M.S. THESIS

Zero-shot defocus deblurring using dual-pixel image

듀얼 픽셀 이미지 기반 제로샷 디포커스 디블러링

August 2022

Graduate School of Artificial Intelligence
Seoul National University

Jaehyoung Yoo

Zero-shot defocus deblurring using dual-pixel image

듀얼 픽셀 이미지 기반 제로샷 디포커스 디블러링

지도 교수 한보형

이 논문을 공학석사 학위논문으로 제출함
2022년 8월

서울대학교 대학원
협동과정 인공지능전공

유 재 형

유재형의 공학석사 학위 논문을 인준함
2022년 8월

위 원 장: 이경무 (인)
부위원장: 한보형 (인)
위 원: 김선주 (인)

Abstract

Defocus deblurring in dual-pixel (DP) images is a challenging problem due to diverse camera optics and scene structures. Most of the existing algorithms rely on supervised learning approaches trained on the Canon DSLR dataset but often suffer from weak generalizability to out-of-distribution images including the ones captured by smartphones. We propose a novel zero-shot defocus deblurring algorithm, which only requires a pair of DP images without any training data and a pre-calibrated ground-truth blur kernel. Specifically, our approach first initializes a sharp latent map using a parametric blur kernel with a symmetry constraint. It then uses a convolutional neural network (CNN) to estimate the defocus map that best describes the observed DP image. Finally, it employs a generative model to learn scene-specific non-uniform blur kernels to compute the final enhanced images. We demonstrate that the proposed unsupervised technique outperforms the counterparts based on supervised learning when training and testing run in different datasets. We also present that our model achieves competitive accuracy when tested on in-distribution data.

Keyword : Dual-pixel sensor, defocus deblurring, zero-shot learning, non-uniform blur kernels.
Student Number : 2020-24766

Contents

Abstract 1

Contents 2

List of Tables 4

List of Figures 5

Chapter 1. Introduction 6

1.1. Background 6

1.2. Overview 9

1.3. Contribution 11

Chapter 2. Related Works 12

2.1. Defocus Deblurring 12

2.2. Defocus Map 13

2.3. Multiplane Image Representation 14

2.4. DP Blur Kernel 14

Chapter 3. Proposed Methods 16

3.1. Latent Map Initialization 17

3.2. Defocus Map Estimation 20

3.3. Learning Blur Kernels 22

3.4. Implementation Details 25

Chapter 4. Experiments 28

4.1. Dataset 28

4.2. Quantitative Results 29

4.3. Qualitative Results 31

Chapter 5. Conclusions 37

5.1. Summary 37

5.2. Discussion 38

Bibliography 39

Abstract in Korean 43

List of Tables

3.1 Ablation study on regularization terms 26

4.1 Quantitative comparison on Smartphone dataset 29

4.2. Quantitative comparison on DSLR dataset 30

List of Figures

1.1 Dual-pixel image formulation 7

1.2 An overview of our approach 10

3.1 Sharp image restoration using symmetric blur kernel..... 16

3.2 Visualization of the initialized latent map 18

3.3 Visualization of learned non-uniform blur kernel..... 23

3.4 Detail network architecture 25

4.1 Visualization of the proposed method 32

4.2 Qualitative comparison on Smartphone dataset 34

4.3 Qualitative comparison on DSLR dataset 35

Chapter 1. Introduction

1.1. Background

Image deblurring is a classical ill-posed inverse problem, which has been studied for a long time [18, 28]. In general, the problem can be formulated as,

$$y = x * k + n, \tag{1}$$

where y is the observed blurry image, x is the latent sharp image, k is the blur kernel, and n is the noise. Non-blind deblurring [19, 26] methods take a two-step approach: Estimate the blur kernel first and then restore the image. In other words, non-blind image deblurring restores the sharp latent x from its observation y given blur kernel k. Restoring sharp latent is mostly based on the maximum-a-posterior framework, and various image priors [5, 9] for improving the image quality have been proposed. However, these optimization-based approaches are difficult to work around and do not successfully restore images.

In this paper, we tackle non-blind defocus deblurring using a dual-pixel (DP) sensor. Defocus deblurring is a subtask of image deblurring that occurs when an image is captured with a shallow depth of field (DOF). We use a DP sensor to solve defocus deblurring by using defocus cues on both left and right images. The DP sensor has two photodiodes with separate lenses per pixel capturing two

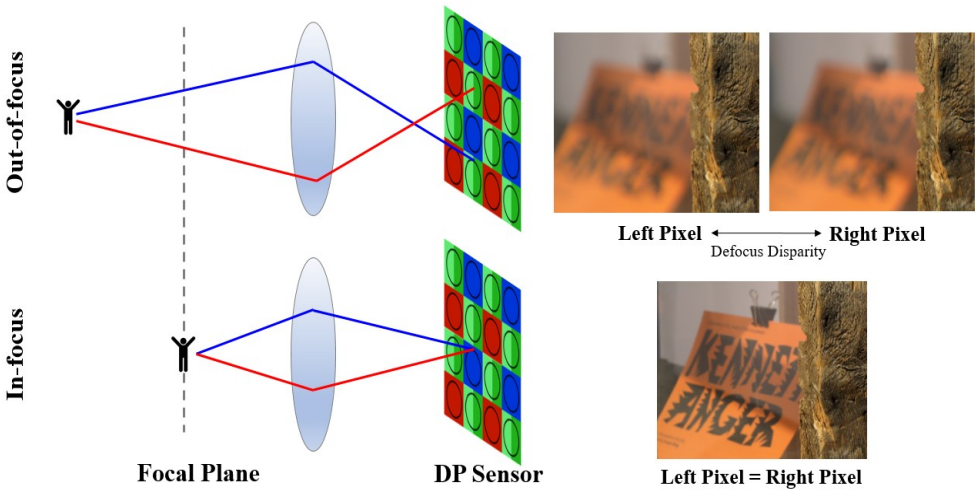

Fig 1.1: A dual-pixel (DP) image formulation. The out-of-focus objects exhibit different amount of defocus disparity between left and right image. On the other hand, in-focus objects do not cause defocus disparity between two image.

sub-aperture left-right images at once. Canon first proposed the DP design to improve autofocus in their mirrorless camera. DP is useful for autofocus as an in-focus scene induces equal intensity between the two views, while an out-of-focus area causes defocus blur causing a disparity between the left and right DP sub-aperture views (Fig. 1.1). DP sensors have now become the mainstream of smartphone cameras, and their applications include: depth estimation [7, 15, 31], synthetic bokeh [1, 27], and defocus deblurring [2-4]. However, the number of datasets is limited because camera manufacturers do not provide access to the raw DP data.

Defocus deblurring is difficult to solve due to non-uniform blurring (i.e. spatially varying). Unlike motion blur, where a blur

kernel that is uniform across the image may be sufficient to reflect camera shake during exposure, defocus blur is mostly affected by the distance from the camera focal plane to the target object. The size of the blur increases as the object moves away from the camera focal plane. Because the depth of the image is not uniform across the image, defocus deblurring requires a spatially varying non-uniform blur kernel. Thus, the defocus deblurring of a DP sensor can be modeled as,

$$y^L = x * f^L(d),$$
$$y^R = x * f^R(d), \tag{2}$$

where d is the defocus map which encodes the magnitude of the blur or the signed distance from the camera focal plane, and f is the camera-dependent non-uniform blur kernel function, as the characteristics of the camera optic can also affect the blur kernel. Therefore, defocus deblurring is equivalent to estimating a defocus map d, when we have prior information about f. Note that we do not consider the noise term from Eq. (1) in our work.

On the other hand, data-driven defocus deblurring [2-4, 10, 11, 15, 29] using the deep neural network has dominated conventional optimization-based methods. Abuolaim et al. [2] first introduced a large-scale DP dataset based on Canon DSLR and a U-Net-like network for defocus deblurring using DP images. Since then, various learning frameworks and neural network architecture benchmarked on the Canon dataset was introduced. Although such data-driven supervised learning achieves improved performance, it is still

questionable whether it generalizes well to images from other camera optics. Few works address the generalization problem of DP defocus deblurring, as the Canon dataset is currently the only one available for training. Xin et al. [29] showed that their unsupervised optimization framework can restore sharp edges better for smartphone cameras than the supervised methods trained using DSLR cameras. They carefully calibrate the blur kernel of the target smartphone camera and use it for optimization. Their work is impressive, but requires a pre-calibration step for each camera, which is tedious and difficult for non-professionals. Moreover, their method is not fully unsupervised since they use a ground-truth non-uniform blur kernel for optimization.

Recently, Ren et al. [17] proposed a neural blind deconvolution method that jointly estimates the motion blur kernel and sharp latent without relying on massive training data. Although the task is limited to uniform blur models, the proposed zero-shot learning framework using a CNN directly generates a sharp latent and a blur kernel that maximizes a posteriori given the observed image.

1.2. Overview

Inspired by [17, 29], we propose a zero-shot defocus deblurring method using only a DP image pair at test time. To solve this fully-unsupervised problem, we utilize the symmetric constraints [16] of the DP blur kernel in the initialization phase. Using a symmetrically

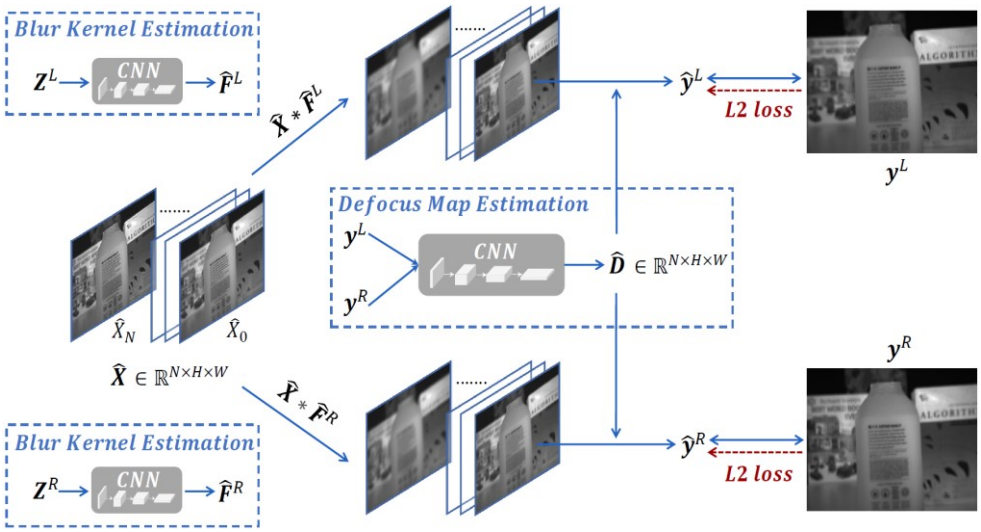

Fig 1.2: An overview of our approach. We propose a zero−shot defocus deblurring only using an observed blurry image y^L and y^R. The framework outputs $\hat{y}^L$ and $\hat{y}^R$ by blurrying the latent map $\hat{\boldsymbol{X}}$ using the blur kernel $\hat{\boldsymbol{F}}$ and soft−blend it by defocus map $\hat{\boldsymbol{D}}$. Then, we minimize the $L2$ loss between the observation (y) and estimation $(\hat{y})$ to train CNNs and the latent map $\hat{\boldsymbol{X}}$.

modeled blur kernels for left−right DP images, the proposed method blurs the initial latent map. Each layer of the latent map is then optimized to maximizes a posteriori probability of observed blurry DP image pairs. After the optimization, the in−focus areas for each layer of the latent map are restored due to the symmetric constraint from the left−right blur kernels. Then, the initialized latent map is soft−blended by the estimated defocus map, and outputs a defocus deblurred latent image. For estimating the defocus map, we use a convolutional neural network (CNN) that takes DP image pairs as an

input. The defocus map estimation network is trained to maximize a posteriori given the observed blurry DP images. For further improvement, we incorporate blur kernel estimation network to learn scene-specific and non-uniform blur kernels.

1.3. Contribution

Our contributions are summarized as below:

1. We propose a zero-shot defocus deblurring method using DP images. Our fully unsupervised framework first exploits the symmetric property of the DP blur kernel and then learns scene-specific non-uniform blur kernels.
2. We extend the prior work by introducing a more generalized framework that can handle both front and back defocus blur without a pre-calibration step for the target camera. By adopting the neural network with proper initialization, our method runs 20× faster than the previous work.
3. Our method generalizes better than the supervised method on unseen data and shows competitive performance to the supervised method on public benchmark dataset.

Chapter 2. Related Works

2.1. Defocus Deblurring

Single image defocus deblurring has been studied for a long time and various methods have been proposed ranging from classical approaches [8, 9] to more recent deep learning-based approaches [11, 34]. Karaali et al. [8] proposed a classical approach using edge-based gradient method for estimating spatially varying defocus blur using a single image. Moving into the era of deep learning, Lee et al. [11] proposed a deblurring filters based on neural network, composed of stacks of small-sized separable filters applied iteratively for effectively managing large defocus blur. Son et al. [34] proposed another deep learning approach exploiting the characteristics of inverse kernels. They utilize the property that the shape of inverse kernel remains the same and only the size changes as the amount of defocus blur changes.

Defocus deblurring using dual images from the DP sensor was recently introduced. Abuolaim et al. [2] provided the first high-resolution defocused DP image pairs with the corresponding all-in-focus ground-truths images using a Canon DSLR camera. Also, to predict an all-in-focus-image from the DP image pair, they train a Dual-Pixel Defocus deblurring Network (DPDNet) which adapts a U-Net-like architecture. Lee et al. [11] proposed an Iterative Filter

Adaptive Network (IFAN) which incorporates iterative adaptive convolution layers to handle large defocus blur in a spatially-varying manner. In addition, because DP data is limited and difficult to acquire, there is also a line of research for mathematically modeling and simulating the DP image formation pipeline to generate realistic DP data to address the problem of data scarcity [3, 15]. Leveraging these synthetic DP images, Abuolaim et al. [3] proposed a Recurrent Dual-Pixel Deblurring (RDPD) method using CNN-LSTM architecture that improves the deblurring results.

Recently, Xin et al. [29] introduced a non-blind defocus deblurring based on an optimization framework to recover an all-in-focus image using a pair of observed DP images and a carefully captured ground-truth blur kernel. Their work has been shown to restore high-frequency details better than the DPDNet in images captured by smartphone cameras, without using any training datasets. In contrast to these prior works, our method tackles defocus deblurring without using real blur kernels and large training datasets.

2.2. Defocus Map

Estimating the defocus map is important for estimating the defocus deblurring of an image. This is because the amount of defocus blur is heavily dependent on the defocus map which encodes the relative depth from the focal plane. Zhou et al. [35] proposed a simple approach to estimate the amount of defocus blur at edges by

re-blurring the defocused image using a Gaussian kernel and measuring the ratio of gradients between defocused and re-blurred images. Lee et al. [10] proposed the first end-to-end defocus map estimation network using domain adaptation with the new large-scale dataset for supervised learning. Punnappurath et al. [16] proposed to recover a depth estimation by using a point spread function to model the defocus disparity from the DP sensors, Using the symmetry property of the model, they proposed unsupervised method that does not require ground truth depth.

2.3. Multiplane Image Representation

Multiplane image (MPI) is a 3-dimensional layered representation of the scene to produce a view that are spatially consistent [22]. MPI is widely used in scene rendering [21, 24, 33] because its soft blending supports differentiable optimization properties along with the ability to represent occlusion. MPI can also be used for defocus deblurring [29] since the size of the blur kernel is positively related to the distance to the focal plane of the camera. Hence, each layer of MPI can represent the size of the blur kernel which increases as it reaches to the last layer.

2.4. DP Blur Kernel

Non-blind defocus deblurring used parametric blur models such

as disk [6] or Gaussian kernel [8, 14]. As non-blind deblurring is susceptible to errors in blur kernel, several works [12, 13] estimate camera-dependent blur kernels from calibration patterns. For modeling blur kernels in the DP sensor, Punnappurath et al. [16] used Canon DSLR and a pre-defined calibration pattern to estimate the ground truth point spread function (PSF) for left and right DP view. Then, they introduced a translating disk-shaped blur kernel based on the derived PSF which exhibits symmetric property between left and right. To provide a more realistic blur kernel that exhibits a donut-shaped depletion due to optical aberrations [23], Abuolaim et al. [3] proposed a parametric model based on the 2D Butterworth filter. To reflect the spatially-varying property, their work can generate many representative PSF by varying the parameter of the filter.

Recently, Ren et al. [17] proposed a joint optimization algorithm to solve for both estimating uniform blur kernel and generating sharp latent in blind motion deblurring task. Inspired by their work, we introduce to learn non-uniform blur kernels jointly with a sharp latent after some initialization step using the parametric DP blur model [3].

Chapter 3. Proposed Methods

This thesis proposes a scene-specific defocus deblurring method using DP images. The optimization is based on a zero-shot learning framework [20] that trains a scene-by-scene CNNs at test time using only the input image (i.e., blurred DP images). The proposed method does not require the ground-truth sharp image or ground-truth blur kernel. The unsupervised learning concept for defocus deblurring was first proposed by Xin et al. [29], where they jointly optimize for both latent image and the defocus map, given the ground-truth blur kernels. While their method only focuses on the front focus (i.e., objects behind the focal plane) defocus blur, we propose a more general approach to deal with the front and back focus using CNNs and without requiring the ground-truth blur kernel.

Instead of co-optimizing the latent image and defocus map, we optimize each one sequentially to make our problem easy to solve. First, we show that sharp image can be restored by only using a symmetrically property of parametric modeled blur kernel and describe on how to initialize the sharp latent map. Next, we introduce a method for estimating the defocus map by maximizing a posteriori probability of observed blurry image, given the initialized sharp latent map and the parametric modeled blur kernel. Finally, we further improve our model by learning scene-specific non-uniform blur kernel.

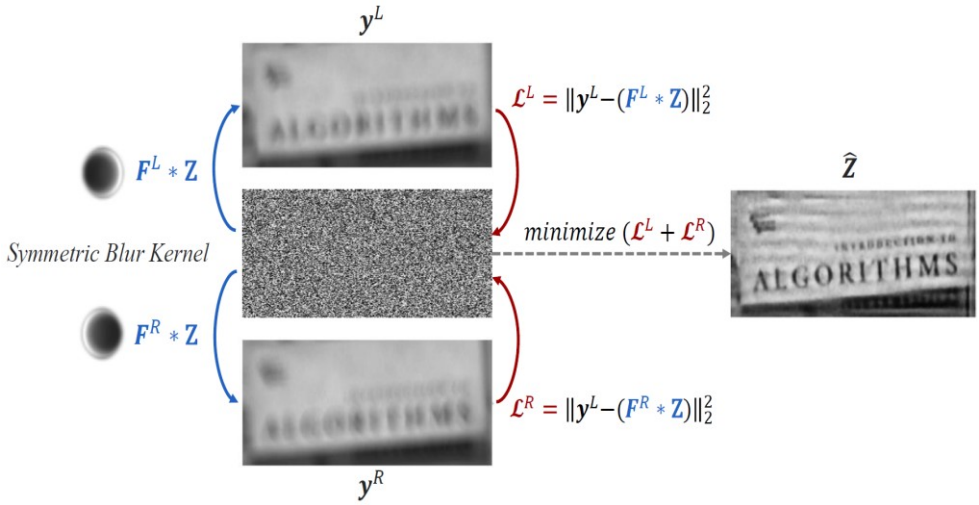

Fig 3.1: A sharp image can be restored using only the input image and a parameterized blur kernel of the correct size. A convolution is applied to a random vector $\boldsymbol{Z}$ with a left−right symmetric blur kernel to obtain a pair of blurry DP images. Then, the outputs of convolutions are optimized to minimize loss with respect to the input image. Because the parameterized left and right blur kernels are symmetric, we can restore sharp image, despite not using a ground−truth blur kernel.

3.1. Latent Map Initialization

The multiplane image (MPI) representation [22] consists of a latent map $\hat{\boldsymbol{X}} \in \mathbb{R}^{(N+1)\times H\times W}$ and alpha map $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{(N+1)\times H\times W}$, each with N fronto−parallel planes where the pixels in each plane are fixed at certain depth (i.e., or also in certain blur size). MPI can be used for defocus deblurring [29] since the size of the blur kernel is positively

related to the distance to the focal plane of the camera. Hence, each layer of MPI can represent the size of the blur kernel which increases as it reaches the last layer.

Now, let $\hat{X}_i \in \mathbb{R}^{H\times W}$ be the i^{th} layer of a latent map and $f_i^L \in \mathbb{R}^{(2i+1)\times(2i+1)}$ and $f_i^R \in \mathbb{R}^{(2i+1)\times(2i+1)}$ be the corresponding left-right blur kernel. Following Eq. (2), we can derive the uniformly-blurred DP images $\hat{y}_i^L \in \mathbb{R}^{H\times W}$ and $\hat{y}_i^R \in \mathbb{R}^{H\times W}$ for each layer by applying convolution to $\hat{X}_i$ with f_i^L and f_i^R, respectively.

$$\hat{y}_i^L = \hat{X}_i * f_i^L,$$
$$\hat{y}_i^R = \hat{X}_i * f_i^R, \tag{3}$$

where the blur kernel for each layer can be obtained by linearly downsizing from the maximum size kernel $f_N^L \in \mathbb{R}^{(2N+1)\times(2N+1)}$ and $f_N^R \in \mathbb{R}^{(2N+1)\times(2N+1)}$, and zero-padding afterward. Now using the downsized blur kernel for each layer, we can optimize for $\hat{X}_i$ by maximizing a posteriori probability given the observed blurry image. By minimizing the loss between $\hat{y}_i$ and the observed blurry image y, we can restore a partially sharp image for each and every layer of the latent map as shown in Fig. 3.1 (i.e., only the in-focus region for each layer will be restored). Although a parameterized blur kernel may not correctly represent the real blur kernel, its symmetric properties help and successfully restores sharp images.

Based on the above properties, we initialize the partially sharp latent map by optimizing every layer to maximize a posteriori of observed blurry image. Formally, we minimize the L2 loss as below,

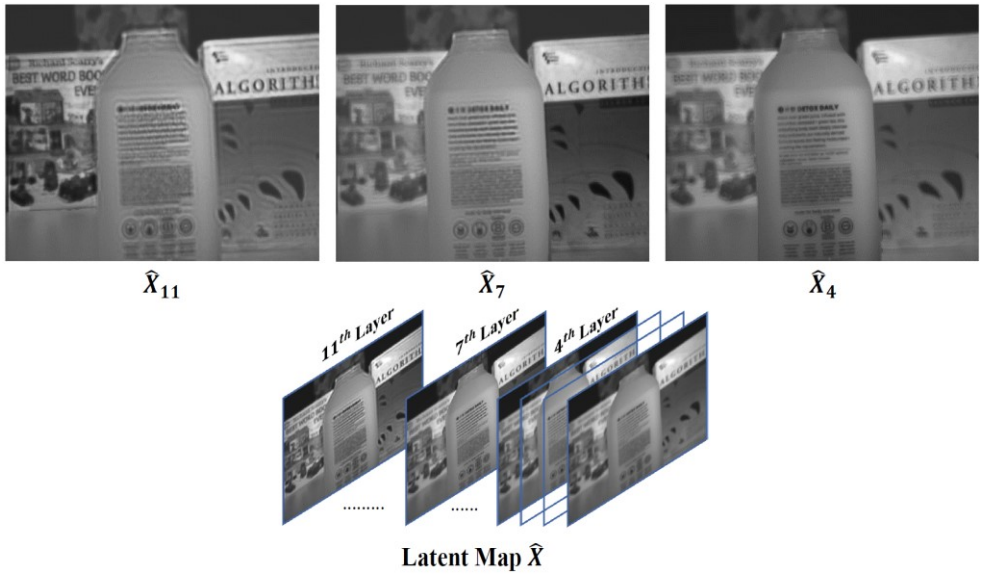

Fig 3.2: Visualization of the initialized latent map $\hat{\boldsymbol{X}}$ of MPI representation. Each layer X_i is blurred using a blur kernel of the corresponding size and optimized to minimize the loss as shown in Fig. 3.1. After the optimization, the in-focus regions of X_i are deblurred while other regions are not. The initialized latent map $\hat{\boldsymbol{X}}$ is then used for the defocus map estimation.

$$\mathcal{L} = \sum_{i=0}^{N} (\left\| y^L - \hat{y}_i^L \right\|_2^2 + \left\| y^R - \hat{y}_i^R \right\|_2^2). \tag{4}$$

The result of the latent map initialization can be seen in Fig. 3.2. Because the scene depth is not uniform, the amount of defocus blur varies across the image. Therefore, applying a uniform blur kernel to each layer results in successful deblurring in some in-focus regions (i.e., the correct size of blur kernel applied), while other regions are not. This means that in order to achieve image-level defocus deblurring, it is necessary to estimate the defocus map that encodes blur amount information for each pixel.

3.2. Defocus Map Estimation

In MPI representation, the latent image $\hat{x} \in \mathbb{R}^{H \times W}$ can be derived by soft−blending the latent map $\hat{\boldsymbol{X}}$ with the alpha map $\hat{\boldsymbol{\alpha}}$ using the over operator as below,

$$\hat{x}_k = \sum_{i=0}^{N} (\hat{X}_{i,k} \hat{\alpha}_{i,k} \prod_{i+1}^{N} (1 - \hat{\alpha}_{j,k})), \tag{5}$$

where k is the index for each pixel.

In soft−blending the latent map to the latent image, the alpha map should put more weights on in−focus layer of an MPI for each pixel. Therefore, estimating the index of an in−focus layer can be also seen as finding the correct blur kernel size for each pixel in a non−uniform defocus deblurring scenario. This aligns well with the defocus map estimation used in the literature where the defocus map is used to encode the amount of defocus blur per pixel in a defocus blurred image. In this thesis, we refer the term defocus map as $\widehat{\boldsymbol{D}} \in \mathbb{R}^{(N+1) \times H \times W}$ and define using the alpha map as below,

$$\widehat{D}_{i,k} = \hat{\alpha}_{i,k} \prod_{i+1}^{N} (1 - \hat{\alpha}_{j,k}), \tag{6}$$

where $\widehat{D}_{i,k} \in \mathbb{R}$ denotes the k^{th} pixel value from i^{th} layer of defocus map. In addition, the index to the pixel position will be omitted from hereafter. Then, Eq. (5) can be rephrased as,

$$\hat{x} = \sum_{i=0}^{N} \hat{X}_i \odot \widehat{D}_i, \tag{7}$$

where $\odot$ denotes the element−wise product, $\hat{X}_i \in \mathbb{R}^{H \times W}$, and $\widehat{D}_i \in \mathbb{R}^{H \times W}$. For estimating the defocus map $\widehat{\boldsymbol{D}}$, we train a CNN that takes

DP image pair as an input. To be specific, the CNN G_D uses DP images for utilizing the intrinsic focus cue and outputs an alpha map.

$$\hat{\alpha} = G_D(y^L, y^R). \tag{8}$$

The alpha map is then translated to a defocus map $\widehat{\boldsymbol{D}}$ using Eq. (6). Then, we blur each layer of the initialized latent map $\widehat{\boldsymbol{X}}$ with corresponding blur kernel f_i^L and f_i^R followed by the soft-blending using $\widehat{\boldsymbol{D}}$, from Eq. (7). Our final output is the non-uniform blurred DP images $\hat{y}^L$ and $\hat{y}^R$,

$$\begin{aligned} \hat{y}^L &= \hat{X}_0 \odot \widehat{D}_0 + \textstyle\sum_{i=1}^{N} (\hat{X}_i * f_i^L) \odot \widehat{D}_i, \\ \hat{y}^R &= \hat{X}_0 \odot \widehat{D}_0 + \textstyle\sum_{i=1}^{N} (\hat{X}_i * f_i^R) \odot \widehat{D}_i, \end{aligned} \tag{9}$$

where $\widehat{D}_0$ and $\hat{X}_0$ mean the layer without a defocus disparity for defocus map and latent map, respectively. By maximizing a posteriori probability given observed blurry images y^L and y^R, our framework can estimate the defocus map without using any ground-truth labels. For training the defocus map estimation network G_D, we minimize the $L2$ loss as below,

$$\mathcal{L} = \|y^L - \hat{y}^L\|_2^2 + \|y^L - \hat{y}^R\|_2^2. \tag{10}$$

Generalization for front and back focus

The overall shape of the blur is flipped horizontally depending on whether the scene point is in front or behind the focal plane of the DP image [16]. Following the work [3], we define front focus when

the blurry object is behind the focal plane and back focus when it is in front of the focal plane. Note that Eq. (9) and the prior work [29] is based on front-focus scenario only.

However, most images contain both front and back focus areas. Hence, we generalize Eq. (9) to adopt both the front and back focus of the scene. We define the latent map as $\widehat{\boldsymbol{X}} \in \mathbb{R}^{(2N+1)\times H\times W}$, defocus map $\widehat{\boldsymbol{D}} \in \mathbb{R}^{(2N+1)\times H\times W}$, back focus blur kernel as $\boldsymbol{b_N} \in \mathbb{R}^{(2N+1)\times(2N+1)}$, and the front focus blur kernel $\boldsymbol{f_N} \in \mathbb{R}^{(2N+1)\times(2N+1)}$. As the actual blur kernel for front and back focus differs [3, 29], two types of blur kernel should be considered. Then, Eq. (9) extends to,

$$\begin{aligned}\hat{y}^L &= \textstyle\sum_{i=-N}^{-1}(\hat{X}_i * b_i^L) \odot \widehat{D}_i + \hat{X}_0 \odot \widehat{D}_0 + \sum_{i=1}^{N}(\hat{X}_i * f_i^L) \odot \widehat{D}_i,\\ \hat{y}^R &= \textstyle\sum_{i=-N}^{-1}(\hat{X}_i * b_i^R) \odot \widehat{D}_i + \hat{X}_0 \odot \widehat{D}_0 + \sum_{i=1}^{N}(\hat{X}_i * f_i^R) \odot \widehat{D}_i. \end{aligned} \quad (11)$$

3.3. Learning Blur Kernels

Defocus blur is known to be non-uniform (i.e., spatially-varying), where its shape and density vary across the image. Also, the aspect of defocus blur can vary depending on the camera optic characteristics. Hence, it is not appropriate to apply a uniform blur kernel across images for defocus deblurring. After initializing the latent map and estimating the defocus map with a uniform parameterized blur kernel, we further improve our method by learning scene-specific non-uniform blur kernels.

Ren et al. [17] proposed to use a fully-connected network (FCN) to generate a uniform blur kernel using the DIP framework [25].

Although FCN is still a good option, the computation and memory requirements increase linearly as we have to estimate multiple non-uniform blur kernels. Furthermore, the defocus deblurring dataset is comprised of high-resolution images. Therefore, we find CNN more suitable as it requires fewer resources compared to FCN by taking advantage of spatial locality.

For front focus-only scenes, we train two generative networks G_F^L and G_F^R to predict left and right blur kernels. In the initialization phase, a symmetrically modeled blur kernel is used and only one network is required to estimate it. However, for further improvement, we found that it is better to use two separate networks for each blur kernel to compensate for some asymmetric properties that occur in the outer part of the image [29]. Each network takes a random tensor $\boldsymbol{Z} \in \mathbb{R}^{P\times(8N+4)\times(8N+4)}$ as an input and outputs estimated blur kernel $\widehat{\boldsymbol{F}}_{\boldsymbol{N}} \in \mathbb{R}^{P\times(2N+1)\times(2N+1)}$ as below,

$$\widehat{\boldsymbol{F}}_N^L = G_F^L(\boldsymbol{Z}^L),$$
$$\widehat{\boldsymbol{F}}_N^R = G_F^R(\boldsymbol{Z}^R). \tag{12}$$

Then, we apply the softmax layer to an output of the network to guarantee non-negativity and satisfy equality constraints that blur kernel should sum to 1. Note that the P is the number of non-uniform blur kernels. Next, we divide the latent map into P patches where each patch is blurred using corresponding blur kernels. The non-uniformly blurred latent map is then blended to latent image using the defocus map as in Eq. (9). Again, by minimizing the loss from Eq. (10), we learn to generate the scene-specific non-uniform

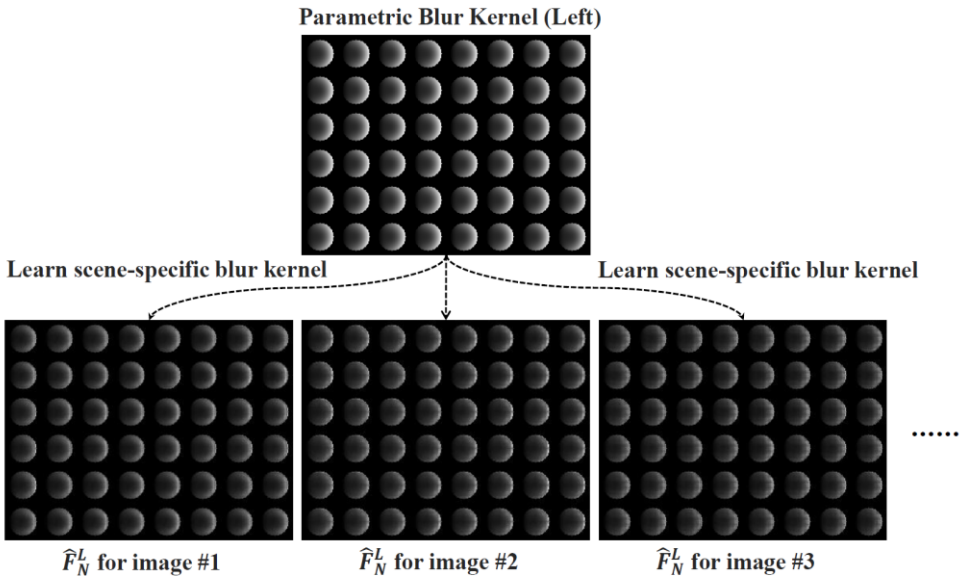

Fig 3.3: Visualization of non-uniform left blur kernels learned using a generative network. The network is first initialized to predict the parameterized blur kernel before the training. The latent map is divided into 8×6 patches, where each patch is blurred using a corresponding blur kernel to derive a blurred latent map. Then, we learn scene-specific non-uniform blur kernels by minimizing the loss from Eq. (10).

blur kernels as shown in Fig. 3.3. Note that the defocus map estimation network G_D is fixed, while G_F^L and G_F^R are jointly optimized with the latent map $\widehat{\boldsymbol{X}}$. To ease the optimization, we first initialize the network to predict the parameterized blur kernel. We also found that co-optimizing the input random tensor $\boldsymbol{Z}$ can make the network smaller and shorten the training time. For scenes that contain both front and back focus, we train four networks to predict left and right kernels for front and back focus blur, respectively.

3.4. Implementation Details

Network Architecture

We train a convolutional neural network (CNN) G_D for estimating the defocus map $\widehat{\boldsymbol{D}}$. G_D uses concatenated DP image as an input to utilize the intrinsic focus cue and outputs an alpha map $\widehat{\boldsymbol{\alpha}}$. The alpha map is then translated to the defocus map by Eq. (6), followed by a bilinear upsampling to match the spatial resolution of the original DP image. Then, interpolated defocus map is used for estimating the blurry DP image pairs $\hat{y}^L$ and $\hat{y}^R$, given the initialized latent map and the parameterized blur kernels as shown in Eq. (9). The detailed architecture for G_D is presented in Fig. 3.4.

Also, we train CNNs for generating scene-specific non-uniform blur kernels. For front focus-only scenes, we train two generative networks G_F^L and G_F^R to estimate blur kernels for left and right DP images. Each network takes a randomly initialized trainable tensor $\widehat{\boldsymbol{Z}} \in \mathbb{R}^{P\times(8N+4)\times(8N+4)}$ as an input and outputs estimated blur kernels $\widehat{\boldsymbol{F}}_{\boldsymbol{N}} \in \mathbb{R}^{P\times(2N+1)\times(2N+1)}$ followed by the softmax layer. The blur kernel from each layer (i.e. or channel) of $\widehat{\boldsymbol{F}}_{\boldsymbol{N}}$ is then applied to P patches of the latent map. The detailed architecture for G_F is also shown in Fig. 3.4.

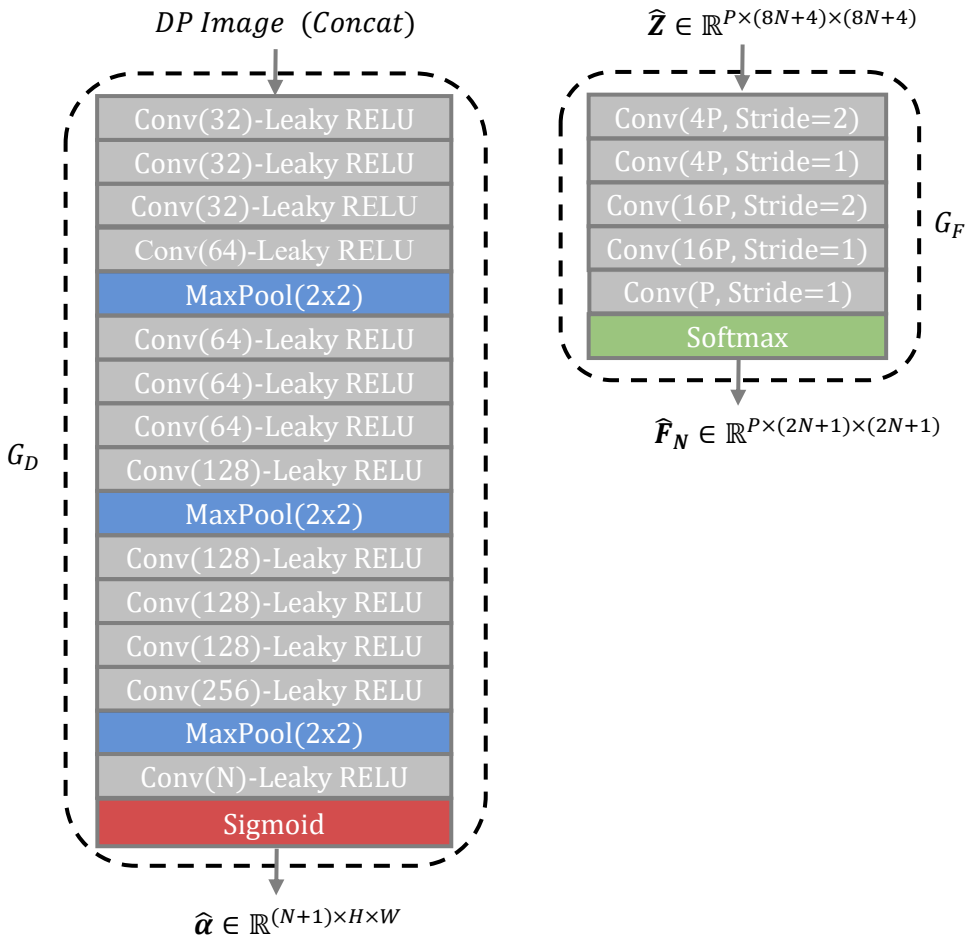

Fig 3.4: A detailed architecture for G_D (Left) and G_F (Right). Left: G_D takes concatenated DP image as an input and outputs estimated alpha map which is translated to a defocus map. Right: G_F estimates the non-uniform blur kernels from randomly initialized tensor $\hat{z}$, which is jointly optimized with the network.

Regularization Terms

We introduce two regularization terms that was effective in improving the performance of the proposed method: Smoothness on

Table 3.1: Ablation study on the regularization terms. $\mathcal{L}$ refers to the loss term from Eq. (10).

Method	Google Pixel dataset		
	PSNR ↑	SSIM ↑	MAE ↓
$\mathcal{L} + \mathcal{L}_{latent}$	32.549	0.866	0.0145
$\mathcal{L} + \mathcal{L}_{defocus}$	32.637	0.867	0.0140
$\mathcal{L} + \mathcal{L}_{defocus} + \mathcal{L}_{latent}$	32.730	0.872	0.0139

defocus image and latent image. The defocus image $\hat{d} \in \mathbb{R}^{H \times W}$ can be derived from the defocus map $\widehat{\boldsymbol{D}} \in \mathbb{R}^{(N+1) \times H \times W}$ as below,

$$\hat{d}_k = \sum_{i=0}^{N} (2i+1)\widehat{D}_k, \tag{13}$$

where $\hat{d}_{\boldsymbol{k}} \in \mathbb{R}$ and $\widehat{D}_{\boldsymbol{k}} \in \mathbb{R}^{N+1}$ denotes the k^{th} pixel value. Then, we apply Total Variation (TV) regularization on $\hat{d}_{\boldsymbol{k}}$ as below,

$$\mathcal{L}_{defocus} = \frac{1}{H \times W} \sum_k TV(\hat{d}_k). \tag{14}$$

Also, to encourage smoothness on the latent sharp image $\hat{x} \in \mathbb{R}^{H \times W}$, we adopt an edge-aware smoothness regularization term defined as below,

$$\mathcal{L}_{latent} = \frac{1}{H \times W} \sum_k V_E(\hat{x}, E(\hat{x})). \tag{15}$$

where E means an edge map and V_E refers to the edge-aware smoothness term introduced in [29]. The ablation study of regularization terms is presented in Table 3.1.

Chapter 4. Experiments

We evaluate our method on the Canon DSLR dataset [2] and the Google Pixel dataset from [29]. To check the generalization performance, we first compare our method with recent state-of-the-art defocus deblurring models [2, 3, 11, 29] on the Pixel dataset, where supervised methods are all trained using the Canon dataset. We then demonstrate the general in-distribution performance of our method on the Canon dataset, which serves as a public benchmark for DP defocus deblurring.

4.1. Dataset

Google Pixel Dataset

Smartphone cameras have a fixed narrow aperture and exhibit a large depth of field (DOF) that differs from the variable aperture of a DSLR camera. Smartphone cameras use a focus motor to adjust focus by changing the distance between the lens and the image sensor [32] when the DSLR camera adjusts the aperture size. Hence, the aspect of defocus deblur is different between the two cameras [16, 29]. Xin et al. [29] captured a uniformly sampled focus stack with the nearest focal distance corresponding to $13.7cm$ and the furthest to the infinity. Then, they generate ground-truth sharp images and defocus maps using commercial software. The dataset provides 17 front focus-only scenes, including indoor and outdoor,

for evaluation. The image is in grayscale raw format and cropped to 1008×1344 resolution with vignetting correction.

Canon DSLR Dataset

DSLR cameras change aperture size to adjust the focus. A wide aperture gets more light at the expense of a shallow DOF that causes defocus blur in areas outside the DOF. On the other hand, a narrower aperture results in a greater DOF, but at the expense of light. Abuolaim et al. [2] captured a pair of DP images of the same static scene at two aperture sizes: the widest and narrowest aperture size possible in a lens configuration. The image captured at the narrowest aperture is used as a ground-truth sharp image and the image captured at the widest aperture serves as a blurry image to construct the dataset. The dataset includes both front-focused and back-focused scenes, providing 500 pairs of indoor and outdoor scenes in sRGB format at 1120×1680 resolution.

4.2. Quantitative Results

Google Pixel Dataset

For the Pixel dataset, we use an MPI with 18 layers with a maximum blur kernel of size 35×35. Due to the sequential optimization pipeline and the use of CNNs, the full optimization takes

Table 4.1: Quantitative comparison with the recent defocus deblurring methods [2, 3, 11, 29] on the Google Pixel dataset. The supervised methods are trained using Canon dataset and tested on the raw DP images captured from a smartphone camera. Our method is better in every image quality metrics, despite being unsupervised.

Method	Google Pixel dataset			Learning
	PSNR ↑	SSIM ↑	MAE ↓	
Xin et al. [29]	30.150	0.826	0.0166	Unsupervised
DPDNet [2]	31.559	0.849	0.0165	Supervised
RDPD [3]	32.246	0.861	0.0149	Supervised
IFAN [11]	31.985	0.862	0.0150	Supervised
Ours	32.730	0.872	0.0139	Unsupervised

6 minutes on NVIDIA A100 GPU, which is $20\times$ faster than [29]. Our method outperforms all the other supervised methods [2, 3, 11] in quantitative metrics as shown in Table 4.1. Therefore, it can be said that data-driven approaches using one type of camera do not generalize well to other types of cameras.

Canon DSLR Dataset

As we ran our experiments, we found that the size of defocus blur on a DSLR camera is about twice the size of defocus blur from a smartphone camera. Hence, the maximum blur kernel size should reach around $60-70$ pixels to properly remove the defocus blur.

Table 4.2: Quantitative comparison with recent supervised defocus deblurring methods [2, 3, 11] on the Canon DSLR dataset. Note that the performance is measured at 560×848 which is half the resolution of the original image. Although our method is unsupervised, the performance is competitive to other supervised methods.

Method	Canon DSLR Dataset			Learning
	PSNR ↑	SSIM ↑	MAE ↓	
DPDNet [2]	25.47	0.805	0.040	Supervised
RDPD [3]	25.60	0.783	0.040	Supervised
IFAN [11]	26.54	0.828	0.036	Supervised
Ours	25.52	0.800	0.041	Unsupervised

Furthermore, we should consider the front-back focus scene for the Canon dataset, doubling the number of layers used in MPI. This made it difficult to apply our method and had to halve the image resolution in each dimension. By reducing the resolution, we were able to use an MPI with 35 layers with a maximum blur kernel of size 35×35. Although our method is unsupervised, the quantitative performance in for in-distribution data is competitive to other supervised methods.

4.3. Qualitative Results

We provide visualizations of defocus deblurred images and qualitative comparison with other supervised methods in this section., The proposed method is capable of restoring defocus deblurred

images without using any ground-truth labels, as shown in Fig. 4.1.

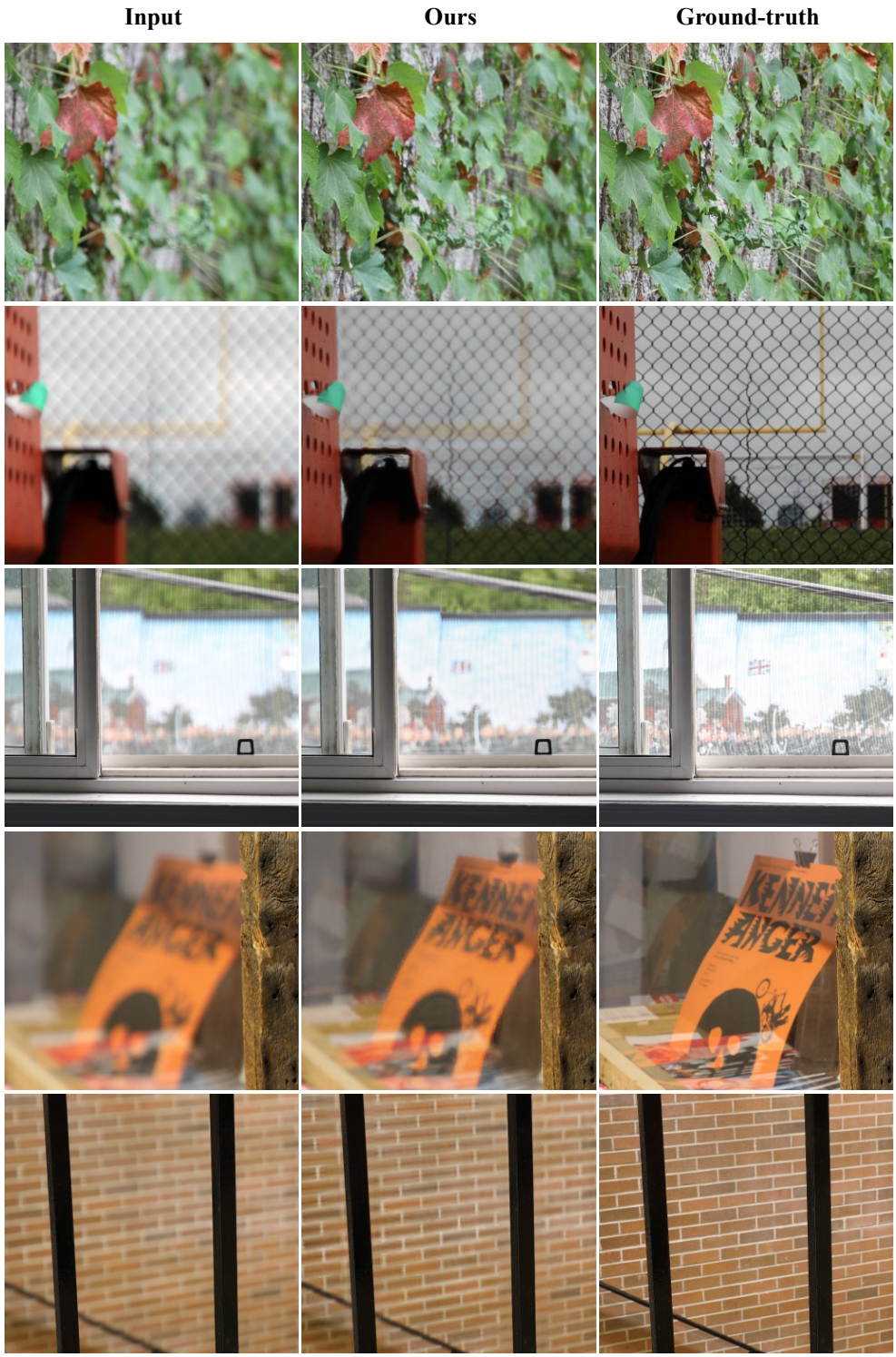

Fig 4.1: Visualization of the proposed defocus deblurring method. The first column is the blurry image, second is the restored image, and the last column is the ground-truth image.

Also, our method is better than recent supervised methods in qualitative point of view as shown in Fig. 4.2. For out-of-distribution scenarios, trained on DSLR image and tested on smartphone images,

supervised methods fail to restore some areas and introduce aliasing. Therefore, it shows that data-driven approaches using one type of camera do not generalize well to other types of cameras.

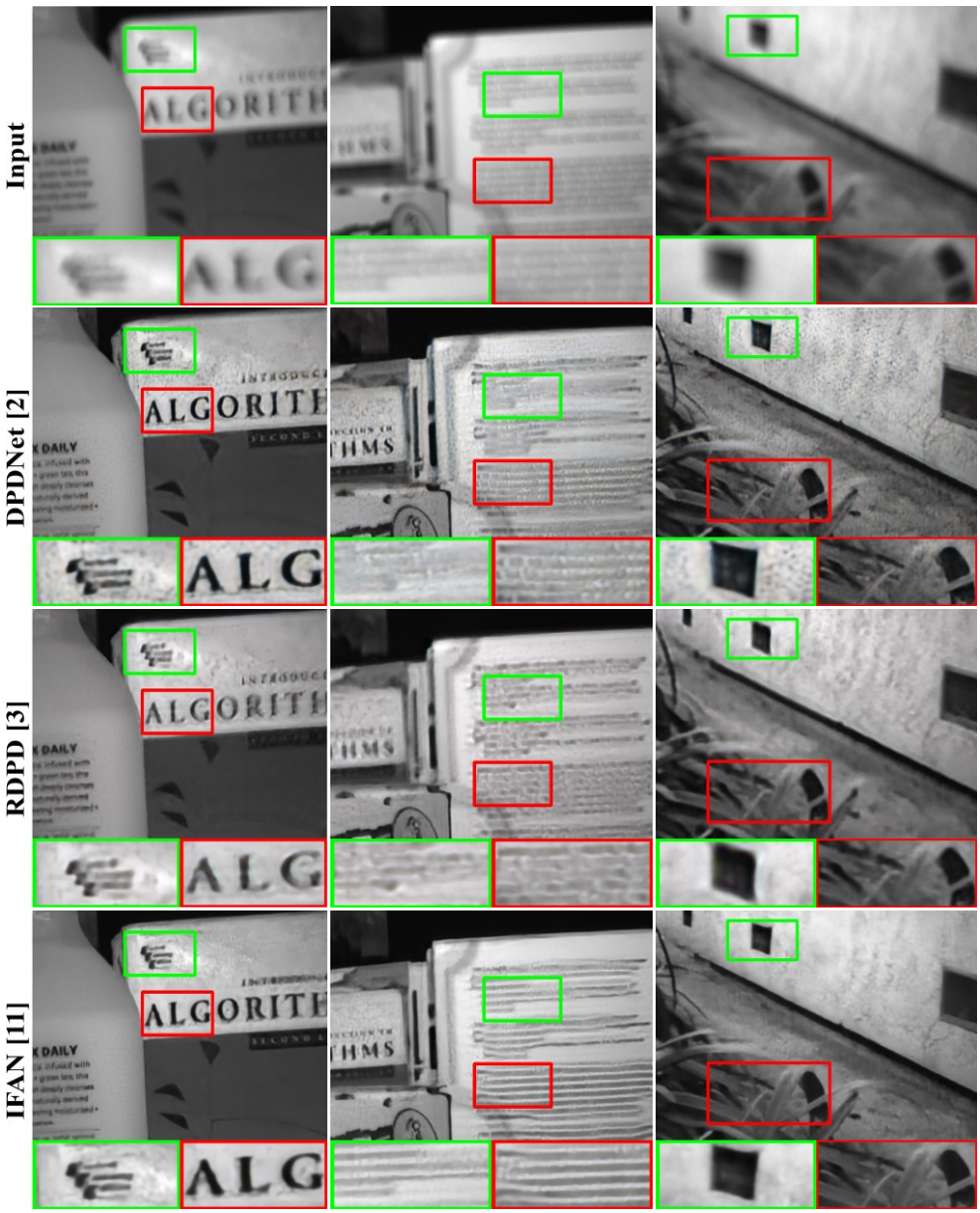

Fig 4.2: We compare with supervised methods [2, 3, 11] on the Google Pixel dataset. The first row is the blurry image and the last row is the ground-truth image. Our method shows less failure in defocus deblurring compared to other supervised methods.

Despite being unsupervised, the proposed methods also shows competitive image quality compared to supervised methods for in-distribution scenario. When trained and tested on the Canon DSLR dataset, our method is robust and fails less than the other supervised methods, as shown in Fig. 4.3.

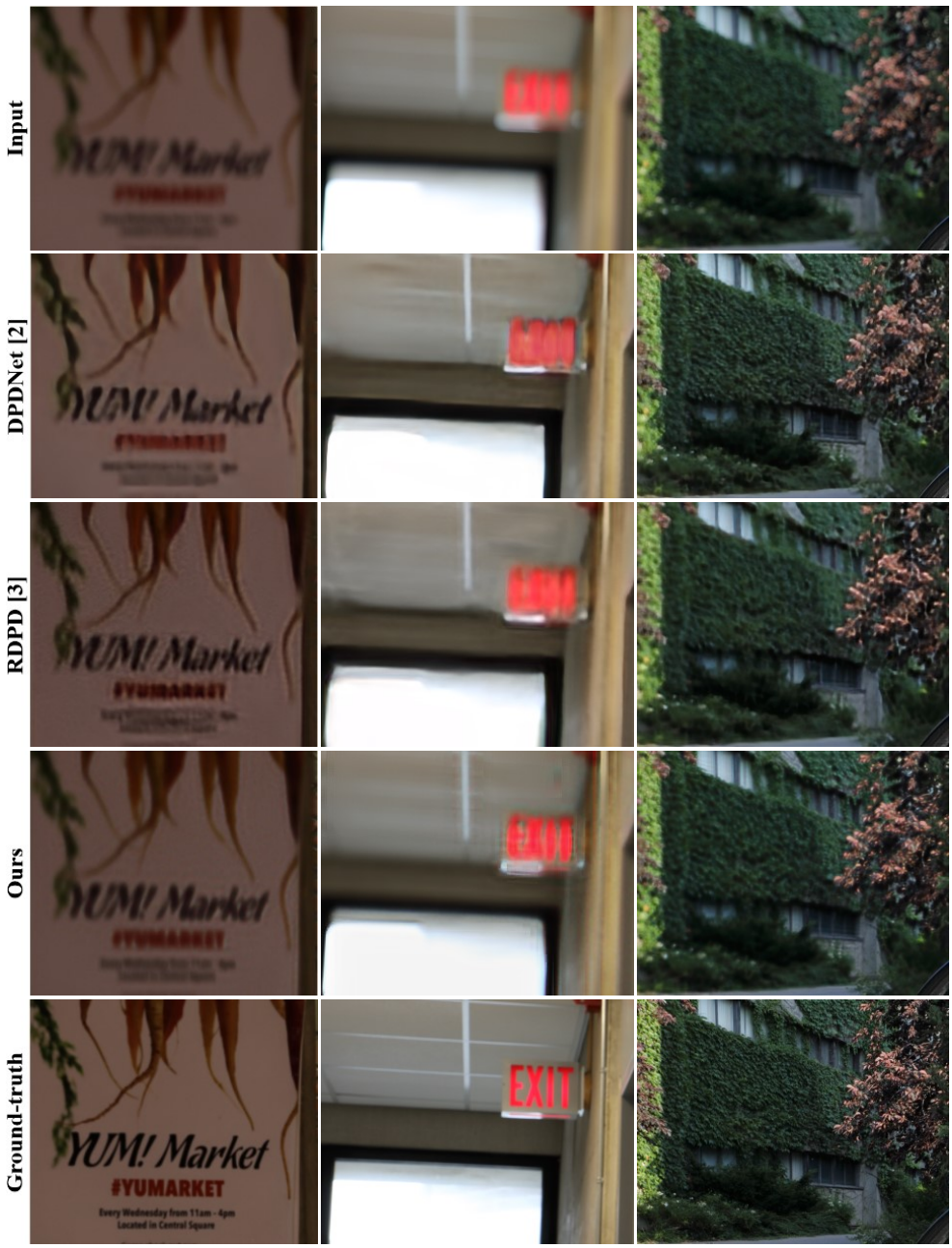

Fig 4.3: Qualitative comparisons with supervised methods [2, 3] on the Canon DSLR dataset. The first row is the blurry image and the last row is the ground-truth image. Our method shows competitive performance to supervised methods, despite being unsupervised.

Chapter 5. Conclusions

5.1. Summary

This thesis proposed a zero-shot defocus deblurring method using DP images. Proposed method does not require training data and can be used in defocus deblurring smartphone DP images, where it is difficult to collect large datasets. It shows competitive performance to other supervised methods, despite being unsupervised. We utilize the parametric blur kernel and its symmetric property in defocus deblurring. Although real DP blur kernels may be symmetric only to some extent [3, 29], we showed that symmetric modeling can still help to restore sharp image. By initializing a latent map first and then using a CNN for estimating the defocus map, we were able to speed up the optimization time by 20× compared with [29]. Then, we fix the defocus map and learn for scene-specific left and right non-uniform blur kernels using CNNs while jointly optimizing for the sharp latent image.

5.2. Discussion

In this thesis, we demonstrated that current state-of-the-art supervised defocus deblurring models [2, 3, 11] do not generalize well on defocus blurred images captured from the different cameras

it was trained on. Because our method does not require a ground-truth blur kernel, it can be easily used in the absence of training data for smartphone cameras. Even in the presence of training data, it is difficult to handle various smartphone cameras because of different optical designs due to various pixel structures (i.e. Non-bayer CFAs) and smaller pixel size. Hence, collecting device-dependent training data will become very expensive and labor-intensive task. Therefore, we think that future research for the supervised defocus deblurring should also consider camera-agnostic methods. We believe that the proposed method can play an important role in such direction.

Bibliography

[1] Abuolaim, Abdullah, Mahmoud Afifi, and Michael S. Brown. "Multi-View Motion Synthesis via Applying Rotated Dual-Pixel Blur Kernels." In WACV, 2022.

[2] Abuolaim, Abdullah, and Michael S. Brown. "Defocus deblurring using dual-pixel data." In ECCV, 2020.

[3] Abuolaim, Abdullah, et al. "Learning to reduce defocus blur by realistically modeling dual-pixel data." In ICCV, 2021.

[4] Abuolaim, Abdullah, Radu Timofte, and Michael S. Brown. "NTIRE 2021 challenge for defocus deblurring using dual-pixel images: Methods and results." In CVPR, 2021.

[5] Chan, Tony F., and Chiu-Kwong Wong. "Total variation blind deconvolution." IEEE Transactions on Image Processing (TIP), 1998.

[6] D'Andrès, Laurent, et al. "Non-parametric blur map regression for depth of field extension." IEEE Transactions on Image Processing (TIP), 2016.

[7] Garg, Rahul, et al. "Learning single camera depth estimation using dual-pixels." In ICCV, 2019.

[8] Karaali, Ali, and Claudio Rosito Jung. "Edge-based defocus blur estimation with adaptive scale selection." IEEE Transactions on Image Processing (TIP), 2017.

[9] Krishnan, Dilip, and Rob Fergus. "Fast image deconvolution using hyper-Laplacian priors." In NIPS, 2009.

[10] Lee, Junyong, et al. "Deep defocus map estimation using domain adaptation." In CVPR, 2019.

[11] Lee, Junyong, et al. "Iterative filter adaptive network for single image defocus deblurring." In CVPR, 2021.

[12] Mannan, Fahim, and Michael S. Langer. "Blur calibration for depth from defocus." In Computer and Robot Vision (CRV), 2016.

[13] Mosleh, Ali, et al. "Camera intrinsic blur kernel estimation: A reliable framework." In CVPR, 2015.

[14] Oliveira, Joao P., Mario AT Figueiredo, and Jose M. Bioucas-Dias. "Parametric blur estimation for blind restoration of natural images: Linear motion and out-of-focus." IEEE Transactions on Image Processing (TIP), 2013.

[15] Pan, Liyuan, et al. "Dual pixel exploration: Simultaneous depth estimation and image restoration." In CVPR, 2021.

[16] Punnappurath, Abhijith, et al. "Modeling defocus-disparity in dual-pixel sensors." International Conference on Computational Photography (ICCP), 2020.

[17] Ren, Dongwei, et al. "Neural blind deconvolution using deep priors." In CVPR, 2020.

[18] Richardson, William Hadley. "Bayesian-based iterative method of image restoration." JoSA 62.1 (1972): 55-59.

[19] Shi, Jianping, Li Xu, and Jiaya Jia. "Just noticeable defocus blur detection and estimation." In CVPR, 2015.

[20] Shocher, Assaf, Nadav Cohen, and Michal Irani. "“zero-shot” super-resolution using deep internal learning." In CVPR, 2018.

[21] Srinivasan, Pratul P., et al. "Pushing the boundaries of view

extrapolation with multiplane images." In CVPR, 2019.

[22] Szeliski, Richard, and Polina Golland. "Stereo matching with transparency and matting." Sixth International Conference on Computer Vision, 1998.

[23] Tang, Huixuan, and Kiriakos N. Kutulakos. "Utilizing optical aberrations for extended-depth-of-field panoramas." In ACCV, 2012.

[24] Tucker, Richard, and Noah Snavely. "Single-view view synthesis with multiplane images." In CVPR, 2020.

[25] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." In CVPR, 2018.

[26] Vasu, Subeesh, Venkatesh Reddy Maligireddy, and A. N. Rajagopalan. "Non-blind deblurring: Handling kernel uncertainty with cnns." In CVPR, 2018.

[27] Wadhwa, Neal, et al. "Synthetic depth-of-field with a single-camera mobile phone." ACM Transactions on Graphics (ToG), 2018.

[28] Wiener, Norbert, et al. Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications. Vol. 113. No. 21. Cambridge, MA: MIT press, 1949.

[29] Xin, Shumian, et al. "Defocus Map Estimation and Deblurring from a Single Dual-Pixel Image" In ICCV, 2021.

[30] Zamir, Syed Waqas, et al. "Restormer: Efficient Transformer for High-Resolution Image Restoration." In CVPR, 2022..

[31] Zhang, Yinda, et al. "Du 2 net: Learning depth estimation from dual-cameras and dual-pixels." In ECCV, 2020.

[32] Zhang, Yupeng, et al. "Autofocus system and evaluation

methodologies: a literature review." Sens. Mater 30.5 (2018): 1165-1174.

[33] Zhou, Tinghui, et al. "Stereo magnification: Learning view synthesis using multiplane images." arXiv preprint arXiv:1805.09817 (2018).

[34] Son, Hyeongseok, et al. "Single image defocus deblurring using kernel-sharing parallel atrous convolutions." In ICCV. 2021.

[35] Zhuo, Shaojie, and Terence Sim. "Defocus map estimation from a single image." Pattern Recognition 44.9 (2011): 1852-1858.

Abstract in Korean

듀얼 픽셀(DP) 이미지 센서를 사용하는 스마트폰에서의 Defocus Blur 현상은 다양한 카메라 광학 구조와 물체의 깊이 마다 다른 흐릿함 정도로 인해 원 영상 복원이 쉽지 않습니다. 기존 알고리즘들은 모두 Canon DSLR 데이터에서 훈련된 지도 학습 접근 방식에 의존하여 스마트폰으로 촬영된 사진에서는 잘 일반화가 되지 않습니다. 본 논문에서는 훈련 데이터와 사전 보정된 실제 Blur 커널 없이도, 한 쌍의 DP 사진만으로도 학습이 가능한 Zero-shot Defocus Deblurring 알고리즘을 제안합니다. 특히, 본 논문에서는 대칭적으로 모델링 된 Blur Kernel을 사용하여 초기 영상을 복원하며, 이후 CNN(Convolutional Neural Network)을 사용하여 관찰된 DP 이미지를 가장 잘 설명하는 Defocus Map을 추정합니다. 마지막으로 CNN을 사용하여 장면 별 Non-uniform한 Blur Kernel을 학습하여 최종 복원 영상의 성능을 개선합니다. 학습과 추론이 다른 데이터 세트에서 실행될 때, 제안된 방법은 비지도 기술 임에도 불구하고 최근에 발표된 지도 학습을 기반의 방법들보다 우수한 성능을 보여줍니다. 또한 학습 된 것과 같은 분포 내 데이터에서 추론할 때도 지도 학습 기반의 방법들과 정량적 또는 정성적으로 비슷한 성능을 보이는 것을 확인할 수 있었습니다.