RefocusGAN: Scene Refocusing using a Single Image

Parikshit Sakurikar[1], Ishit Mehta[1], Vineeth N. Balasubramanian[2]
and P. J. Narayanan[1]

[1] Center for Visual Information Technology, Kohli Center on Intelligent Systems,
International Institute of Information Technology, Hyderabad, India
[2] Department of Computer Science and Engineering,
Indian Institute of Technology, Hyderabad, India

Abstract. Post-capture control of the focus position of an image is a useful photographic tool. Changing the focus of a single image involves the complex task of simultaneously estimating the radiance and the defocus radius of all scene points. We introduce RefocusGAN, a deblur-then-reblur approach to single image refocusing. We train conditional adversarial networks for deblurring and refocusing using wide-aperture images created from light-fields. By appropriately conditioning our networks with a focus measure, an in-focus image and a refocus control parameter δ, we are able to achieve generic free-form refocusing over a single image.

Keywords: epsilon focus photography, single image refocusing

1 Introduction

An image captured by a wide-aperture camera has a finite depth-of-field centered around a specific focus position. The location of the focus plane and the size of the depth-of-field depend on the camera settings at the time of capture. Points from different parts of the scene contribute to one or more pixels in the image and the size and shape of their contribution depends on their relative position to the focus plane. Post-capture control of the focus position is a very useful tool for amateur and professional photographers alike. Changing the focus position of a scene using a single image is however an ill-constrained problem as the in-focus intensity and the true point-spread-function for each scene point must be jointly estimated before re-blurring a pixel to the target focus position.

Multiple focused images of a scene, in the form of a focal stack, contain the information required to estimate in-focus intensity and the focus variation for each scene point. Focal stacks have been used in the past for tasks such as estimating a sharp in-focus image of the scene [1, 20], computing the depth-map of the scene [19, 33], and free-form scene refocusing [11, 39]. In this paper, we introduce RefocusGAN, a comprehensive image refocusing framework which takes

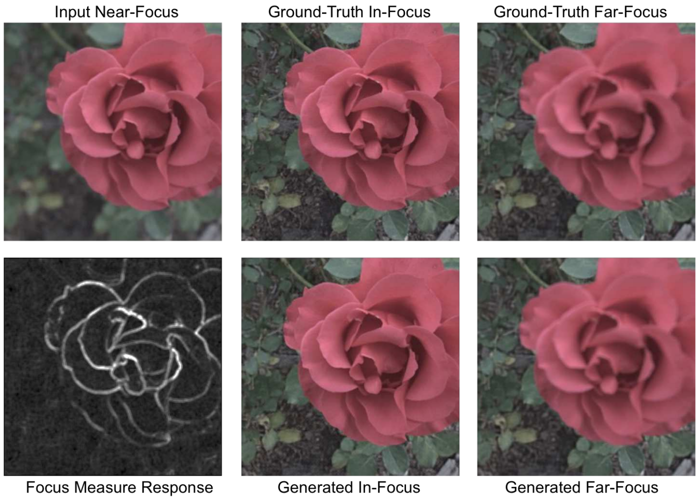

Fig. 1. Refocusing a single-image: We use an input wide-aperture image along with its focus measure response to create a deblurred, in-focus radiance image. The radiance image is then used together with the input image to create a refocused image. The second and third columns show the quality of our deblurring and refocusing stages.

only a single input image and enables post-capture control over its focus position. This is a departure from current methods in computational photography that provide post-capture control over depth-of-field using full focal stacks.

Our work is motivated by the impressive performance of deep neural networks for tasks such as image deblurring, image-to-image translation and depth-map computation from a single image. We propose a two-stage approach to single image refocusing. The first stage of our approach computes the radiance of the scene points by deblurring the input image. The second stage uses the wide-aperture image together with the computed radiance to produce a refocused image based on a refocus control parameter δ. We train conditional adversarial networks for both stages using a combination of adversarial and content loss [15]. Our networks are additionally conditioned by a focus measure response during deblurring and the computed radiance image during refocusing. We train our networks using wide-aperture images created from a large light-field dataset of scenes consisting of flowers and plants [29]. The main contribution of this paper is our novel two-stage algorithm for high-quality scene refocusing over a single input image. To the best of our knowledge, this is the first attempt at comprehensive focus manipulation of a single image using deep neural networks.

2 Related Work

Controlling the focus position of the scene is possible if multiple focused images of the scene are available, usually in the form of a focal stack. Jacobs et al.

[11] propose a geometric approach to refocusing and create refocused images by appropriately blending pixels from different focal slices, while correctly handling halo artifacts. Hach et al. [8] model real point-spread-functions between several pairs of focus positions, using a high quality RGBD camera and dense kernel calibration. They are thereby able to generate production-quality refocusing with accurate bokeh effects. Suwajanakorn et al. [33] compute the depth-map of the scene from a focal stack and then demonstrate scene refocusing using the computed depth values for each pixel. Several methods have been proposed in the past to compute in-focus images and depth maps from focal stacks [4, 19, 20, 26, 33]. Most of these methods enable post-capture control of focus but use all the images in the focal stack. Zhang and Cham [38] change the focus position of a single image by estimating the amount of focus at each pixel and use a blind deconvolution framework for refocusing. Methods based on Bae and Durand [3] also estimate the per-pixel focus map but for the task of defocus magnification. These methods are usually limited by the quality of the focus estimation algorithm as the task becomes much more challenging with increasing amounts of blur.

Deep neural networks have been used in the past for refocusing light-field images. Wang et al. [35] upsample the temporal resolution of a light-field video using another aligned 30 fps 2D video. The light-field at intermediate frames is interpolated using both the adjacent light-field frames as well as the 2D video frames using deep convolutional neural networks. Any frame can then be refocused freely as the light-field image at each temporal position is available. Full light-fields can themselves be generated using deep convolutional neural networks using only the four corner images as shown in [13]. A full 4D RGBD light-field can also be generated from a single image using deep neural networks trained over specific scene types as shown in [29]. Srinivasan et al. [28] implicitly estimate the depth-map of a scene by training a neural network to generate a wide-aperture image from an in-focus radiance image. These methods suggest that it is possible to generate light-fields using temporal and spatial interpolation. However, these methods have not been applied for focus interpolation.

Deep neural networks have been used for deblurring an input image to generate an in-focus image. Schuler et al. [27] describe a layered deep neural network architecture to estimate the blur kernel for blind image deblurring. Nimisha et al. [25] propose an end-to-end solution to blind deblurring using an autoencoder and adversarial training. Xu et al. [37] propose a convolutional neural network for deblurring based on separable kernels. Nah et al. [21] propose a multi-scale convolutional neural network with multi-scale loss for generating high-quality deblurring of dynamic scenes. Orest et al. [15] show state-of-the-art deblurring for dynamic scenes using a conditional adversarial network and use perceptual loss as an additional cue to train the deblurring network.

In this paper, we introduce RefocusGAN, a new approach to change the focus position of a single image using deep neural networks. Our approach first deblurs an input wide-aperture image to an in-focus image and then uses this in-

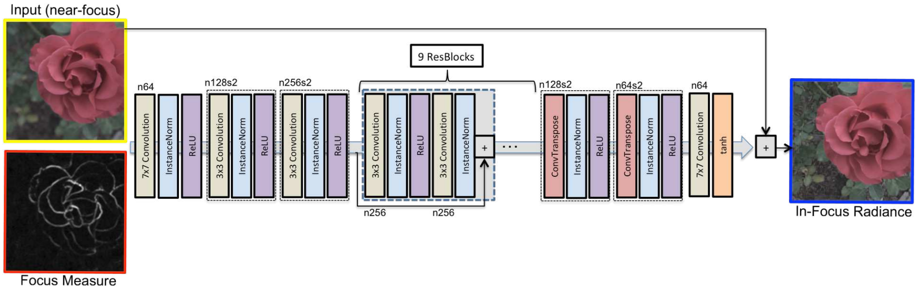

Fig. 2. The architecture of the deblurring cGAN. It receives a wide-aperture image and its focus measure channel as input and computes an in-focus radiance image.

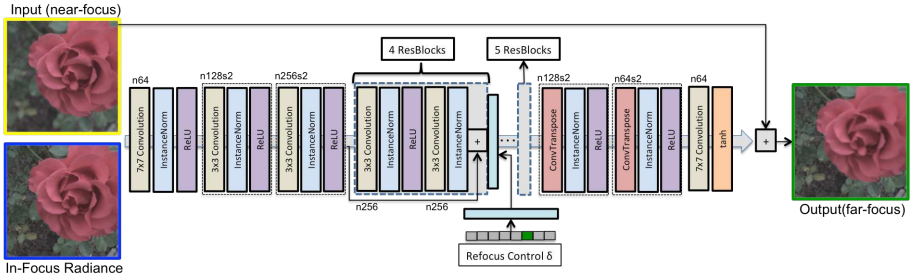

Fig. 3. The architecture of the refocusing cGAN. It uses the generated in-focus image together with the original wide-aperture image and a refocus control parameter δ to compute a refocused image.

focus image in conjunction with the wide-aperture image to simulate geometric refocusing.

3 Single Image Scene Refocusing

A standard approach to scene refocusing uses several wide-aperture images from a focal stack to generate a new image with the target depth-of-field. Refocusing is typically modeled as a composition of pixels from several focal slices to create a new pixel intensity. This reduces the task of refocusing to selecting a set of weights for each pixel across focal slices as is described in [11]. Other methods that use all the slices of a focal stack first estimate the depth map of the scene and a corresponding radiance image, and then convolve the radiance image with geometrically accurate blur kernels, such as in [8]. In the case of single images, it is difficult to simultaneously estimate the true radiance as well as the defocus radius at each pixel. Moreover, the complexity of the size and shape of the defocus kernel at each pixel depends on the scene geometry as well as the quality of the lens. A deep learning approach to refocus a wide-aperture image using a single end-to-end network does not perform very well and this is discussed in more detail in Section 5.

Refocusing a wide-aperture image can be modeled as a cascaded operation involving two steps in the image space. The first step is a deblurring operation that computes the true scene radiance $\hat{G}^r$ from a given wide-aperture image G^i, where i denotes the focus position during capture. This involves deblurring each pixel in a spatially varying manner in order to produce locally sharp pixels. The second step applies a new spatially varying blur to all the sharp pixels to generate the image corresponding to the new focus position $G^{i+\delta}$, where δ denotes the change in focus position. The required scene-depth information for geometric refocusing can be assumed to be implicit within this two-stage approach. Srinivasan et al. [28] have shown how the forward process of blurring can actually be used to compute an accurate depth-map of the scene. Our two-stage approach to refocusing a wide-aperture image is briefly described below.

In the first stage, an in-focus radiance image is computed from a given wide-aperture image G^i and an additional focus measure m evaluated over G^i. The focus measure provides a useful cue that improves the quality of deblurring:

$$\hat{G}^r = \mathcal{G}^1_{\theta_G}\left(G^i : m(G^i)\right) \tag{1}$$

In the second stage, the generated in-focus image is used together with the input wide-aperture image to generate the target image corresponding to a shifted focus position $i + \delta$.

$$G^{i+\delta} = \mathcal{G}^2_{\theta_G}\left(G^i : \hat{G}^r, \delta\right) \tag{2}$$

We train end-to-end conditional adversarial networks for both these stages. While the deblurring network $\mathcal{G}^1_\theta$ is motivated by existing blind image-deblurring works in the literature, we provide motivation for our second network $\mathcal{G}^2_\theta$ by producing a far-focused slice from a near-focused slice using a simple optimization method.

Adversarial Learning: Generative adversarial networks (GANs) [6] define the task of learning as a competition between two networks, a generator and a discriminator. The task of the generator is to create an image based on an arbitrary input, typically provided as a noise vector, and the task of the discriminator is to distinguish between a real image and this generated image. The generator is trained to created images that are perceptually similar to real images, such that the discriminator is unable to distinguish between real and generated samples. The objective function of adversarial learning can be defined as:

$$\underset{\mathcal{G}}{min}\,\underset{\mathcal{D}}{max}\,\mathcal{L}_{GAN}, \tag{3}$$

where $\mathcal{L}_{GAN}$ is the classic GAN loss function:

$$\mathcal{L}_{GAN} = E_{y\sim p_r(y)}\left[\log \mathcal{D}(y)\right] \,+\, E_{z\sim p_z(z)}\left[\log(1 - \mathcal{D}(\mathcal{G}(z)))\right], \tag{4}$$

where $\mathcal{D}$ represents the discriminator, $\mathcal{G}$ is the generator, y is a real sample, z is a noise vector input to the generator, p_r represents the real distribution over target samples and p_z is typically a normal distribution.

Conditional adversarial networks (cGANs), provide additional conditioning to the generator to create images in accordance with the conditioning parameters. Isola et al. [10] provide a comprehensive analysis of GANs for the task of image-to-image translation, and propose a robust cGAN architecture called pix2pix, where the generator learns a mapping from an image x and a noise vector z to an output image y as: $\mathcal{G} : x, z \to y$. The observed image is provided as conditioning to both the generator and the discriminator. We use cGANs for the tasks of de-blurring and refocusing and provide additional conditioning parameters to both our networks as defined in the following sections.

3.1 Deblurring a Wide-Aperture Image

We use a conditional adversarial network to deblur a wide aperture image G^i and estimate its corresponding scene radiance $\hat{G}^r$ as described in Equation 1. Our work draws inspiration from several deep learning methods for blind image-deblurring such as $[15, 21, 27, 37]$. Our network is similar to the state-of-the-art deblurring network proposed by Orest et al. [15]. Our generator network is built on the style transfer network of Johnson et al. [12] and consists of two strided convolution blocks with a stride of $\frac{1}{2}$, nine residual blocks and two transposed convolution blocks. Each residual block is based on the ResBlock architecture [9] and consists of a convolution layer with dropout regularization [30], instance-normalization [34] and ReLU activation [22]. The network learns a residual image since a global skip connection (ResOut) is added in order to accelerate learning and improve generalization [15]. The residual image is added to the input image to create the deblurred radiance image. The discriminator is a Wasserstein-GAN [2] with gradient penalty [7] as defined in [15]. The architecture of the critic discriminator network is identical to that of PatchGAN $[10, 16]$. All convolution layers except for the last layer are followed by instance normalization and Leaky ReLU [36] with an α=0.2.

The cGAN described in [15] is trained to sharpen an image blurred by a motion-blur kernel of the form $I_B = K * I_S + \eta$, where I_B is the blurred image, I_S is the sharp image, K is the motion blur kernel and η represents additive noise. In our case, the radiance image G^r has been blurred by a spatially varying defocus kernel and therefore the task of deblurring is more complex. We thereby append the input image G^i with an additional channel that encodes a focus measure response computed over the input image. We compute $m(G^i)$ as the response of the Sum-of-modified-Laplacian (SML) [23] filter applied over the input image. We also provide the input image along with this additional channel as conditioning to the discriminator. The adversarial loss for our deblurring network can be defined as:

$$\mathcal{L}_{cGAN} = \sum_{n=1}^{N} -\mathcal{D}^1_{\theta_D}(\mathcal{G}^1_{\theta_G}(x^i), x^i), \tag{5}$$

where $x^i = G^i : m(G^i)$ is the input wide-aperture image G^i concatenated with the focus measure channel $m(G^i)$.

In addition to the adversarial loss, we also use perceptual loss [12] as suggested in [15]. Perceptual loss is L2-loss between the CNN feature maps of the generated deblurred image and the target image:

$$\mathcal{L}_X = \frac{1}{W_{ij}H_{ij}} \sum_x \sum_y (\phi_{ij}(I^S)_{xy} - \phi_{ij}(\mathcal{G}_{\theta_G}(I^B))_{xy})^2, \tag{6}$$

where ϕ_{ij} is the feature map in VGG19 trained on ImageNet [5] after the j^{th} convolution and the i^{th} max-pooling layer and W and H denote the size of the feature maps. In this case, I^S and I^B represent the ground truth in-focus image and the input wide-aperture image respectively. The loss function for the generator is a weighted combination of adversarial and perceptual loss $\mathcal{L} = \mathcal{L}_{cGAN} + \lambda\mathcal{L}_X$.

The structure of our deblurring cGAN is shown in Figure 2. A few wide-aperture images along with the computed in-focus radiance image are shown in Figure 7.

3.2 Refocusing a Wide-Aperture Image

The in-focus image computed from the above network not only represents the true scene radiance at each pixel, but can also serve as proxy depth information in conjunction with the input wide-aperture image. We motivate our second refocusing network $\mathcal{G}^2_{\theta_G}$ using a simple method that can refocus a near-focus image to a far-focus image and vice versa, using the computed radiance image.

As shown in the example in Figure 4, a near-focused image G^1 can be converted to a far focused image G^n using the radiance image $\hat{G}^r$ resulting from the deblurring network. Here 1 and n are used to denote the near and far ends of the focus spread of a focal stack. To refocus these images, the first step would be to compute the per-pixel blur radius between the input image G^1 and the radiance image $\hat{G}^r$. This can be achieved using a blur-and-compare framework wherein the in-focus pixels of the radiance image are uniformly blurred by different radii and the best defocus radius σ is estimated for each pixel using pixel-difference between a blurred patch and the corresponding patch in G^1. Inverting these defocus radii as $\sigma' = \sigma_{max} - \sigma$ followed by re-blurring the radiance image is the natural way to create the refocused image. This method can also be used to convert a far-focused image to a near focused image as shown in the second row of Figure 4. Free-form refocusing between arbitrary focus positions is not trivial though since there is no front-to-back ordering information in the estimated defocus radii.

For free-form scene refocusing, we use a conditional adversarial network similar to our deblurring network. We use the same cGAN architecture of the previous section, with different conditioning and an additional refocus control parameter δ. The refocus control parameter is used to guide the network to produce a target image corresponding to a desired focus position. The input to the network is the original wide-aperture image G^i concatenated with the scene radiance image $\hat{G}^r = \mathcal{G}^1_{\theta_G}(G^i : m(G^i))$ computed by the deblurring network. The refocus

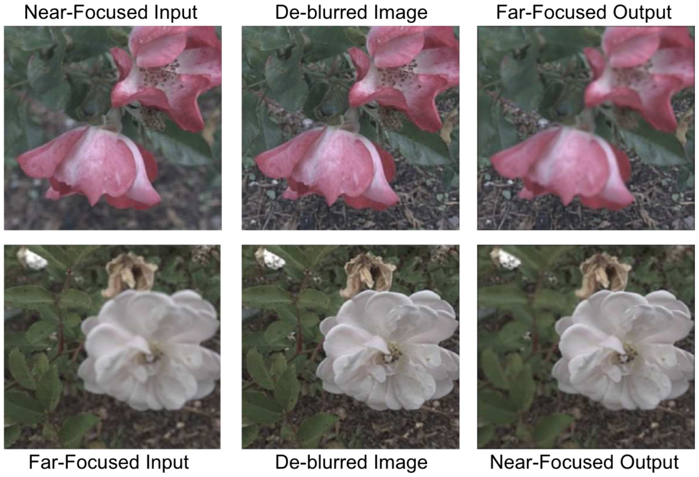

Fig. 4. Refocusing using a simple image-processing operation over the input wide-aperture image G^1 and the deblurred in-focus image $\hat{G}^r$. The first row shows the input near-focused image, the deblurred in-focus image from the network and the computed far-focused image. The second row shows equivalent far-to-near refocusing.

parameter δ encodes the shift between the input and output images and is provided to the network as a one-hot vector. The refocus vector corresponding to δ is concatenated as an additional channel to the innermost layer of the network, using a fully connected layer to convert the one-hot vector into a 64×64 channel.

The structure of the refocusing cGAN is shown in Figure 3. We use the same structure for the discriminator and the generator as that of the deblurring cGAN. The loss function for the generator is a summation of adversarial loss and perceptual loss. The discriminator network is conditioned using the input image and the in-focus radiance image. The cGAN loss for this network can be defined as:

$$\mathcal{L}_{cGAN} = \sum_{n=1}^{N} -\mathcal{D}^2_{\theta_D}(\mathcal{G}^2_{\theta_G}(x^i), x^i), \tag{7}$$

where $x^i = G^i : \hat{G}^r$ is the input wide-aperture image G^i concatenated with the scene radiance image $\hat{G}^r = \mathcal{G}^1_{\theta_G}(G^i : m(G^i))$. Refocused images generated from the input wide-aperture image, the in-focus image and different refocus parameters are shown in Figures 8,9.

4 Training Details

For training both networks, we compute multiple wide-aperture images from a large light-field dataset of scenes consisting of flowers and plants [29]. The method used to generate training images from light-fields is explained in the following section.

4.1 Focal Stacks from Light-Fields

A focal stack is a sequence of differently focused images of the scene with a fixed focus step between consequent images of the stack. A focal stack can be understood as an ordered collection of differently blurred versions of the scene radiance. A focal slice G^i is a wide-aperture image corresponding to a focus position i and can be defined as:

$$G^i = \int\int h^i(x, y, d_{x,y}) * \hat{G}^r(x, y)\, dx\, dy\,, \tag{8}$$

where h^i is the spatially varying blur kernel dependent on the spatial location of the pixel and the depth $d_{x,y}$ of its corresponding scene point and $\hat{G}^r$ is the true radiance of the scene point which is usually represented by the in-focus intensity of the pixel.

An ideal focal stack, as defined by Zhou et al. [39], consists of each pixel in focus in one and only one slice. Focal stacks can be captured by manually or programmatically varying the focus position between consequent shots. Programmed control of the focus position is possible nowadays on DSLR cameras as well as high-end mobile devices. Canon DSLR cameras can be programmed using the MagicLantern API [18] and both iOS and Android mobile devices can be controlled using the Swift Camera SDK and Camera2 API respectively. Capturing a focal stack as multiple shots suffers from the limitation that the scene must be static across the duration of capture, which is difficult to enforce for most natural scenes. Handheld capture of focal stacks is also difficult due to the multiple shots involved. Moreover, being able to easily capture focal stacks is a somewhat recent development and there is a dearth of large quantities of focal stack image sequences.

A focal stack can also be created from a light-field image of the scene. The Lytro light-field camera based on Ng et al. [24] captures a 4D light-field of a scene in a single shot, and can thereby be used for dynamic scenes. The different angular views captured by a light-field camera can be merged together to create wide-aperture views corresponding to different focus positions. A large number of light-fields of structurally similar scenes have been captured by Srinivasan et al. [29]. Several other light-field datasets also exist such as the Stanford light-field archive [31] and the light-field saliency dataset [17]. Large quantities of similar focal stacks can be created from such light-field datasets.

Srinivasan et al. [29] captured a large light-field dataset of 3343 light-fields of scenes consisting of flowers and plants using the Lytro Illum Camera. Each image in the dataset consists of the angular views encoded into a single light-field image. A 14×14 grid of angular views can be extrapolated from the light-field, each having a spatial resolution of 376×541. Typically, only the central 8×8 views are useful as the samples towards the corners of the light-field suffer from clipping as they lie outside the camera's aperture. This dataset is described in detail in [29]. A few sample images from this dataset are shown in Figure 5. For our experiments, we use a central 7×7 grid of views to create focal stacks, so as to have a unique geometric center to represent the in-focus image. We generate a focal stack at a high focus resolution for each of these light-field images using the synthetic photography equation defined in [24]:

$$G^i(s,t) = \int\int L\left(u, v, u + \frac{s-u}{\alpha_i}, v + \frac{t-v}{\alpha_i}\right) du\, dv. \tag{9}$$

Here G^i represents the synthesized focal slice, $L(u, v, s, t)$ is the 4D light-field represented using the standard two-plane parameterization and α_i represents the location of the focus plane. This parameterization is equivalent to the summation of shifted versions of the angular views captured by the lenslets as shown in [24]. We vary the shift-sum parameter linearly between -s_{max} to $+s_{max}$ to generate 30 focal slices between the near and far end of focus.

Fig. 5. A few examples of the light-field images in the Flowers dataset of [29].

To reduce the size of focal stacks to an optimal number of slices, we apply the composite focus measure [26] and study the focus variation of pixels across the stack. We use this measure as it has been shown to be more robust than any single focus measure for the task of depth-from-focus [26]. For each pixel, we record the normalized response of the composite measure at each slice. We build a histogram of the number of pixels that peaked at each of the 30 slices across the 3343 light-field dataset. We find that in close to 90% of the images, all the pixels peak between slices 6 and 15 of the generated focal stack. The depth variation of the captured scenes is mostly covered by these ten focal slices. We thereby subsample each focal stack to consist of ten slices, varying from slice 6 to slice 15 of our original parameterization. Our training experiments use these 10-sliced focal stacks computed from the light-field dataset.

For training, the 3343 focal stacks are partitioned into 2500 training samples and 843 test samples. Each focal slice is cropped to a spatial resolution of 256×256 pixels. The s_{max} parameter while computing focal slices is set to 1.5 pixels. For the deblurring network, we use all the ten focal slices from the 2500 focal stacks for training. For the refocusing network, we experiment with three different configurations. In the first configuration a single refocus parameter of $\delta = +8$ is used. In the second configuration, the refocus parameter has four distinct values: $\delta = \{-9, -5, +5, +9\}$. In the third configuration, the refocus parameter can take any one of 19 possible values from -9 to $+9$. The deblurring network is trained for 30 epochs (∼50 hours) and all configurations of the refocusing network are trained for 60 epochs (∼45 hours). All training experiments were performed on an Nvidia GTX 1080Ti. The learning rate is set to 0.0001

Table 1. Quantitative evaluation of our deblurring network. PSNR and SSIM is reported for the test-split of the light-field dataset. We compare the performance of the deblurring network with and without the additional Sum-of-Modified-Laplacian (SML) focus measure channel. There is a marginal but useful improvement in the quality of deblurring on using the focus measure channel. As an indication of overall performance, we generate an in-focus image using the composite focus measure [26] applied on all slices of the focal stack and report its quality. Note that our method uses only a single image.

Deblurring Experiment	PSNR	SSIM
Ours (without additional Focus Measure)	34.88	0.937
Ours (with additional Focus Measure)	35.02	0.938
Composite Focus Measure (uses entire stack)	38.697	0.965

initially for all network configurations. The learning rate is linearly decreased to zero after half the total number of epochs are completed. All networks are trained for a batch size of 1 and the Adam solver [14] is used for gradient descent. The λ parameter for scaling content loss is set to 100 as suggested in [15].

5 Experiments and Results

We provide a quantitative evaluation of the performance of our two-stage refocusing approach in Tables 1,2. We compare the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) of the refocused images with the ground truth images from the focal stacks. Since this is the first work that comprehensively manipulates the focus position from a single image, there is no direct comparison of the generated refocused images with existing geometric techniques over focal stacks. However, we generate in-focus images using the composite focus measure [26] applied across the full focal stack and report the quantitative reconstruction quality in Table 1. We show the quantitative performance of our networks individually and report the PSNR and SSIM of the computed in-focus radiance image in comparison with the ground-truth central light-field image.

Our two-stage approach to refocusing is motivated by our initial experiments wherein we observed that an end-to-end refocusing network does not work well. Our experiments spanned several network architectures such as the purely convolutional architecture of the disparity estimation network of [13], the separable kernel convolutional architecture of [37], the encoder-decoder style deep network with skip-connections of [32] and the conditional adversarial network of [15]. These networks exhibit poor refocusing performance in both cases of fixed pairs of input-output focal slices as well as for the more complex task of free-form refocusing. Since the networks are only given input wide-aperture images while training, there may be several pixel intensities which do not occur sharply in either the input or output images, and the task of jointly estimating all true intensities and re-blurring them is difficult to achieve within a reasonable compute power/time budget for training. In Table 2, we compare our two-stage approach to refocusing with an equivalent single-stage, end-to-end network. This essentially compares the performance of our refocusing network with and without the additional radiance image computed by the deblurring network. It can be seen that the two-stage method clearly outperforms a single-stage approach to refocusing.

Table 2. Quantitative evaluation of our refocusing network. The PSNR and SSIM values are reported on the test-split of the light-field dataset. The first two rows show the performance of our refocusing network without an additional in-focus image. This corresponds to an end-to-end, single stage approach to refocusing. The next three rows show the performance on using different refocus control parameters in our two-stage experiments. The final row shows the test performance of our refocusing network which was trained using ground truth in-focus images G^r but tested using the radiance images computed by the deblurring network $\hat{G}^r$. Note that the two-stage approaches significantly outperform their single-stage counterparts. The high PSNR and SSIM values quantitatively suggest that our network enables high-quality refocusing.

Experiment	Type	Refocus Control Steps	PSNR	SSIM
Without G^r	single-stage	+8	38.73	0.97
Without G^r	single-stage	{-9,-5,+5,+9}	38.4	0.956
With G^r	two-stage	+8	44.225	0.992
With G^r	two-stage	{-9,-5,+5,+9}	43.4	0.988
With G^r	two-stage	{-9,-8,..,0,..,+8,+9}	40.42	0.975
With AIF($\hat{G}^r$)	two-step	-9,-5,+5,+9	38.63	0.958

The deblurring network uses an additional focus measure channel to compute the radiance image $\hat{G}^r$. The benefit of using the focus measure is indicated in Table 1. For the refocusing network, we perform experiments on three different configurations. The configurations differ from each other in the number of refocus control parameters and are shown in Table 2. The first configuration is a proof-of-concept network and is trained on a single refocus parameter. This clearly exhibits the best performance as the training samples have a high degree of structural similarity. The network with four control parameters performs better than the network with 19 parameters, which can be seen in Table 2. This can be attributed to two separate issues. The focal stacks created from the light-field dataset consist of ten slices that roughly span the depth range of the scene from near-to-far. However, in the absence of scene content at all depths, certain focal slices may be structurally very similar to adjacent slices. Training these slices with different control parameters can confuse the network. Secondly, in the case of the 19 parameter configuration, the total number of training samples increases to 250000 as there are 100 samples from each focal stack. We use a subset of size 30000 from these training images sampled uniformly at random. In the case of refocusing with 4 control parameters, the focus shift between input and output images is clearly defined and the network thereby captures the relationship better. All the training samples from the dataset can be used directly to train this network as there are only 12 training samples per focal stack in the four parameter configuration.

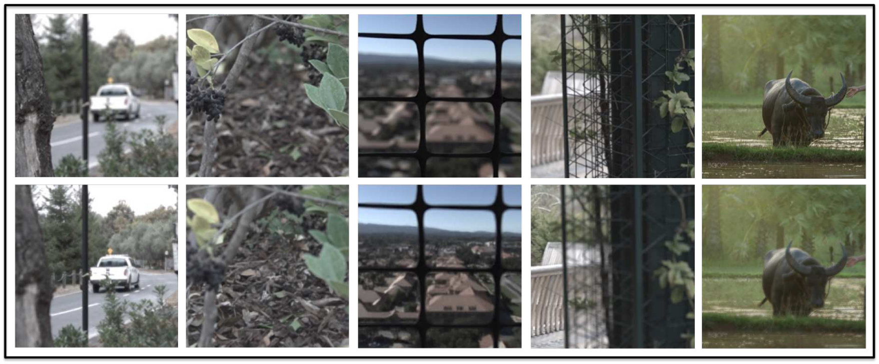

Fig. 6. The performance of our two-stage refocusing framework on generic images. The first row has the input wide-aperture image and the second row shows the refocused image. The first four columns show the performance on structurally different light-field focal slices from another light-field dataset while the last column shows the performance on an image captured by a wide-aperture camera.

We show qualitative deblurring and refocusing results for several test samples in Figures 7,8,9. In Figure 6, we show the performance of our refocusing framework on generic images from different light-fields that were not images of flowers or plants, and also show the performance on an image captured using a wide-aperture camera. The performance suggests that our networks are implicitly learning both tasks quite well and can be used for high-quality refocusing of standalone images.

6 Conclusion

We present a two-stage approach for comprehensive scene refocusing over a single-image. Our RefocusGAN framework uses adversarial training and perceptual loss to train separate deblurring and refocusing networks. We provide a focus measure channel as an additional conditioning for deblurring a wide-aperture image. We use the deblurred in-focus image as an additional conditioning for refocusing. Our quantitative and qualitative results suggest high-quality performance on refocusing. Our networks exhibit useful generalization and can further benefit from fine-tuning and training over multiple datasets together. In the future, we plan to work on creating a refocusing network based on a free-form refocus parameter that is independent of the number and spread of focal slices.

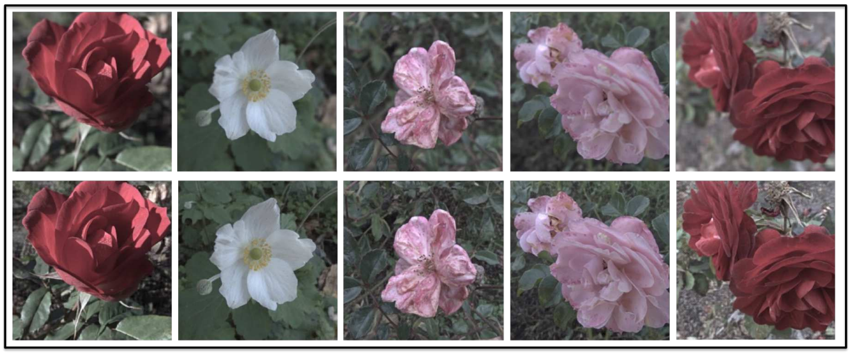

Fig. 7. In-focus radiance images created using the deblurring network. The top row shows the input wide-aperture images and the bottom row shows the deblurred output from our deblurring network.

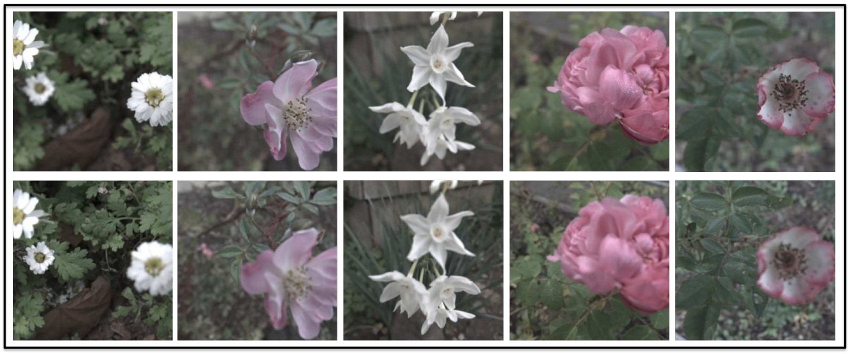

Fig. 8. Near-to-Far Refocusing generated with δ=+9 using our refocusing network. The top row shows the input wide-aperture images and the bottom row shows the output refocused images.

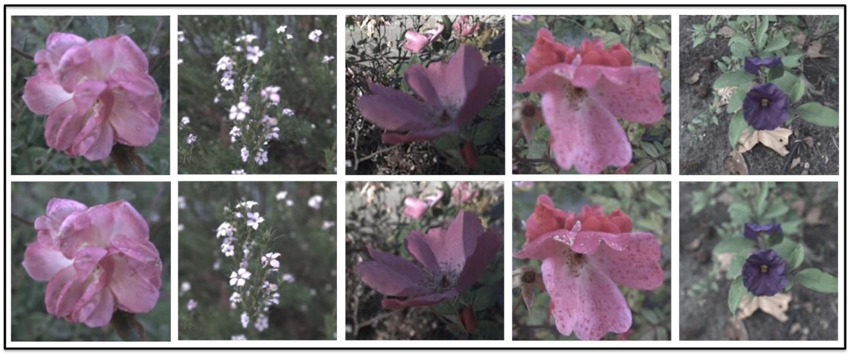

Fig. 9. Far-to-Near Refocusing generated with δ=−9 using our refocusing network. The top row shows the input wide-aperture images and the bottom row shows the output refocused images.

References

1. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Interactive digital photomontage. In: ACM Transactions on Graphics. vol. 23, pp. 294–302. ACM (2004)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 214–223 (2017)
3. Bae, S., Durand, F.: Defocus magnification. In: Computer Graphics Forum. vol. 26, pp. 571–579 (2007)
4. Bailey, S.W., Echevarria, J.I., Bodenheimer, B., Gutierrez, D.: Fast depth from defocus from focal stacks. The Visual Computer **31**(12), 1697–1708 (2015)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems (NIPS). pp. 2672–2680 (2014)
7. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems (NIPS). pp. 5767–5777 (2017)
8. Hach, T., Steurer, J., Amruth, A., Pappenheim, A.: Cinematic bokeh rendering for real scenes. In: Proceedings of the 12th European Conference on Visual Media Production. pp. 1:1–1:10. CVMP '15 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5967–5976 (2017)
11. Jacobs, D.E., Baek, J., Levoy, M.: Focal stack compositing for depth of field control. Stanford Computer Graphics Laboratory Technical Report **1** (2012)
12. Johnson, J., Alahi, A., Fei-Fei, L., Li, C., Li, Y.W., fei Li, F.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV) (2016)
13. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. ACM Transactions on Graphics **35**(6), 193:1–193:10 (Nov 2016)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
15. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8183–8192 (2018)
16. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: European Conference on Computer Vision (ECCV). pp. 702–716 (2016)
17. Li, N., Ye, J., Ji, Y., Ling, H., Yu, J.: Saliency detection on light field. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2014)
18. Magic lantern. http://magiclantern.fm/
19. Möller, M., Benning, M., Schönlieb, C.B., Cremers, D.: Variational depth from focus reconstruction. IEEE Transactions on Image Processing **24**, 5369–5378 (2015)

20. Nagahara, H., Kuthirummal, S., Zhou, C., Nayar, S.K.: Flexible depth of field photography. In: European Conference on Computer Vision (ECCV) (2008)
21. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 257–265 (2017)
22. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning. pp. 807–814. ICML (2010)
23. Nayar, S.K., Nakagawa, Y.: Shape from focus. Trans. on Pattern Analysis and Machine Intelligence (PAMI) **16**(8), 824–831 (1994)
24. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P., et al.: Light field photography with a hand-held plenoptic camera. Computer Science Technical Report CSTR **2**(11), 1–11 (2005)
25. Nimisha, T.M., Singh, A.K., Rajagopalan, A.N.: Blur-invariant deep learning for blind-deblurring. In: IEEE International Conference on Computer Vision (ICCV). pp. 4762–4770 (2017)
26. Sakurikar, P., Narayanan, P.J.: Composite focus measure for high quality depth maps. In: IEEE International Conference on Computer Vision (ICCV). pp. 1623–1631 (2017)
27. Schuler, C.J., Hirsch, M., Harmeling, S., Schlkopf, B.: Learning to deblur. Trans. on Pattern Analysis and Machine Intelligence (PAMI) **38**(7), 1439–1451 (2016)
28. Srinivasan, P.P., Garg, R., Wadhwa, N., Ng, R., Barron, J.T.: Aperture supervision for monocular depth estimation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
29. Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to synthesize a 4d RGBD light field from a single image. In: IEEE International Conference on Computer Vision, (ICCV). pp. 2262–2270 (2017)
30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
31. Stanford light-field archive. http://lightfield.stanford.edu/
32. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 237–246 (2017)
33. Suwajanakorn, S., Hernandez, C., Seitz, S.M.: Depth from focus with your mobile phone. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
34. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. CoRR **abs/1607.08022** (2016)
35. Wang, T.C., Zhu, J.Y., Kalantari, N.K., Efros, A.A., Ramamoorthi, R.: Light field video capture using a learning-based hybrid imaging system. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017) **36**(4) (2017)
36. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. CoRR **abs/1505.00853** (2015)
37. Xu, L., Ren, J.S.J., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1. pp. 1790–1798. NIPS'14 (2014)
38. Zhang, W., Cham, W.K.: Single-image refocusing and defocusing. IEEE Transactions on Image Processing **21**(2), 873–882 (2012)
39. Zhou, C., Miau, D., Nayar, S.K.: Focal sweep camera for space-time refocusing. Technical Report, Department of Computer Science, Columbia University **CUCS-021-12** (2012)