k^2$ kernel tensor. Applying a spatially varying filter with a kernel size $k = 65$ on a 1280×1280 3-channel image would require $1280^2 \times 65^2 \times 4 = 25.79$ GB memory, and 19.34 GFLOPS at inference time[1]. To reduce memory and computation, we operate on a lower resolution image ($h \times w$ where $h < H$ and $w < W$) and rely on the subsequent recurrent guided upsampling module to generate high-resolution results. Furthermore, the lens blur module operates as 1×1 filters in a *learned feature space* with $c << k^2$ dimensions, instead of directly applying blur in the image space. As a consequence, we only have to predict a kernel tensor of $h \times w \times c$.

3.2.1 Lens Blur in Learned Feature Space. The lens blur module is comprised of a feature extraction network and a kernel prediction network. The feature extraction network takes as input the low resolution all-in-focus RGB image of size $h \times w$ and produces a

---

[1]Filtering the image with spatially varying kernels amounts to an element-wise multiplication between the kernel and the image tensor. Since both tensors are of size 1280×1280×65²×3, the total computational complexity is 1280×1280×65²×3 = 19.34 GFLOPS.

feature map of size $h \times w \times c$. It consists of four convolution layers interleaved by ReLU units. To capture multi-scale features, the final feature maps are obtained by concatenating all intermediate feature maps, which mainly encodes low-level features around each local image patch (*e.g.*, color, blur, texture, *etc*).

Given the depth map, focal depth and aperture radius, the kernel prediction network is used to infer 1×1 blur kernels for each image location by generating a $h \times w \times c$ kernel tensor, which is then applied to the extracted feature map of each color channel to render a shallow DoF effect. Formally, given the extracted feature $F_i \in \mathbb{R}^{h \times w \times c}$ for color channel $i \in \{R, G, B\}$, and the predicted kernel tensor $K \in \mathbb{R}^{h \times w \times c}$, the shallow DoF image is rendered by:

$$L_i(x, y) = \sum_{j=1}^{c} K(x, y, j) \times F_i(x, y, j), \qquad (2)$$

where $L_i(x, y)$ denotes the color value at location (x, y) of the rendered DoF image. Since the kernel prediction and feature extraction networks are jointly trained with shallow DoF supervision, they learn in a data-driven fashion to generate desired kernels and feature space, leading to high quality shallow DoF results.

As the predicted depth itself may not be accurate, relying only on the depth to infer blur kernels increases the risk of producing unpleasant artifacts. Therefore, the final kernel prediction network also takes the all-in-focus image as input, which allows it to correct for errors in the predicted depth map based on the image context. This results in more robust inference. Please refer to the supplementary material for more network architecture details.

3.2.2 Control Parameters for Focal Depth and Aperture Size. The lens blur network requires both the focal depth and aperture radius to generate an output image. We encode focal depth with a signed depth map $\tilde{D}$ by subtracting the focal depth from the predicted normalized depth map: $\tilde{D} = D - d_f$, where d_f denotes the focal depth (0 implies that depth is in focus).

In order to encode the aperture size into the network input, one could simply tile the desired aperture size and concatenate it with the signed depth map. However, we found that this approach led to unsatisfactory results, as it was difficult for the network to learn the correlation between the blur amount and the value of aperture radius. We instead propose a simpler and more effective alternative moviated by the physics of DoF, where the network takes the signed depth map as input, and is trained to produce shallow DoF images for the *largest* aperture radius handled by our method, r^m. To successfully do this, the network must learn to add different amounts of blur based on the distance to the focal plane. At test time, we then allow the user to control the blur amount by *scaling* the input depth map with the ratio $\alpha = \frac{r^*}{r^m}$, where $r^* \leq r^m$ denotes the target aperture radius. We find that this method leads to much more robust training and allows for continuous aperture radius control at test time.

3.2.3 Shallow DoF Upsampling. One could employ an out-of-the-box super-resolution method to increase output resolution, but in our case we have additional information in the original high resolution all-in-focus image. Most existing super-resolution methods do not assume this additional information and therefore cannot
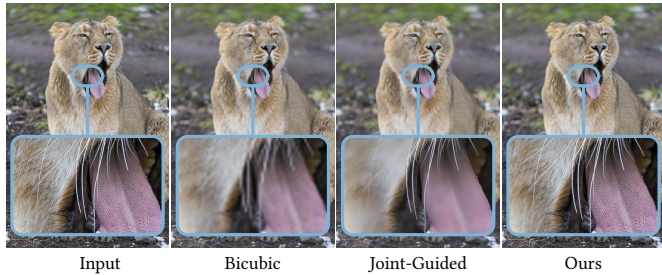
Fig. 4. Comparison of different upsampling methods. Ours preserves details in the in-focus region without introducing artifacts. *Photo credit: Tambako The Jaguar.*

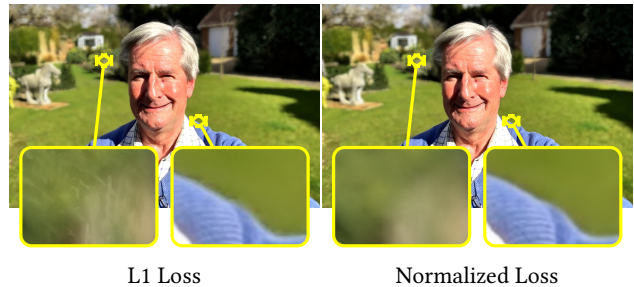L1 Loss　　　　　　　　　Normalized Loss

Fig. 5. Comparison of different loss functions. Our normalized loss reduces the artifact on the focus/defocus boundary, as well as in high contrast regions in the background. *Photo credit: Herry Lawford*

easily restore the clear in-focus regions. Another possible solution would be to use a joint guided upsampling approach [He et al. 2010] with the high-resolution all-in-focus image as guidance. However, as this method is not specifically designed for our task, it results in a sub-optimal solution, oversmoothing the in-focus region, and creating artifacts in the defocused region where the guidance image has strong texture (Fig. 4).

A key observation is that we should upsample the in-focus regions by leveraging the all-in-focus guidance image, while the blurry defocused regions should instead rely more on the low-resolution shallow DoF image. Motivated by this, we propose a guided upsampling network, which extracts features separately from the low-resolution space (*i.e.*, the low-resolution input image, the predicted DoF image and the depth map) and the high-resolution space (*i.e.* the high-resolution all-in-focus image). The low-resolution features are upsampled to the same resolution of the high-resolution ones using bilinear upsampling. The network then predicts two high-resolution spatial weight maps M_A and M_L, from the concatenation of high and low-resolution features. The final high-resolution shallow DoF image H is achieved through a weighted combination of the high-resolution all-in-focus image and the upsampled shallow DoF image:

$$H = M_A \odot A + M_L \odot \hat{L}, \tag{3}$$

where $\odot$ denotes element-wise multiplication, A denotes the high-resolution all-in-focus image, and $\hat{L}$ represents the shallow DoF image upsampled from the low-resolution version L (Equation (2)) using bilinear interpolation. As shown in Fig. 4, our guided upsampling module can render more pleasant results than single image super resolution and guided filtering.

3.2.4 Joint Training. Since both lens blur and guided upsampling modules are fully differentiable, we jointly train the two parts by minimizing the differences between the produced low and high-resolution results and the corresponding ground truth. However, for rendering the shallow DoF images, the most challenging parts of the scene are at object boundaries with depth discontinuities. These regions only constitute a small portion of the whole image but significantly contribute to the perceived quality. Traditional ℓ_1 and ℓ_2 losses are unable to capture these challenging regions, leading to severe artifacts around object boundaries, as shown in

Fig. 5. Therefore, we propose to use the following normalized loss:

$$J_f(X|\theta) = \frac{\sum_i w_i L(X_i, Y_i)}{\sum_i w_i}, \tag{4}$$

where X and Y denote prediction and ground truth, respectively; i indicates spatial index; L represents some pixel-wise loss; and θ denotes network parameters to be optimized. The loss weight w_i for location i is computed as the local error $(|X_i - Y_i|)^\alpha$. The hyper-parameter α is empirically set to 1.5 and works well. In our experiments, we adopt the absolute difference as the pixel-wise loss L. By substituting L and w_i, Equation (4) can be written as

$$J_f(X|\theta) = \frac{\sum_i |X_i - Y_i|^{\alpha+1}}{\sum_i |X_i - Y_i|^\alpha}. \tag{5}$$

Unlike a standard ℓ_p loss, this normalization places more emphasis on pixels with large errors, *i.e.*, hard examples, and ignores easy cases (Fig. 5).

Both the lens blur and guided upsampling modules are then jointly trained using the normalized loss as follows:

$$\arg\min_{\theta_l, \theta_h} J_f(L|\theta_l) + \beta \times J_f(H|\theta_h), \tag{6}$$

where the low-resolution prediction L and its high-resolution counterpart H are computed using Equation (2) and (3), θ_l and θ_h denote network parameters of lens blur and guided upsampling module, respectively. The loss is optimized by stochastic gradient solver. More implementation details about network training and architecture can be found in the supplementary material.

4 DATASETS

In order to train the proposed method, we require training samples consisting of an all-in-focus and shallow DoF image pair, the corresponding focal depth, aperture size, and ground truth depth map. As collecting data with all of these characteristics is challenging, we combine different sources of data to supervise the different components of our method.

4.1 iPhone Depth Dataset

Existing depth prediction and shallow DoF datasets are very limited in terms of scene diversity (*e.g.*, indoor rooms [Silberman et al. 2012], street scenes [Geiger et al. 2012], and flower images [Srinivasan et al. 2017]), which are not suitable for DoF effect synthesis in general

scenes. To facilitate network training, we collect a new data set of RGB-D images using the dual-lens camera that comes with recent iPhones. We developed a custom iOS app that saves the images and depth maps from the iPhone camera, and deployed it to 15 users who helped capture a wide variety of scenes in different cities. After manually filtering out inaccurate depth samples, we finally obtain 2462 RGB-D images, (depth maps are computed at a lower 768×1024 resolution). The dataset is randomly split into a training set with 2262 images and a test set with 200 images. A manual check is performed to guarantee that there is no near-duplicate samples between the training and test sets. Fig. 7 shows two examples of the test images and depth maps.

Given the input RGB images and depth maps, we could render approximate shallow DoF images for supervision using a ray-tracing based rendering method [Yang et al. 2016]. However the rendered DoF images are not artifact-free, due to i) the noise in the captured depth maps, and ii) the depth discontinuity artifacts caused by the missing appearance information of the occluded background that is needed for the exact rendering. Consequently, we found that training our rendering modules on this dataset led to sub-optimal results. Thus, we resort to a synthetic dataset for training our lens blur module and guided upsampling module.

4.2 Synthetic Shallow DoF Image Data Set

In order to achieve artifact-free training data, we introduce a cost-effective synthetic shallow DoF dataset consisting of both all-in-focus images and depth maps where we can ensure that i) the depth maps are accurate ii) the appearance information of occluded regions is available, and iii) the focal plane is known.

To this end, we select 300 training images from our iPhone depth data set, which do not contain any foreground objects and serve as the background images. For foreground objects, we augment the matting data set from [Xu et al. 2017] with additional manually annotated data, which in total consists of 3662 foreground images and the corresponding binary masks. To synthesize an all-in-focus image, we randomly select one background image and two foreground objects, and composite them together, forming three depth planes. The background depth are either all zero or smoothly vary from b_0 to b_1 either horizontally or from bottom to top, where $b_0 > b_1 \in [0, 1]$. The depth within two foreground planes are homogeneous with values $f_0 > f_1$, where f_0 and f_1 denote the depth values of objects within the front and middle planes, respectively, and are randomly selected from the range of $[0, 1]$, Given the synthesized image, depth map, and a random focal depth, we then render shallow DoF effects using the method [Yang et al. 2016] *separately* on the three image planes and composite them together from back to front to generate the final shallow DoF image. Since the depth values within each foreground region are assumed to be homogeneous, depth discontinuities only occur at object boundaries. This compositing based approach effectively avoids discontinuity artifacts that arise from image-based shallow DoF rendering approximations, as content *behind* the foreground occluders is known, and can be correctly modeled by ray tracing rather than hallucinated.

According to the above rules, we have synthesized 18K training samples and 500 test ones. Fig. 6 shows two examples of the synthetic

Fig. 6. Examples from the synthetic dataset.

data. While this data does not look realistic, we find that the lens blur and guided upsampling module trained on the synthetic data can generalize well to real images. This is because rendering shallow DoF effects mainly involves *low-level* image operations, so the high level scene context is less important. We also find randomly corrupting the foreground depth plane through dilation or erosion to be an effective data-augmentation strategy during training. As shown in our experiments, training on corrupted depth can further improve the robustness against inaccurate depth prediction and forces the lens blur network to make prediction also based on input images rather than depth only.

Besides using corrupted depth, we have also explored training with depth maps generated by our depth prediction module. However, since our depth prediction module is trained on real images, it fails to generalize well to synthetic scenes. As a consequence, we train an additional depth prediction module on the synthetic data set. We then use the synthetic predicted depth as input to train the lens blur and guided upsampling. Our experiments (Section 5.3) show that training with predicted depth achieves similar performance to training with corrupted depth.

5 EXPERIMENTS

5.1 Depth Prediction

We first evaluate the performance of our depth prediction module. We compare our approach with two state-of-art methods including [Laina et al. 2016] and MegaDepth[Li and Snavely 2018], which are trained to predict absolute depth values on the NYU v2 data set [Silberman et al. 2012] and the MegaDepth data set [Li and Snavely 2018], respectively. In addition, we also implement a baseline method named Aperture, which has the same network architecture as ours and is trained with supervision only on DoF images, similar to the method in [Srinivasan et al. 2018]. For a fairer comparison, we follow [Srinivasan et al. 2018] and fit a 5-knot linear spline to minimize the squared error of each prediction with respect to the ground truth. This is needed to correct for scale differences. Especially for aperture supervision, which can produce an inverted depth map.

Fig. 7 and 8 illustrate qualitative comparisons on the iPhone test set and random internet images, respectively. It can be observed that
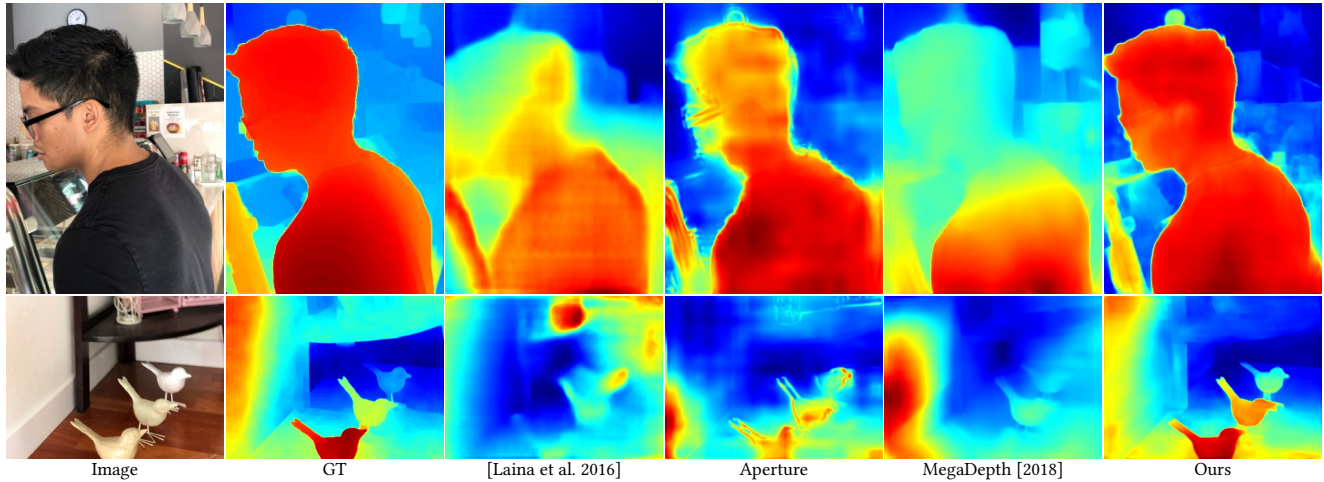
| Image | GT | [Laina et al. 2016] | Aperture | MegaDepth [2018] | Ours |

Fig. 7. Predicted depth on iPhone test set.

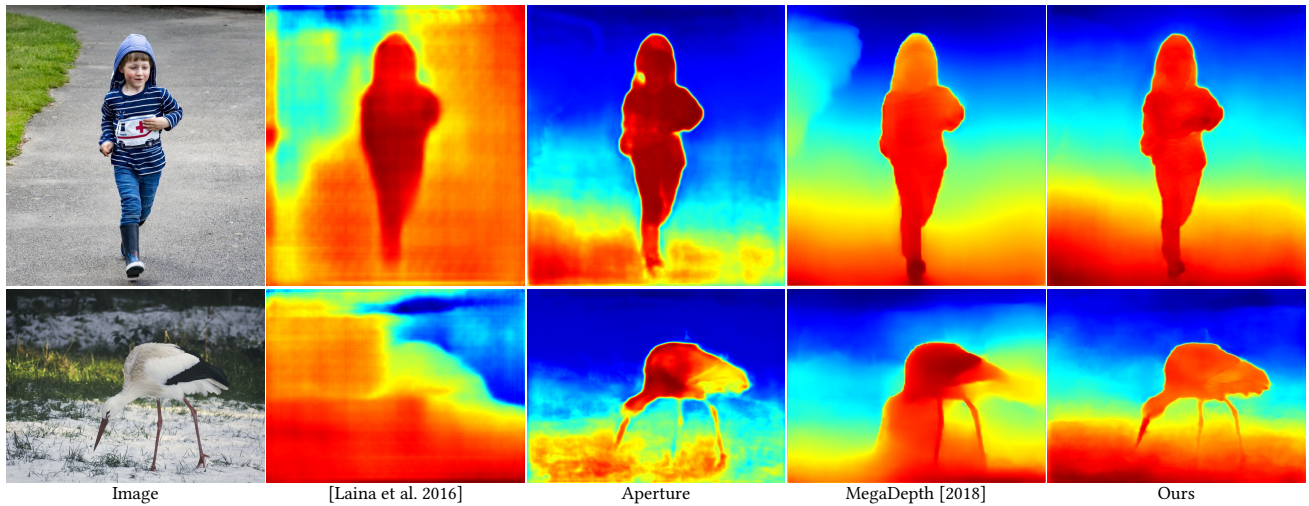| Image | [Laina et al. 2016] | Aperture | MegaDepth [2018] | Ours |

Fig. 8. Predicted depth of images from internet. *Photo credits: Ashley Buttle and Torsten Behrens*

Table 1. Comparison of depth prediction methods on our iPhone depth test set in terms of mean absolute error (MAE). Results in parentheses are achieved after training the corresponding methods on our iPhone depth training set, and the Aperture model is trained end-to-end using shallow DoF supervision.

Method	[Laina et al. 2016]	MegaDepth [2018]	Aperture	Ours
MAE	0.199 (0.138)	0.182 (0.142)	0.146	0.116

the models trained on existing depth datasets struggle to generalize to both our test set and internet images, while our method trained on the proposed depth data set achieves perceptually more accurate results across the test images. Compared with the Aperture baseline, which is our model trained end-to-end with supervision on shallow DoF images, we observe that our method more clearly delineates object boundaries.

A comparison of results are reported in terms of mean absolute error (MAE) in Tab. 1. We report the performance of [Laina et al. 2016] and MegaDepth [Li and Snavely 2018] trained on their proposed datasets and tested on our iPhone dataset, as well as trained on our same training set.

5.2 Evaluation on DoF results

To evaluate the quality of our generated shallow DoF results, we compare to a strong baseline consisting of a state-of-the-art single image depth prediction method, MegaDepth [Li and Snavely 2018], and an approximate ray tracing based shallow DoF effect method from [Yang et al. 2016]. For a fair comparison, we use the model of MegaDepth trained on our iPhone depth data set. We also compare to direct DoF image regression (without explicitly inferring the depth map), using a state-of-the-art image-to-image translation method [Isola et al. 2017], which has previously shown

Fig. 9. Qualitative comparison of shallow DoF results on images from internet (Cropped). First row: input image; second row: results of direct regression; third row: results of Aperture; fourth row: results by MegaDepth [2018]+[Yang et al. 2016]; last row: our results. Please zoom-in to see details. *Photo credits: Erin Page, daveynin, mrchrishill and Jewish Women's Archive.*

successful in blur magnification to generate shallow DoF images. To train the image-to-image translation model, we feed the focal point, desired aperture size and the source image to the network, and ask the network to predict the resulting shallow DoF image. In addition, we have also compared to the Aperture baseline (See Section 5.1), which combines our depth prediction network with a differentiable compositional DoF rendering approach [Srinivasan et al. 2018], and is trained in an end-to-end manner with DoF supervision. We quantitatively evaluate all compared methods on the iPhone test set, with ground truth rendered by [Yang et al. 2016] based on ground truth iPhone depth maps. The comparison in terms of PSNR and SSIM are shown in Tab. 2, and qualitative results on internet images are illustrated in Fig. 9. Our method has achieved superior performance against the other baselines, both quantitatively and qualitatively. This is in part due to the multi-task training on a

segmentation task, and our new iPhone Depth data set with more diverse scenarios. The direct regression method often fails to render shallow DoF effects and suffers from severe artifacts, while results generated by Aperture and the combination of two state-of-the-art prior works are more perceptually plausible, but have artifacts in the predicted depth discontinuities, which affect the shallow DoF rendering. Note that the quantitative evaluation on iPhone test set would be inevitably affected by the artifacts in the rendered ground truth as mentioned in Section 4.1. Therefore, we complement the evaluation with a user study on internet photos (Section 5.4).

5.3 Ablative Study

To understand the contribution of each module, we compare our final method to another two baselines. In the first baseline, we replace the depth prediction network with the Aperture method

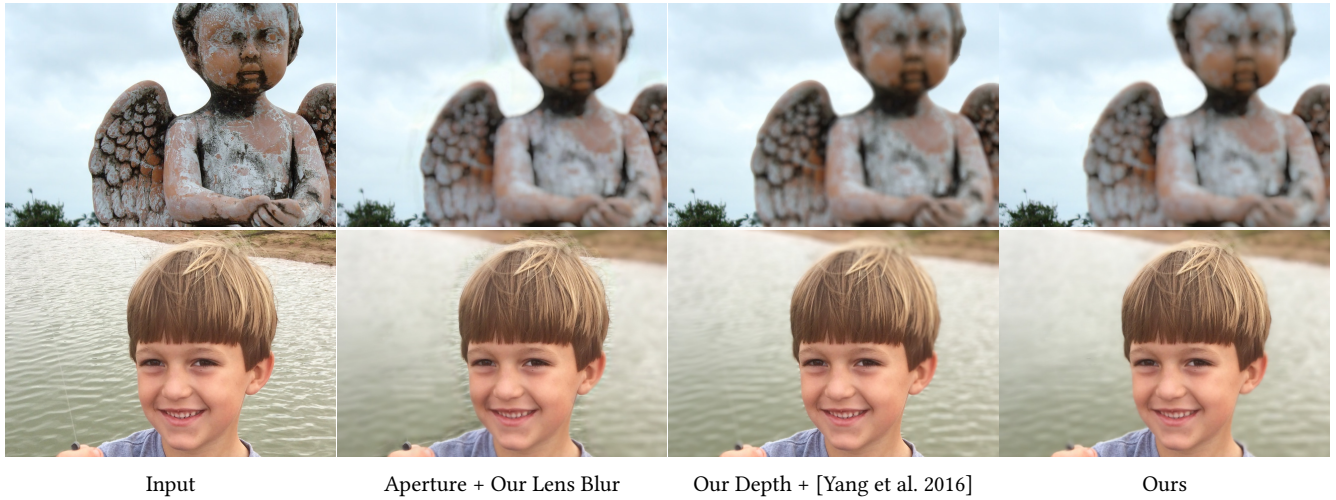| Input | Aperture + Our Lens Blur | Our Depth + [Yang et al. 2016] | Ours |

Fig. 10. Qualitative comparison of shallow DoF results (Cropped). *Photo credits: Christine and mrchrishill.*

Table 2. Comparison of DoF results on iPhone test set.

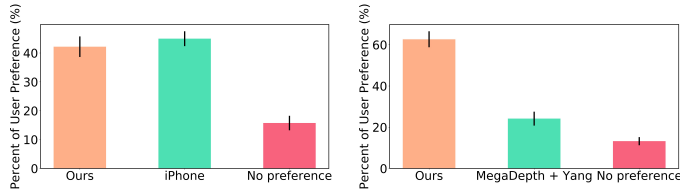| Method	Direct Regression	Aperture	MegaDepth[2018] + [Yang et al. 2016]	Ours
PSNR	25.84	26.88	26.67	28.235
SSIM	0.815	0.844	0.836	0.908

Fig. 11. The results of our user study. Our method is compared to iPhone portrait mode images (left), and the ray-tracing based approach [Yang et al. 2016] (right). Note that the both the iPhone and [Yang et al. 2016] are computed with measured depth, whereas our method uses only a single image. Error bars show 95% confidence intervals.

Table 3. Comparison of DoF results on iPhone test set. RealDepth and SynDepth denotes the depth prediction modules which are trained on the iPhone depth dataset and synthetic data set, respectively. GT, Pred, and Corrupted indicate that the lens blur and guided upsampling modules are trained with ground truth, predicted and corrupted depth, respectively.

Method	Baseline1	Baseline2	Baseline3	Ours
Depth Module	RealDepth	RealDepth	SynDepth	RealDepth
Training Input	GT	Pred	Pred	Corrupted
PSNR	28.038	28.237	22.380	28.235

(Section 5.1) and use our lens blur and guided upsampling to render shallow DoF, which we refer to as Aperture + Our Lens Blur. In the second baseline, we combine our depth prediction network and the DoF rendering method [Yang et al. 2016], which we refer to as Our Depth + [Yang et al. 2016]. As shown in Fig. 10, Aperture + Our Lens Blur performs poorly mainly due to inaccurate depth predictions, suggesting that that depth supervision is still essential for high quality results. With more accurate depth prediction, Our Depth + [Yang et al. 2016] method works well, however, it still suffers from artifacts at depth discontinuities on the foreground boundary. In comparison, our lens blur method is trained on an artifact-free data set and takes both depth and images as input, which allows it to compensate subtle errors in depth prediction, and avoid common artifacts of image-based shallow DoF approximations.

Recall that our lens blur module is performed in a learned feature space. To investigate the effect of feature dimension on the final

results, we evaluate the performance of our method with respect to different numbers of feature channels on the test set of the synthetic shallow DoF dataset. As we see in Fig. 12, the quality of the rendered results increases with respect to the channel number and begins to saturate at 32 channels. For both efficiency and effectiveness, we adopt 32 channels in our final models, which amounts to total of 12.5% of the kernel size represented, showing that we achieve a large memory savings with almost no quality loss by operating in a learned feature space.

To analyze the impact of learning lens blur and guided upsampling from corrupted depth maps (Section 4.2), we compare our method with three baselines. The depth prediction modules of the first two baselines and our method are the same and they are trained on the iPhone depth data (denoted as the RealDepth models). However, for training the lens blur and guided upsampling modules, the first baseline uses the ground truth synthetic depth, and the second baseline uses the depth generated by a depth prediction model trained on the synthetic data, denoted as the SynDepth model. As shown in Tab. 3, our method trained with corrupted depth achieves similar performance to the second one trained with predicted depth, which is superior to training with ground truth depth. This indicates that it can be sub-optimal to train the lens blur and guided upsampling modules on the ground truth depth because the networks can be sensitive to the errors made by the depth prediction module.
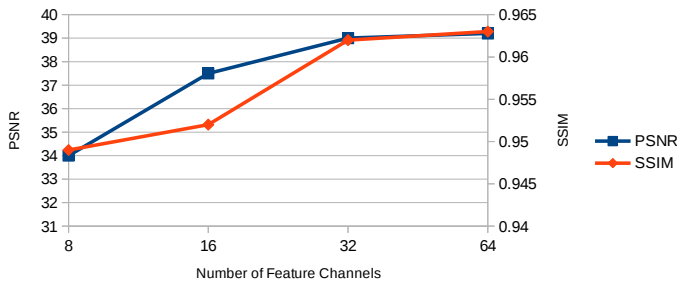
Fig. 12. PSNR and SSIM of our method with respect to numbers of feature channels on the test set of the synthetic shallow DoF dataset

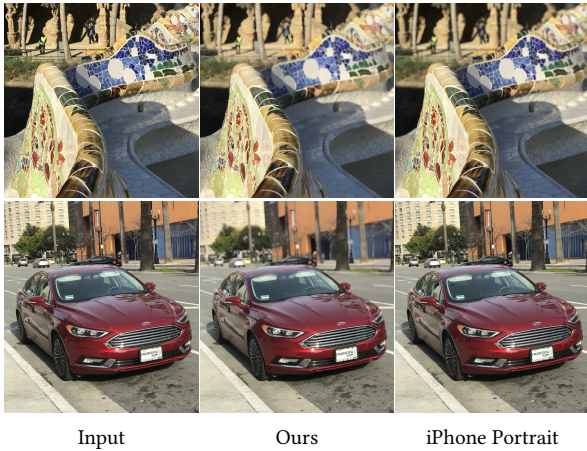| Input | Ours | iPhone Portrait |

Fig. 13. Qualitative comparison of our method and iPhone portrait mode (Cropped). In the first row, our method achieves comparable results against iPhone. In the second row, iPhone fails to render shallow DoF effects.

Fig. 14. Refocusing allows us to change both the synthetic aperture size (top) and the focal plane (bottom). *Photo credits: DAVID BURILLO and Greg Tsai.*

Fig. 15. Failure cases. The leg of the bird, and part of the grass are incorrectly blurred and focused. *Photo credits: Eric Sonstroem and C Watts.*

The second baseline uses predicted depth to achieve more robustness, but it requires training an additional SynDepth model, because the RealDepth model completely fails on the synthetic depth data. Furthermore, we evaluate a third baseline by replacing the depth prediction module of the second baseline with the SynDepth model. The comparison between the second and the third baselines suggests that the SynDepth model trained on synthetic depth fails to generalize to real scenes, leading to low quality of the final DoF results. As a matter of fact, the MAE of the SynDepth on the iPhone test set is 0.30, which is significantly worse than other competitors listed in Tab. 1. This further confirms the necessity of training depth prediction and lens blur modules on separate data sets.

5.4 User Study

To further evaluate the perceptual quality of our results, We conducted two user studies. In the first user study, we compare our results to the photos captured with iPhone portrait mode. Since we cannot control the focal depth in iPhone photos, and we cannot create defocus foreground images, we generate our results by manually matching focus parameters. In the second user study, we compare ours against MegaDepth[Li and Snavely 2018] + [Yang et al.

2016] (*i.e.* the state-of-the-art depth prediction method + RGB-D lens blur rendering) on images downloaded from the internet, which are captured by different cameras with various objects and scenes.

These two studies consisted of 86 and 58 participants respectively, During the study, each participant was shown 20 image sets, each consisting of the input all-in-focus image and the two shallow DoF images. All images and results in the study are included in the supplementary materials. In each image set, participants were asked which result they preferred, or if there was no preference. Fig. 11 shows the results of this experiment, we can see that when compared to the iPhone portrait mode, our method performs comparably, despite the fact that our method uses *predicted* depth while iPhone uses measured depth maps. In the second user study, our approach significantly outperforms the MegaDepth [2018] + [Yang et al. 2016]. We note that the iPhone portrait mode only works in a very limited depth range, which has been heavily optimized for (Fig. 13 shows two examples), whereas in the second user study, the scenes are more diverse, and both competitors only rely on a single image as input. Fig. 14 shows results of changing the focus parameters on the same image.

5.5 Limitations and Future Work

Our method has several limitations. First, single image depth estimation is still a challenging problem, and errors in the depth computation are reflected in the rendered DoF results (Fig. 15). Therefore, one of the future works will be improving our depth prediction module by using larger training datasets and better model architectures.

Second, we note that most computation of our method has to be performed at a low resolution due to current GPU memory limitations. To address this problem, we proposed a recurrent guided upsampling module which allows us to generate arbitrarily high resolution outputs at test time. However, it may further improve the performance if we can optimize the whole system directly on high resolution DoF images.

Finally, we note that it leads to a decrease in quality if we jointly train all the modules on our current datasets. The reasons are twofold: i) Rendering shallow DoF images with existing methods based on the collected depth leads to unpleasant artifacts, so any network trained with the rendered results as ground truth simply learns to recreate the artifacts in these existing approaches; ii) The synthetic dataset is artifact-free. However, training on this dataset in an end-to-end manner results in poor generalization ability to real scenes. We believe that a large representative dataset with high quality real-world shallow DoF images could be used to jointly train all the modules in our system for improved results, if such a dataset could be captured on a similar variety of dynamic scenes.

6 CONCLUSION

We have presented a neural network architecture, which combined with carefully crafted datasets consisting of both real and synthetic images, is capable of generating high quality shallow Depth-of-field results from a single RGB image. With the introduction of the lens blur module in low-res feature space and the guided upsampling module that can be recurrently applied to increase resolution, our model can generate high-resolution shallow DoF images with controllable focal depth and aperture size at interactive rates.

ACKNOWLEDGMENTS

REFERENCES

Jonathan T Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. 2015. Fast bilateral-space stereo for synthetic defocus. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 4466–4474.

Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. 2015. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 3 (2015), 569–582.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*. 2366–2374.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. 2016. High-quality depth from uncalibrated small motion clip. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5413–5421.

Paul Haeberli and Kurt Akeley. 1990. The accumulation buffer: hardware support for high-quality rendering. *ACM SIGGRAPH computer graphics* 24, 4 (1990), 309–318.

Kaiming He, Jian Sun, and Xiaoou Tang. 2010. Guided image filtering. In *Proceedings of European Conference on Computer Vision*. Springer, 1–14.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5967–5976.

Neel Joshi and C Lawrence Zitnick. 2014. Micro-baseline stereo. *Technical Report MSR-TR-2014–73* (2014), 8.

Felix Klose, Oliver Wang, Jean-Charles Bazin, Marcus Magnor, and Alexander Sorkine-Hornung. 2015. Sampling based scene-space video processing. *ACM Transactions on Graphics* 34, 4 (2015), 67.

Martin Kraus and Magnus Strengert. 2007. Depth-of-Field Rendering by Pyramidal Image Processing. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 645–654.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 2016. Deeper depth prediction with fully convolutional residual networks. In *3D Vision, 2016 Fourth International Conference on*. IEEE, 239–248.

Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. 2010. Real-time lens blur effects and focus control. *ACM Transactions on Graphics* 29, 4 (2010), 65.

Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. 2009. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 453–464.

Zhengqi Li and Noah Snavely. 2018. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Fayao Liu, Chunhua Shen, and Guosheng Lin. 2015. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5162–5170.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *Proceedings of European Conference on Computer Vision*. 746–760.

Pratul P. Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T. Barron. 2018. Aperture Supervision for Monocular Depth Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. 2017. Learning to Synthesize a 4D RGBD Light Field from a Single Image. In *Proceedings of IEEE International Conference on Computer Vision*. 2262–2270.

Supasorn Suwajanakorn, Carlos Hernandez, and Steven M Seitz. 2015. Depth from focus with your mobile phone. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3497–3506.

Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 2017. Learning to Detect Salient Objects with Image-Level Supervision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3796–3805.

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. 2017. Deep image matting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Yang Yang, Haiting Lin, Zhan Yu, Sylvain Paris, and Jingyi Yu. 2016. Virtual DSLR: High Quality Dynamic Depth-of-Field Synthesis on Mobile Platforms. *Electronic Imaging* 2016, 18 (2016), 1–9.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2881–2890.