Deep Learning for Image Deblurring and Reflection Removal

Jie YANG
School of Computer Science
The University of Adelaide

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY

Supervised by:
Prof. Javen Qinfeng Shi
Dr. Lingqiao Liu

May 17, 2021

Contents

List of Figures

List of Tables

UNIVERSITY OF ADELAIDE

Abstract

Faculty of ECMS
School of Computer Science

Doctor of Philosophy

Deep Learning for Image Deblurring and Reflection Removal

by Jie YANG

This thesis focuses on two highly ill-posed inverse problems in low-level computer vision, *i.e.* image deblurring and reflection removal. Digital photos taken in the real-world are likely to suffer from certain types of degration, for example, the motion of camera and objects cause image blur, lights from objects in front of glass lead to reflections that will obstruct the background behind the glass, *etc*. While in some scenarios image blur and reflections may be appealing to photographers, more often they are undesirable, and both image blur and reflections can reduce the performance of other computer vision systems. In those situations, it is significant to obtain clear sharp images from corrupted ones by image deblurring and reflection removal. Image deblurring aims to recover the sharp image alone or with the blur kernel and reflection removal aims to recover the clear background image alone or with the reflection image.

We focus to use deep learning based approach to address the image blurring and reflection removal problem in this thesis. Conventional methods usually rely on manually defined priors and image features, which may not reflect the nature of real data and the type and range of blur and reflections that can be handled are limited. By learning from data, we are able to model more general image blur and reflections.

For image deblurring, we focus on removing pixel-wise heterogeneous motion blur. We propose a fully convolutional network to estimate a dense motion flow from a blurry image and recovers the clear image from the estimated motion flow. Learning a prior over the latent image would require modeling all possible image content, while an easier task is to learn the motion flow instead, which allows the model to focus on the cause of the blur, irrespective of the image content. Our network is the first universal end-to-end mapping from the blurred image to the dense motion flow. To train the network, we simulate motion flows to generate synthetic blurred-image-motion-flow pairs. The proposed method outperforms the state-of-the-art on both synthetic and challenging realistic blurred images.

We address the reflection removal problem in two different approaches. The first is through supervised learning which requires mixed-background-reflection image triplets as training data. To obtain sufficient training data, we propose to simulate the reflections from two clear images, which represent background and reflection layer respectively, using a general reflection model. To remove reflection truly well, we argue that it is essential to estimate the reflection and utilize it to estimate the background image. We propose a cascade neural network to estimate both the background

image and the reflection. The network uses the estimated background image to estimate the reflection, and then use the estimated reflection to estimate the background image, which significantly improves reflection removal.

The second approach is through self-supervised learning, which alleviates the necessity of ground-truth training data. We propose a reflection removal framework relying on learning from real-world image pairs with reflection taken from multiple views. Our method only relies on the supervision from the geometry correspondence and consistency between the multi-view consistency. A series of novel consistency losses are introduced to effectively and robustly utilize the imperfect cues derived from the multi-view consistency. By training on easily obtained real data without ground-truth, the model generalizes better on real-world images.

Publications and Other Academic Work

During my Ph.D study, I have the following publications (including published and prepared to be submitted):

- Dong Gong, **Jie Yang**, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017 (My contribution includes the design and implementation of the neural network, conduction and analysis of experiments)
- **Jie Yang**, Dong Gong, Lingqiao Liu, Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018
- Yinglong Wang, Dong Gong, **Jie Yang**, Qinfeng Shi, Anton van den Hengel, Dehua Xie, Bing Zeng. Deep Single Image Deraining via Modeling Haze-like Effect. IEEE Transactions on Multimedia, 2020
- **Jie Yang**, Dong Gong, Lingqiao Liu, Mingkui Tan, Anton van den Hengel, Qinfeng Shi. Self-supervised image reflection removal with multi-view consistency cues. Prepared to be submitted to International Joint Conference on Artificial Intelligence (IJCAI), 2021

And the following technical report:

- **Jie Yang**, Dong Gong, Lingqiao Liu, Qinfeng Shi. Deep neural networks for music source separation. 2019

This thesis contains the aforementioned work in image deblurring and reflection removal, *i.e.*

- From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur
- Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal
- Self-supervised image reflection removal with multi-view consistency cues

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Jie Yang

May 2021

Acknowledgements

First of all, I would like to express my deepest gratitude to my principle supervisor Prof. Qinfeng Shi, for the continuous support of my Ph.D study and research. He is always able to look into the essence of the problem, which deeply impresses me. Under his instructions, I laid a good foundation of machine learning and optimization, which proves to be significant to the further research in deep learning.

The advise from my co-supervisor, Dr. Lingqiao Liu, is also very inspiring. His insights, knowledge and experiences in deep leaning and computer vision helped me a lot in my research.

My sincere thanks also go to Dr. Dong Gong. Without him, I would not enter the field of low-level vision. He gave me numerous guidance, from experiments to academic writing. The discussions with him always enlighten me.

Also, thanks other professors in the school. Prof. Chunhua Shen gave me valuable advice in the work of image deblurring. Thanks Prof. Anton van den Hengel and Prof. Ian Reid for editing my papers. Prof. Ian Reid also gave me very useful advice during my major review. My colleagues and friends in the lab, Qianggong Zhang, Ming Cai, Zhipeng Cai, Rafael Felix, helped and supported me a lot, not only in research but also in daily life.

Thanks to all my diving instructors and Adelaide University Scuba Club. Diving into the underwater world makes me peace when I feel anxious in my research.

I also would like to thank Wei Li, Gengwen Liu, Chen Chen and Liying Yu. They are the best housemates I ever had. The time we lived together is so memorable.

Most importantly, I need to appreciate my parents, who supported me all the time, especially during my darkest time. Without them, I cannot make through it.

Chapter 1

Introduction

Contents

This chapter provides an introduction to the problems this thesis aims to address. We also detail the objectives, methods and the main contributions of our work.

1.1 Image blurring and Reflection Removal

There are three stages of computer vision: low-level vision, mid-level vision and high-level vision. While high-level vision usually attracts more attention, low-level vision is a fundamental aspect of computer vision. Low-level vision focuses on pixel-level information, and involves extracting fundamental image primitives, like edges and corners, and performing filtering and morphology, *etc*. More specifically, in this thesis we will target at image enhancement, which is a typical type of low-level vision problems. Due to the increasing popularity of using smartphones to capture moments in daily life and demands to deploy surveillance cameras in public for safety, enormous photos and videos are taken by non-professional devices nowadays. Although the image sensors and lens in such devices are developing rapidly, the compact size of the device still limits the capabilities of the embedded photographic components. For example, a small device is more prone to the shake during photographing. Moreover, some types of contamination is difficult to avoid even with professional cameras, for instance reflections from transparent glass such as windows. Therefore, it is significant to leverage the power of algorithms to compensate the limited capabilities of cameras to produce high quality images.

Some typical low-level computer vision tasks include image denoising (Buades, Coll, and Morel, 2005; Mairal et al., 2009; Gu et al., 2014), image deblurring (Chan and Wong, 1998; Krishnan, Tay, and Fergus, 2011; Whyte et al., 2012), image super resolution (Freeman, Jones, and Pasztor, 2002; Glasner, Bagon, and Irani, 2009; Yang et al., 2010), high dynamic range (HDR) imaging (Debevec and Malik, 2008; Granados et al., 2010; Reinhard et al., 2010), image reflection removal (Li and Brown, 2014), image inpainting (Bertalmio et al., 2000), *etc*. These problems, especially when the input is a single image, usually have ill-posed nature, thus are very challenging to solve.

Studying these problems has various benefits. Firstly, generating images of better quality in itself is appealing to the photographers, especially when the device is not professional. Some of these techniques are now being incorporated into modern digital cameras. For example, almost all latest cameras have HDR imaging and panorama imaging functions built-in, and some smartphones utilize algorithms to achieve high performance in low-light conditions comparable to DSLR cameras, *etc*.

Moreover, obtaining higher quality images is a foundation to other mid-level and high-level computer vision tasks. For computer vision systems, the perception of the physical world relies on digital images or videos as input and the quality of the input images or videos has a direct impact on the performance and robustness of the system. For example, for text recognition, if the target text is very blurry and contaminated with some reflections, then it is difficult to correctly recognize the content. Therefore, applying low-level vision algorithms to enhance the input sources can be a good pre-processing for high-level computer vision tasks.

In this thesis, we mainly focus on two particular problems mentioned above: image deblurring and image reflection removal.

1.1.1 Image Deblurring

Photographs taken in real world sometimes suffer from blurring, which is mainly caused by the movement of objects, camera shake or defocus. From an artistic perspective, blur is sometimes intentional in photography, for example, blurry background due to shallow depth of field can highlight the foreground, and the motion blur in background can create a sense of movement in sports photography. However photographers want to avoid unintended blur in images and for majority of the image analysis applications blurs ruins useful data, for instance, a blurred object is more difficult to be recognized than a clear object.

The problem of image deblurring is to restore a latent sharp image from a blurred image. According to whether the blur kernel is known, image deblurring problems can be categorized into two types: non-blind deblurring and blind deblurring. Non-blind deblurring indicates that the blur kernel is assumed to be known and a sharp image can be induced from both the blurry image and the blur kernel. By contrast, blind deblurring refers to the situation where the blur kernel is unknown, and the task therefore becomes estimating both the clear image and the blur kernel from the degraded image.

A blurred image $\mathbf{Y}$ can be modeled as

$$\mathbf{Y} = \mathcal{K} * \mathbf{X} + \mathbf{N}, \tag{1.1}$$

where $\mathbf{X}$ denotes the latent sharp image, $\mathbf{N}$ refers to additive noise, and $\mathcal{K}$ denotes the blur kernel. Traditionally, the blur kernel is usually assumed to be spatially invariant. However, this assumption is easily violated by complex motion or other factors in reality. To model more general spatially variant blur, the blur kernel should be heterogeneous, which means each pixel can have different values in the blur kernel map. Numerous methods have been proposed to solve both non-blind (Bar, Kiryati, and Sochen, 2006; Krishnan and Fergus, 2009; Zoran and Weiss, 2011; Schuler et al., 2013; Schmidt et al., 2013) and blind deblurring problems (Fergus et al., 2006; Jia, 2007; Cho and Lee, 2009; Krishnan, Tay, and Fergus, 2011; Whyte et al., 2012; Pan et al., 2016a). In this thesis, we mainly focus on blind image deblurring since it is more practical and also more challenging, which is difficult to be addressed by conventional methods.

1.1.2 Reflection Removal

In urban environment and indoor scenes, glass is ubiquitous. Photos taken through transparent glass usually contains reflections that interfere with the background content. Similar to image blurs, although reflections are aesthetic in some scenarios, they can be annoying when people want to focus on the targets behind the glass and the reflections in image may hinder the performance of other vision perceptual systems. The problem of reflection removal is to recover the clear image without reflections. Reflection removal problem can be viewed as an image decomposition into separate layers, *i.e.* background layer and reflection layer in this case. There are infinite possible decompositions of an image into layers, thus it is also a very ill-posed problem. Compared to similar problems such as rain removal or fence removal, where rain and fence have relatively fixed patterns, the pattern of reflections is more diverse, which makes the two layers more ambiguous to separate.

The image reflection can be modeled as:

$$\mathbf{I} = \mathbf{B} + \mathbf{R} \tag{1.2}$$

where $\mathbf{I}$ denotes the observed image, $\mathbf{B}$ denotes the background layer behind the glass, and $\mathbf{R}$ denotes the reflection layer from the other side of the glass. Different from image blur model in Equation1.1, where $\mathcal{K}$ and $\mathbf{N}$ are presumed to be drawn from specific distributions, in the reflection model, on the contrary, $\mathbf{B}$ and $\mathbf{R}$ can both be natural images in similar conditions.

According to the number of images used, the reflection removal can be divided into two categories: multiple image reflection removal and single image reflection removal. Multiple image reflection removal employ images from various view points (Farid and Adelson, 1999; Szeliski, Avidan, and Anandan, 2000; Sarel and Irani, 2004; Gai, Shi, and Zhang, 2012; Sinha et al., 2012; Li and Brown, 2013; Guo, Cao, and Ma, 2014; Xue et al., 2015; Yang et al., 2016), or capture settings (Agrawal et al., 2005; Schechner, Kiryati, and Basri, 2000), with the aim of exploiting additional information to separate the reflection artifacts from the observed objects. Single image reflection removal, on the other hand, uses selected image priors to obtain a good approximation of the target object. Although the use of multiple images somewhat mitigates the massive ill-posed problem created by the reflection removal formulation, these methods require multiple images from several viewpoints and the performance is strongly correlated with the quality of the acquired image pairs/sequences. In practice, acquisition conditions are non-optimal, which often results in image degradation, causing occlusions and blurring in the images. Those constrains make single image methods attract more attention in the community, which is more accessible to general user. However, the validity of assumptions required by conventional methods is also prone to be violated in real scenes.

1.2 Deep Learning Based Methods

In recent years, deep learning has achieved tremendous success in computer vision. It not only learns better perception of the high-level content of the image, but also help to enhance the image from low-level. There has been a significant progress in many low-level vision problems using deep learning based approaches, such as image super resolution (Dong et al., 2015; Ledig et al., 2017), image denoising (Mao, Shen, and Yang, 2016; Zhang et al., 2017b; Lehtinen et al., 2018), *etc*.

To apply deep neural networks to such problems, various frameworks and network architectures have been proposed. The basic components of convolutional networks is dated back to LeNet proposed by LeCun et al. (1998). The modern convolutional network usually consists of a stack of layers, *e.g.* convolution layer, pooling layer, activation layer, fully connected layer, *etc*, with skip connections helping to traverse information in the network.

For low-level vision tasks, the target of the network is usually to learn a mapping from pixel to pixel. Unlike networks designed for image classification or object detection, Fully convolutional network (FCN) (Long, Shelhamer, and Darrell, 2015) proposed for image segmentation transforms image pixels to pixel categories. FCN consists of a bunch of convolutional layers, with downsampling and upsampling inside the network. There is a lot of variation and development to the original FCN since then, and the commonality is the abandonment of fully connected layer. The absence of fully connected layer enables the network to accept variable input dimensions and learn a one-to-one correspondence between the input and predicted image in spatial dimension.

When designing the network, it is important to combine the high-level and low-level information and integrate multi-scale information through the network. Various loss functions are employed for the optimization process of the neural networks since minimizing the pixel-wise loss functions such as mean square error (MSE) between the estimated output and the ground-truth image usually is not sufficient for these ill-posed problems. There could be multiple solutions when using MSE loss, and the network will learn to average among those possible solutions in the pixel space, resulting in over-smoothed output (Ledig et al., 2017). Therefore it is essential to introduce intermediate or more advanced loss functions to limit the solution space. For example, perceptual loss are introduced to measure high-level perceptual and semantic differences between images (Johnson, Alahi, and Fei-Fei, 2016). Adversarial loss from Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) has been extensively employed in image-to-image transformation problems (Isola et al., 2017; Zhu et al., 2017).

Most approaches utilize supervised learning framework, which requires labelled training data. The quality and diversity of training data have significant impact on the generalization ability of the model in real-world scenarios. However, in many problems, the acquisition of real data with ground-truth is time and labour consuming or even impossible. One solution is to use synthetic data for training, which requires a good simulation of the real data. For some problems, the data is easier to generate, such as image denoising and image super resolution, while in other complex problems, the gap between synthetic data and real data would affect the generalization of model trained on synthetic data. Recently, self-supervised and unsupervised methods have been introduced to address the issue of lack of data in some tasks (Chen et al., 2018; Laine et al., 2019; Menon et al., 2020). There is potential to leverage the underlying supervision within data to train the network.

1.3 Motivations and Objectives

The overall objective of this thesis is to develop deep learning based methods for image deblurring and reflection removal problems. Although there are many successful applications of deep learning in other low-level vision problems, adopting relevant techniques for image deblurring and reflection removal is still challenging due to the complex nature of the problems. Conventional methods and previous works of learning-based methods have various shortcomings, including:

- Relying on manually defined priors and image features

  Conventional methods for both image deblurring and reflection removal rely on certain predefined image priors or regularizers to reduce the space for solving ill-posed problems. These priors or regularizers may work well on simple scenarios, *e.g.* where the motion blur is mild and image reflection has low intensity and is relative blurry compared to background. However, in real-world conditions, image blur and reflections can be strong and complex, and the assumptions may be invalid in those cases. Traditional learning-based methods relies on handcrafted image features which are not robust and usually computational expensive, thus limit the representation and prediction abilities.

- Lack of training data

  When applying deep learning to image deblurring and reflection removal problems, lack of training data with ground truth becomes a common issue. It is difficult to capture well-aligned image pairs for image deblurring and image triplets for reflection removal, and it is also impossible to capture ground-truth data with moving objects. Therefore, acquisition of sufficient real data for supervised training in these tasks is infeasible. Some researchers begin to simulate data to overcome this issue, however, the existing image generation model for those tasks may not reflect the nature of the real data, which results in poor generalization on real images.

- Lack of usage of global and mutual information

  Some previous work on image deblurring only work on small image patches, while ignoring global information. And some methods separate the learning process into different stages, or require additional steps. Global information and end-to-end learning is expected to provide more guidance to the problem. In terms of reflection removal, the frameworks proposed before mainly focus on recovering the background layer, while the correlation between background and reflection is neglected. The background and reflection contents, either from single image or multi-view images, are correlated, and the correlation can be utilized to provide additional information for the learning process.

Motivated by these limitations, we are devoted to develop deep learning based methods for image deblurring and reflection removal in the following perspectives:

- For image deblurring, we aim to develop a model which is able to handle spatial variant motion blur. We focus on the cause of the motion blur, *i.e.* the motion flow, and design an end-to-end framework for estimation the motion flow for recovery of the clear image.

- For reflection removal, we address the problem in two different ways. The first is through supervised learning and we address the lack of training data problem by using a general reflection generation model. And we design a framework to consider the correlation between background and reflection. The second is through self-supervised learning which seeks supervision from multi-view images with reflections and design framework and loss functions to utilize the geometry correspondence and consistency between those images.

1.4 Main Contributions

The main contributions of this thesis include a deep learning based method for heterogeneous motion blur removal and two frameworks for image reflection removal. We address the problem of lacking real training data by simulating the heterogeneous motion blur and image reflection, and also by proposing a self-supervised framework to learn to separate reflections using only unlabeled data.

Specifically,

- We propose an approach to estimate and remove pixel-wise heterogeneous motion blur by training on simulated examples. The blur model used is flexible and makes almost no assumptions about the underlying images, resulting in effectiveness on diverse data. We end-to-end estimate the dense heterogeneous motion flow from a single blurry image using a universal fully convolutional network. Beyond the previous patch-level learning, we directly perform training and testing on the whole image, which utilizes the spatial context over a wider area and estimates a dense motion flow map accurately. Moreover, our method does not require any post-processing and is computational faster than previous methods. The relevant work is described in Chapter 3.

- We propose to address the single image reflection removal by using a cascade deep neural network. The network we propose takes advantage of the correlation between the background and the reflection. Different from other methods that train a network to estimate background alone from the mixture image, our proposed method estimates not only background, but also the reflection. And we show that this can significantly improve the quality of reflection removal. The estimated background is used to guide the estimation of the reflection, then the estimated reflection is used to further improve the estimation of the background. The relevant work is described in Chapter 4.
- We propose a self-supervised method for single image reflection removal, which does not rely on the ground truth labels, but leverages the supervision from the geometry correspondence and consistency between the multi-view images contaminated by reflections. A series of novel consistency losses that are effective and robust are designed to utilize the imperfect cues derived from the multi-view consistency. The proposed method is the first self-supervised learning approach to handle the real-world reflection removal considering the multi-view setting. The relevant work is described in Chapter 5.

Chapter 2

Literature Review

Contents

This chapter presents a more detailed review on the publications in image deblurring and reflection removal. We first review conventional methods proposed before the era of deep learning. Then we review the basics of neural networks and the applications of deep learning on our targeted image processing tasks.

2.1 Conventional Methods

2.1.1 Image Deblurring

A natural image can become blurred for a variety of reasons, including defocusing, and optical aberrations, while the most common cause of blurred image is the motion of the camera or objects in the scene. Research into image deblurring has a long history. As mentioned in Section 1.1.1, different image priors and regularizers are introduced to constrain the solution space for blind image deblurring and various estimators are proposed to estimate the blur kernel.

Total variation (TV) proposed by Rudin, Osher, and Fatemi (1992) is a typical regularizer used in blind deconvolution (Chan and Wong, 1998; Perrone and Favaro, 2014). Total variation is intrinsically an ℓ_1 norm of the image gradients, thus induces sparsity over image gradients. Krishnan, Tay, and Fergus (2011) extend the ℓ_1 norm to a ℓ_1/ℓ_2-norm version, which deduces the blurry effect in the image without destroying the magnitude of the true gradient. Xu, Zheng, and Jia (2013) propose an approximation of the ℓ_0 norm as a sparsity prior. Pan et al. (2014) also use ℓ_0-regularized prior based on intensity and gradient for text image deblurring. Pan et al. (2016a) impose dark channel prior based on the observation that the dark channel of blurred images is less sparse. Fergus et al. (2006) assume that natural sharpe images obey a heavy-tailed distributions of image gradients and employ a Gaussian mixture model (GMM) to fit this distribution.

In bayesian inference framework, maximum-a-posteriori (MAP) is the most commonly used estimator and an auxiliary operation is usually employed to produce promising deblurring results. Cho and Lee (2009) incorporate the edge emphasizing operation into the iterative MAP. Gong et al. (2016) propose gradient activation based MAP which automatically selects a subset of gradients from the latent image for kernel estimation. Variational bayesian methods have also been applied to image deblurring (Levin et al., 2009; Levin et al., 2011; Zhang, Wipf, and Yanning Zhang, 2013).

The aforementioned methods are based on the assumption that the blur kernel is spatially invariant. However, in practice the blur kernel if often spatially variant due to complex motion or other factors. More flexible blur models are proposed to address spatially variant blur. Tai, Tan, and Brown (2011) develop the projective motion Richardson-Lucy (RL) algorithm to tackle the spatially variant case. A blurry

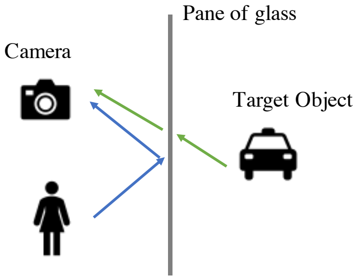


FIGURE 2.1. An illustration of the image formation in which a target object captured through a pane of glass will have reflection artifacts.

image is formulated as the weighted sum of a set of transformed sharp images. Whyte et al. (2012) propose a new parameterized geometric model of the blurring process in terms of the rotational velocity of the camera during exposure. Zhang and Wipf (2013) introduce a non-uniform blind deblurring algorithm with a spatially-adaptive image penalty. Gupta et al. (2010) model the camera motion as a motion density function which can be used to generate the kernel at any location in the image without knowing the temporal ordering of the motion curve. Hirsch et al. (2010) and Hirsch et al. (2011) propose to reduce computational cost by locally uniform overlapping-patch-based models. Levin (2006), Dai and Wu (2009), Hyun Kim, Ahn, and Mu Lee (2013), and Pan et al. (2016b) segment image into layers with different blurs to deblur different motions, but accurate segmentation of a blurred image is required for these methods. However, in any case, the correction of a spatially variant blur is a highly ill-posed problem with many unknowns, therefore it is difficult to recover a sharp image without artifacts.

2.1.2 Reflection Removal

The cause of reflection is shown in Fig 2.1, the presence of panes of glass cause the reflection of the objects in front of the glass to overlap with the target object behind the glass. As introduced in in Section 1.1.2, reflection removal is also a highly ill-posed problem. Solving this ill-posed problem requires either very effective image priors, or auxiliary data such as multiple images captured with motion or polarizers, or user input.

Some methods rely on special image pairs as input. Agrawal et al. (2005) propose to use a flash and no-flash image pair to remove the strong highlight reflection caused by flash, which a very special case of reflection. They capture the properties of image gradients that is invariant between a flash and an ambient image, which are then used to remove the component of image gradients introduced by undesirable reflections. Schechner, Kiryati, and Basri (2000) propose to use focus/defocus image pair to remove reflections. It is based on a method for self calibration of the defocus blur kernels and minimizing the mutual information of the recovered layers. Polarization is also widely used to separate reflections (Schechner, Shamir, and Kiryati, 2000; Sarel and Irani, 2004; Kong, Tai, and Shin, 2014). Rotating a polarizer with different angles yields different level of reflections. These methods are limited by the special skills and devices required when capturing the input data.

Some methods use image sequences as the input. Gai, Shi, and Zhang (2012) assume that the motion of each layer follows an affine transformation and an image prior based on joint relationship of both background and reflection gradient is employed to separate reflections. Li and Brown (2013) use SIFT-flow to calculate the motion field and align images. Guo, Cao, and Ma (2014) assume the targeting background region lies on a planar surface in the scene, and there exists a homography transformation to align the regions. A rank minimization method is applied to solve the decomposition problem robustly.

A more practical solution is to remove reflections from a single image. Levin and Weiss (2007) impose a gradient sparsity prior on the reflection layers, but it requires user assistance to label a small number of gradients as belonging to one of the layers. Li and Brown (2014) propose to model relative smoothness of the two layers by building two likelihoods for each layer from gradient histograms. Arvanitopoulos, Achanta, and Süsstrunk (2017) impose a ℓ_0 gradient sparsity prior to eliminate a substantial amount of gradients of small magnitudes while retaining large magnitude edges and a Laplacian data fidelity term to better enforces consistency in structures of fine details. Shih et al. (2015) exploit a special form of reflections, where the glass is double-pane and there exists shifted and attenuated double reflections. The ghosting effect breaks symmetry between background and reflection, thus provides an effective cue to separate the two layers.

2.1.3 Limitations of Conventional Methods

To sum up, although conventional methods achieve some progress in image deblurring and reflection removal, the limitations are obvious and inevitable.

- Relies on manually defined priors or regularizers. Conventional methods for both image deblurring and reflection removal rely on additional explicitly defined prior or regularizers characterized by restrictive assumptions that can easily be broken in complex real-world scenarios, resulting in poor generalization in such scenes.
- Computational expensive. The optimization process of conventional methods are usually computational expensive and time consuming, thus is impractical for real-time applications.

The limited performance and high cost in processing make conventional methods far from satisfactory in these problems. In order to overcome these shortcoming, deep learning techniques, specifically CNNs, are introduced to the field of image blurring and reflection removal.

2.2 Convolutional Neural Networks

Neural networks have been studied for decades. Initially inspired by human brains, neural networks are make up of neurons that have learnable weights and biases. They receive an input and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. Training of neural networks is to solve a nonlinear optimization problem within a maximum likelihood framework using back-propagation technique.

LeCun et al. (1998) first propose a CNN named LeNet to recognize handwritten digits. Different from regular neural networks, it use convolution layers as hidden layer instead of fully-connected layers. Although LeNet is dated back to 1990s, the modern CNNs still use similar components.

2.2.1 Components of CNNs

The main types of layers to build CNN architectures include convolution layer, pooling layer, fully connected layer and some other auxiliary layers.

Convolution Layer. The convolution layer is the core building block of a CNN that does most of the learning computations. It computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. The filter is small spatially, usually 3×3, but extends through the full depth of the input volume. Since the convolution layer has sparse and local connectivity and parameter sharing scheme, the amount of parameters are significantly reduced.

Pooling Layer. Pooling layer is commonly used to downsample the feature maps and is inserted in-between successive convolution layers in a CNN architecture. It progressively reduces the spatial size of the representation to reduce the amount of parameters and computation in the network, which can also control overfitting. Recently, discarding pooling layers has been found important in training good generative models, such as generative adversarial networks (GANs). Convolution layers with stride greater than 1 is used to downsample the feature map in this case instead.

Fully-connected Layer. Fully-connected layer used in CNNs is the same as in regular neural networks. For tasks such as image classification, fully-connected layer is usually used as the final layer to map the features to the predicted label. However, for tasks that require image-to-image transformation, using fully convolution layers is a more sensible way.

Activation Layer. Activation functions are applied to the output of convolution layers or fully-connected layers to add nonlinearity to the network. ReLU (Nair and Hinton, 2010) is the most widely used nonlinearity for CNNs. There are many other activation functions as well, and may have better performance depending on the applications. For example, in generative adversarial networks, Leaky ReLU (Maas, Hannun, and Ng, 2013) is more often used for better convergence.

Other Layers. There are many other layers included, such as normalization layers. Batch normalization (Ioffe and Szegedy, 2015), which normalizes activations in a network across the mini-batch of a definite size, is able to improve the training of many models. Instance normalization (Ulyanov, Vedaldi, and Lempitsky, 2016), on the other hand, normalizes across each channel in each training sample.

2.2.2 Architectures

Since LeNet, various CNN architectures have been proposed for better performance and adaptation to different problems.

AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) is the first popularized CNNs in computer vision. By using a deep convolutional layers stacked network, it significantly improves the performance on ImageNet ILSVRC challenge (Russakovsky et al., 2015).

VGGNet (Simonyan and Zisserman, 2015) shows that the depth of the network is a critical component for good performance. It adopts an extremely homogeneous architecture that only performs 3×3 convolutions and 2×2 pooling throughout the network. Although VGGNet is designed for image classification, it can be transferred to other tasks utilizing semantic information through a pre-trained network.

ResNet (He et al., 2016) features special skip connections and a heavy use of batch normalization. By employing the residual blocks, it extends the depth of CNNs to a new level and since then the residual block has been adopted as a building block in many other architectures.

FCN (Long, Shelhamer, and Darrell, 2015) was initially designed for semantic segmentation problems. By using a network with fully convolution layer, it takes input of arbitrary size and produce output of the same or arbitrary size. The subsequent networks for semantic segmentation and problems aiming at image-to-image transformation are all based on FCN style architecture.

U-net (Ronneberger, Fischer, and Brox, 2015) is first introduced for medical segmentation. It is found that it also get very good performance on low-level vision problems. The U-net architecture is very good at incorporating low-level information with high-level information by including cascaded skip connections. Therefore, it achieves promising performance on many computational photography problems, which is mainly focus on low-level content while benefits from high-level information.

2.3 Deep Learning Based Methods

Since the rising of deep learning, various learning-based approaches have been proposed for computational photography problems. This section presents some deep learning based methods for image deblurring and reflection removal.

2.3.1 Image Deblurring

Learning based methods have been proposed for both non-blind and blind deconvolution. Figure 2.2 illustrates the taxonomy of existing methods of image deblurring based on deep learning.

For non-blind deblurring, a neural network can be used as a deconvolution network, or as a denoising network to remove the noise from a restored sharp image. Deconvolution is a critical process in recovering a sharp image. Xu et al. (2014) propose a deconvolution CNN (DCNN), which approximates Wiener deconvolution. In

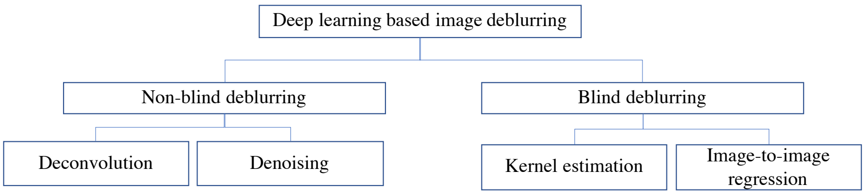


FIGURE 2.2. Taxonomy of existing methods of image deblurring based on deep learning

regularized Wiener deconvolution, a pseudo-inverse kernel can be obtained by a simple operation in the Fourier domain. However, DCNN has a drawback that it needs to be trained for each blur kernel. Ren et al. (2018) improve DCNN by exploiting the low-rank property of the inverse kernels to obtain a unified kernel decomposition which can handle a large number of blur kernels. Deep Image Prior (DIP) (Ulyanov, Vedaldi, and Lempitsky, 2018) combines the iterative scheme of prior-based optimization with the prior modeling ability of neural network. A neural network learns low-level statistics of clean images and can then generate a clean image, because the network is forced to create an image that satisfies the learned prior.

In early applications of neural networks to denoising, a regularized inverse filter is applied to the blurred image first, and the deconvolved image is then sent to a denoising network (Schuler et al., 2013), which attenuates the outliers produced by deconvolution. In later work, the denoising network is embedded in a MAP-based optimization framework (Zhang et al., 2017a; Zhang et al., 2017c). The restored image from a standard deconvolution module is fed into a denoising network, which refines either gradients (Zhang et al., 2017a) or image priors (Zhang et al., 2017c), to improve the rejection of outliers, rather than remove noise directly.

Blind image deblurring either use neural network to estimate blur kernel or perform end-to-end image restoration directly. Sun et al. (2015) propose patch-wise motion vector classification to estimate motion flow from single blurry image and use Markov Random Field (MRF) to refine the dense motion field. The network is trained with uniform motion blur on small patch.

Some work perform kernel estimation in the frequency domain. Schuler et al.

(2015) propose a combination of learning-based feature extraction and kernel estimation specific to image deconvolution. Chakrabarti (2016) propose to use a neural network to infer the Fourier coefficients of the inverse filter from a blurred input image.

Kernel estimation has some drawbacks. Firstly, a simple CNN cannot estimate complex motion kernels. Besides, it is difficult to design a generalized kernel estimation system that will deal with different types of blur. Lastly, this type of method requires deconvolution as an additional process. Nah, Hyun Kim, and Mu Lee, 2017 introduce a multi-scale CNN trained in an end-to-end manner on a large GoPro dataset. They directly restore a latent sharp image without kernel estimation. Tao et al., 2018 design CNNs for different scales to share learnable parameters.

Kupyn et al. (2018) introduce a single-scale training method based on Generative adversarial network (GAN) (Goodfellow et al., 2014). The learning is based on a conditional GAN and the content loss. Zhang et al. (2018) introduce a hybrid network which employs a recurrent neural network (RNN) as well as a CNN. They treat blurring as a process of diffusing the information encoded in sharp edges across an image. To restore the diffused information to its original dense form requires a receptive field covering a large area, thus the deblurring network should be large enough to account for long-range spatial dependencies.

2.3.2 Reflection Removal

Fan et al. (2017) first propose to employ a CNN named CEILNet to remove image reflections. The proposed framework address the problem into a two stage process. An edge map corresponding to the background image is first estimated using one network, and then the other network use the estimated edge map alone with the input image to recover the background image. It is based on the assumption that the predominant edges in the observed image come from the background content.

Based on CEILNet, Wan et al. (2018) propose a framework which combines the edge/gradient inference with the image inference into a unified mechanism. The network also use multi-scaled structure to better preserve the background details. Apart from pixel-wise content loss, they also use a perceptual motivated SSIM (Wang et al., 2004) loss to measure the similarity between the estimated and ground truth images.

Zhang, Ng, and Chen (2018) exploit to leverage high-level semantic information to assist the reflection removal. It employed VGG-19 network to extract features from image and employed a feature loss from VGG-19 network which evaluates the difference between predicted background and ground-truth background in feature space. They also designed an exclusion loss to encourage the background and reflection to contain different edges.

For learning-based reflection removal approaches, the source of training data is a challenging problem. A common practice is to synthesize reflections from clear images. Fan et al. (2017) propose a heuristic approach to simulate reflections considering strong reflections and Zhang, Ng, and Chen (2018) improve the data synthesis method by removing gamma correction and using more flexible reflection decay and blur parameters. Ma et al. (2019) propose to use a neural network to generate reflections instead of manual process. They incorporate the reflection generation and reflection removal into a single network. Wen et al. (2019) propose to synthesize reflection images by predicting a non-linear alpha blending mask instead of synthesizing reflection with a fixed combination factor or kernel. Kim, Huo, and Yoon (2020) utilize physically based rendering in reflection image synthesis. They use existing RGBD/RGB images to estimate meshes, then physically simulate the light transportation between meshes, glass, and lens with path tracing to synthesize training data, which successfully reproduce the spatially variant anisotropic visual effect of glass reflection. To improve the generalization ability on real data, recent methods add a portion of real images to the training data. For example, Zhang, Ng, and Chen (2018) collect a small amount of real data with ground truth using the data collecting method introduced by Wan et al. (2017). To relieve the restriction that the image triplets collected should be well-aligned, Wei et al. (2019) accept misaligned training data by introducing an alignment-invariant loss function.

2.4 Summary

In this chapter, we have reviewed the literatures regarding image deblurring, reflction removal and the the era of deep learning. Various assumptions are made in conventional methods, and limitations of these assumptions restrict the performance of conventional methods, thus learning-based methods are introduced to address these limitations. We have reviewed the fundamental components of convolutional neural networks and recent progress in image deblurring and reflection removal. In the following chapters, we will introduce our proposed methods addressing those issues.

Chapter 3

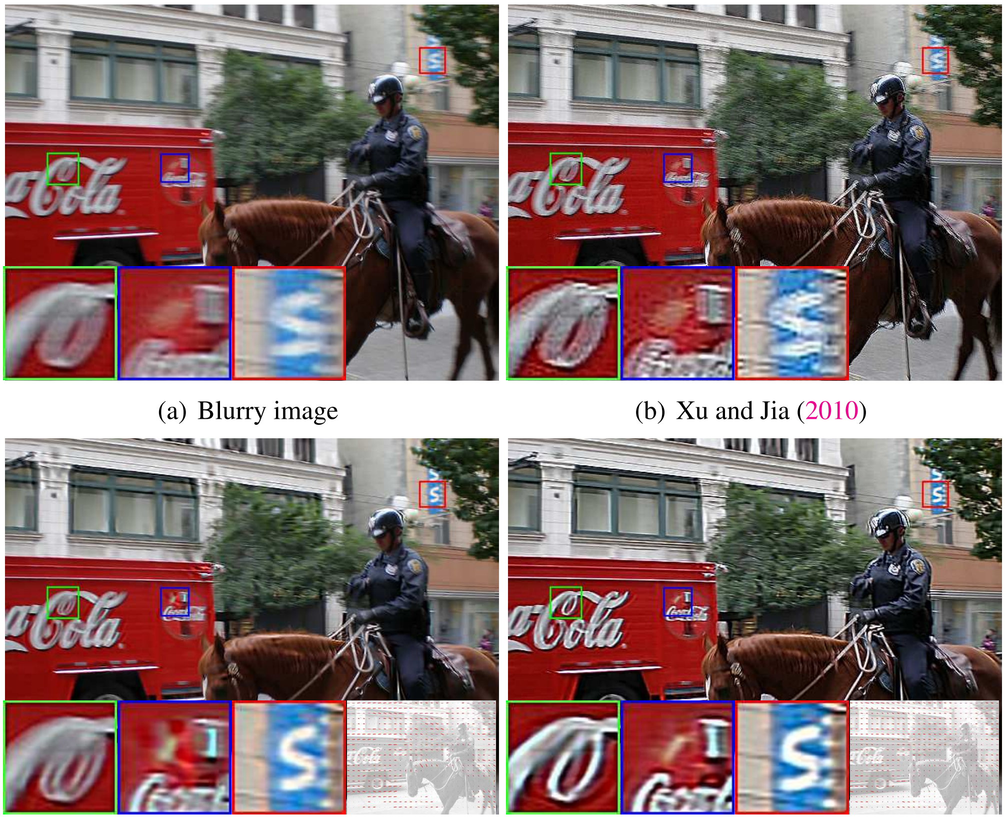
Deep Learning for Removing Heterogeneous Motion Blur

Contents

We introduce our method for image motion blur removal in this chapter. Removing pixel-wise heterogeneous motion blur is challenging due to the ill-posed nature of the problem. The predominant solution is to estimate the blur kernel by adding a prior, but the extensive literature on the subject indicates the difficulty in identifying a prior which is suitably informative, and general. Rather than imposing a *prior based on theory, we propose instead to learn one from the data. Learning a prior over the latent image would require modeling all possible image content. The critical observation underpinning our approach is thus that learning the motion flow instead allows the model to focus on the cause of the blur, irrespective of the image content. This is a much easier learning task, but it also avoids the iterative process through which latent image priors are typically applied. Our approach directly estimates the motion flow from the blurred image through a fully-convolutional deep neural network (FCN) and recovers the unblurred image from the estimated motion flow. Our FCN is the first universal end-to-end mapping from the blurred image to the dense motion flow. To train the FCN, we simulate motion flows to generate synthetic blurred-image-motion-flow pairs thus avoiding the need for human labeling. Extensive experiments on challenging realistic blurred images demonstrate that the proposed method outperforms the state-of-the-art. This work was presented at CVPR 2017.*

3.1 Introduction

In this chapter, we aim at removing heterogeneous motion blur using deep learning based method. Motion blur is ubiquitous in photography, especially when using light-weight mobile devices, such as cell-phones and on-board cameras. While there has been a significant progress on image deblurring (Fergus et al., 2006; Cho and Lee, 2009; Xu and Jia, 2010; Pan et al., 2014; Pan et al., 2016a; Gong et al., 2016), most work focuses on *spatially-uniform* blur. Some recent methods (Whyte et al., 2012; Hirsch et al., 2011; Hu, Xu, and Yang, 2014; Kim and Lee, 2014; Pan et al., 2016b) have been proposed to remove *spatially-varying* blur caused by camera panning, and/or object movement, with some restrictive assumptions on the types of blur, image prior, or both. In this work, we focus on recovering a blur-free latent image from a single observation degraded by *heterogeneous motion blur*, *i.e.* the blur kernels may independently vary from pixel to pixel.



(a) Blurry image

(b) Xu and Jia (2010)

(c) Sun et al. (2015)

(d) Ours

FIGURE 3.1. A blurry image with heterogeneous motion blur from a widely used dataset Microsoft COCO (Lin et al., 2014). Estimated motion flows are shown in the bottom right corner of each image.

Motion blur in real images has a variety of causes, including camera (Whyte et al., 2012; Zheng, Xu, and Jia, 2013) and object motion (Hyun Kim, Ahn, and Mu Lee, 2013; Pan et al., 2016b), leading to blur patterns with complex variations (See Figure 3.1 (a)). In practice, uniform deblurring methods (Fergus et al., 2006; Cho and Lee, 2009; Xu and Jia, 2010) usually fail to remove the non-uniform blur (See Figure 3.1 (b)). Most existing non-uniform deblurring methods rely on a specific motion model, such as 3D camera motion modeling (Gupta et al., 2010; Whyte et al., 2012) and segment-wise motion (Levin, 2006; Pan et al., 2016b). Although a recent method proposed by Kim and Lee (2014) uses a flexible motion flow map to handle heterogeneous motion blur, it requires a time-consuming iterative estimator. In addition to the assumptions about the cause of blur, most existing deblurring methods also rely on predefined priors or manually designed image features. Most conventional methods (*e.g.* Fergus et al., 2006; Levin et al., 2011; Xu, Zheng, and

Jia, 2013) need to iteratively update the intermediate image and the blur kernel with using these predefined image priors to reduce the ill-posedness. However, solving these non-convex problems is non-trivial, and many real images do not conform to the assumptions behind a particular model. Recently, learning-based discriminative methods (Chakrabarti, Zickler, and Freeman, 2010; Couzinie-Devy et al., 2013) have been proposed to learn blur image patterns and avoid the heavy computational cost of blur estimation. However, their representation and prediction abilities are limited by their manually designed features and simple mapping functions. Although a deep learning based method (Sun et al., 2015) aimed to overcome these problems, it restrictively conducts the learning process at the patch-level and thus cannot take full advantage of the context information from larger image regions.

In summary, there are three main problems with existing approaches: 1) the range of applicable motion types is limited, 2) manually defined priors and image features may not reflect the nature of the data and 3) complicated and time-consuming optimization and/or post-processing is required. Generally, these problems limit the practical applicability of blur removal methods to real images, as they tend to cause worse artifacts than they cure.

To handle general heterogeneous motion blur, based on the motion flow model, we propose a deep neural network based method able to directly estimate a pixel-wise motion flow map from a single blurred image by learning from tens of thousands of examples. To summarize, the main contributions of this chapter are:

- We propose an approach to estimate and remove pixel-wise heterogeneous motion blur by training on simulated examples. Our method uses a flexible blur model and makes almost no assumptions about the underlying images, resulting in effectiveness on diverse data.

- We introduce a universal FCN for end-to-end estimation of dense heterogeneous motion flow from a single blurry image. Beyond the previous patch-level learning (Sun et al., 2015), we directly perform training and testing on the whole image, which utilizes the spatial context over a wider area and estimates a dense motion flow map accurately. Moreover, our method does not require any post-processing.

3.2 Related Work

Conventional blind image deblurring To constrain the solution space for blind deblurring, a common assumption is that image blur is spatially uniform (Chan and Wong, 1998; Cho and Lee, 2009; Fergus et al., 2006; Levin et al., 2011; Pan et al., 2016a; Gong et al., 2016). Meanwhile, numerous image priors or regularizers have been studied to overcome the ill-posed nature of the problem, such as the total variational regularizer (Chan and Wong, 1998; Perrone and Favaro, 2014), Gaussian scale mixture priors (Fergus et al., 2006) and ℓ_1/ℓ_2-norms (Krishnan, Tay, and Fergus, 2011), ℓ_0-norms (Xu, Zheng, and Jia, 2013; Pan et al., 2014), and dark channel (Pan et al., 2016a) based regularizers.

Moreover, various estimators have been proposed for more robust kernel estimation, such as edge-extraction-based maximum-a-posteriori (MAP) (Cho and Lee, 2009; Sun et al., 2013), gradient activation based MAP (Gong et al., 2016), variational Bayesian methods (Levin et al., 2009; Levin et al., 2011; Zhang, Wipf, and Yanning Zhang, 2013), *etc*. Although these powerful priors and estimators work well on many benchmark datasets, they are often characterised by restrictive assumptions that limit their practical applicability.

Spatially-varying blur removal To handle spatially-varying blur, more flexible blur models are proposed. A projective motion path model proposed by Tai, Tan, and Brown (2011) formulates a blurry image as the weighted sum of a set of transformed sharp images, an approach which is simplified and extended by Whyte et al. (2012) and Zhang and Wipf (2013). Gupta et al. (2010) model the camera motion as a motion density function for non-uniform deblurring. Several locally uniform overlapping-patch-based models (Hirsch et al., 2010; Hirsch et al., 2011) are proposed to reduce the computational burden. Zheng, Xu, and Jia (2013) specifically modelled the blur caused by forward camera motion. To handle blur caused by object motion, some methods (Levin, 2006; Dai and Wu, 2009; Hyun Kim, Ahn, and Mu Lee, 2013; Pan et al., 2016b) segment images into areas with different types of blur, and are thus heavily dependent on an accruate segmentation of a blurred image. Recently, a pixel-wise linear motion model (Kim and Lee, 2014) is proposed to handle heterogeneous motion blur. Although the motion is assumed to be locally linear, there is no assumption on the latent motion, making it flexible enough to handle an extensive range of possible motion.

Learning based motion blur removing Recently, learning based methods have been used to achieve more flexible and efficient blur removal. Some discriminative methods are proposed for non-blind deconvolution based on Gaussian CRF (Schmidt et al., 2013), multi-layer perceptron (MLP) (Schuler et al., 2013), and deep convolution neural network (CNN) (Xu et al., 2014), etc, which all require the known blur kernels. Some end-to-end methods (Kim, Lee, and Lee, 2016; Mao, Shen, and Yang, 2016) are proposed to reconstruct blur-free images, however, they can only handle mild Gaussian blur. Recently, Wieschollek et al. (2016) introduce an MLP based blind deblurring method by using information in multiple images with small variations. Chakrabarti (2016) trains a patch-based neural network to estimate the frequency information for uniform motion blur removal. The most relevant work is a method based on CNN and patch-level blur type classification (Sun et al., 2015), which also focuses on estimating the motion flow from single blurry image. The authors train a CNN on small patch examples with *uniform* motion blur, where each patch is assigned a single motion label, violating the real data nature and ignoring the correspondence in larger areas. Many post-processing such as MRF are required for the final dense motion flow.

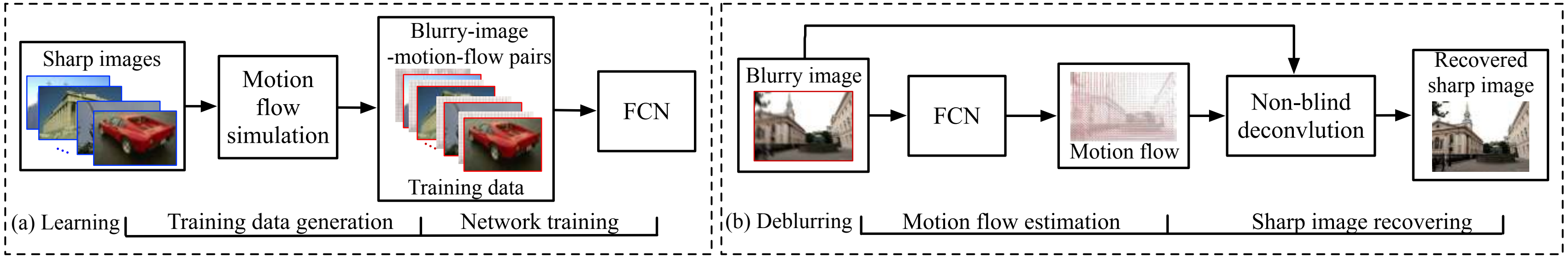


FIGURE 3.2. Overview of our scheme for heterogeneous motion blur removal. (a) We train an FCN using examples based on simulated motion flow maps. (b) Given a blurry image, we perform end-to-end motion flow estimation using the trained FCN, and then recover the sharp image via non-blind deconvolution.

3.3 Estimating Motion Flow for Blur Removal

3.3.1 A Heterogeneous Motion Blur Model

Letting $*$ denote a general convolution operator, a $P \times Q$ blurred image $\mathbf{Y}$ can be modeled as

$$\mathbf{Y} = \mathcal{K} * \mathbf{X} + \mathbf{N}, \tag{3.1}$$

where $\mathbf{X}$ denotes the latent sharp image, $\mathbf{N}$ refers to additive noise, and $\mathcal{K}$ denotes a heterogeneous motion blur kernel map with different blur kernels for each pixel in $\mathbf{X}$. Let $\mathcal{K}_{(i,j)}$ represent the kernel from $\mathcal{K}$ that operates on a region of the image centered at pixel (i, j). Thus, at each pixel of $\mathbf{Y}$, we have

$$\mathbf{Y}(i,j) = \sum_{i',j'} \mathcal{K}_{(i,j)}(i',j')\mathbf{X}(i+i', j+j'). \tag{3.2}$$

If we define an operator $\text{vec}(\cdot)$ which vectorises a matrix and let $\mathbf{y} = \text{vec}(\mathbf{Y})$, $\mathbf{x} = \text{vec}(\mathbf{X})$ and $\mathbf{n} = \text{vec}(\mathbf{N})$ then (3.1) can also be represented as

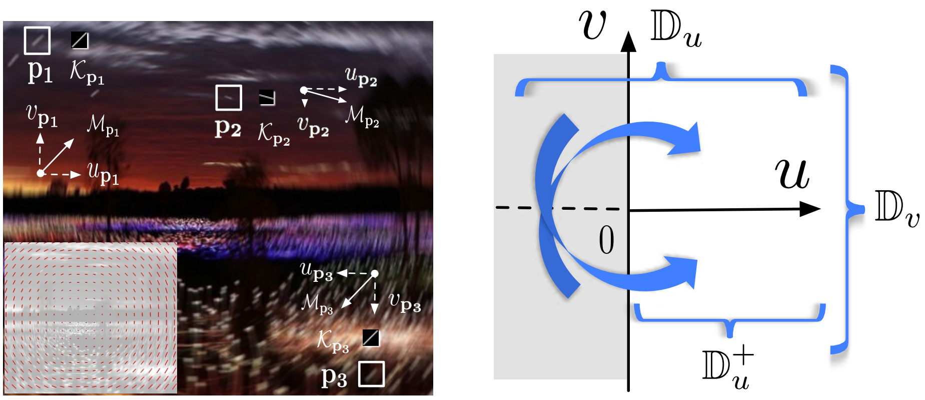
$$\mathbf{y} = \mathbf{H}(\mathcal{K})\mathbf{x} + \mathbf{n}, \tag{3.3}$$

where $\mathbf{H}(\mathcal{K}) \in \mathbb{R}^{PQ \times PQ}$[1]and each row corresponds to a blur kernel located at each pixel (*i.e.* $\mathcal{K}_{(i,j)}$).

3.3.2 Blur Removal via Motion Flow Estimation

Given a blurry image $\mathbf{Y}$, our goal is to estimate the blur kernel $\mathcal{K}$ and recover a blur-free latent image $\mathbf{X}$ through non-blind deconvolution that can be performed by solving a convex problem (Figure 3.2 (b)). As mentioned above, kernel estimation is the most difficult and crucial part.

[1] For simplicity, we assume $\mathbf{X}$ and $\mathbf{Y}$ have the same size.



(a) Motion blur and motion flow

(b) Domain of motion

FIGURE 3.3. Motion blur and motion vector. (a) An example with blur cause by clock-wise rotation. Three examples of the blur pattern, linear blur kernel and motion vector are shown. The blur kernels on $\mathbf{p}_1$ and $\mathbf{p}_3$ caused by motions with opposite directions and have the same appearance. (b) Illustrations of the feasible domain of motion flow.

Based on the model in (3.1) and (3.2), heterogeneous motion blur can be modeled by a set of blur kernels, one associated with each pixel and its motion. By using a linear motion model to indicate each pixel's motion during imaging process (Kim and Lee, 2014), and letting $\mathbf{p} = (i, j)$ denote a pixel location, the motion at pixel $\mathbf{p}$, can be represented by a 2-dimensional *motion vector* $(u_\mathbf{p}, v_\mathbf{p})$, where $u_\mathbf{p}$ and $v_\mathbf{p}$ represent the movement in the horizontal and vertical directions, respectively (See Figure 3.3 (a)). By a slight abuse of notation we express this as $\mathcal{M}_\mathbf{p} = (u_\mathbf{p}, v_\mathbf{p})$, which characterizes the movement at pixel $\mathbf{p}$ over the exposure time. If we have the feasible domain $u_\mathbf{p} \in \mathbb{D}_u$ and $v_\mathbf{p} \in \mathbb{D}_v$, then $\mathcal{M}_\mathbf{p} \in \mathbb{D}_u \times \mathbb{D}_v$, but will be introduced in detail later. As shown in Figure 3.3, the blur kernel on each pixel appears as a line trace with nonzero components only along the motion trace. As a result, the motion blur $\mathcal{K}_\mathbf{p}$ in (3.2) can be expressed as (Brusius, Schwanecke, and Barth, 2011):

$$\mathcal{K}_\mathbf{p}(i', j') = \begin{cases} 0, & \text{if } \|(i', j')\|_2 \geq \frac{\|\mathcal{M}_\mathbf{p}\|_2}{2}, \\ \frac{1}{\|\mathcal{M}_\mathbf{p}\|_2} \delta(v_\mathbf{p} i' - u_\mathbf{p} j'), & \text{otherwise}, \end{cases} \tag{3.4}$$

where $\delta(\cdot)$ denotes the Dirac delta function. We thus can achieve heterogeneous motion blur estimation by estimating the motion vectors on all pixels, the result of

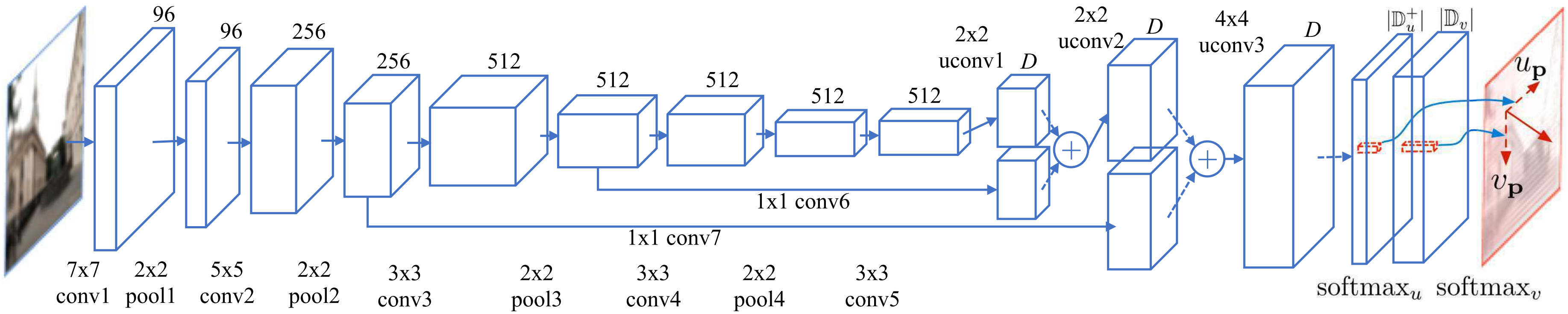


FIGURE 3.4. Our network structure. A blurred image goes through layers and produces a pixel-wise dense motion flow map. conv means a convolutional layer and uconv means a fractionally-strided convolutional (deconvolutional) layer, where $n \times n$ for each uconv layer denotes that the up-sampling size is n. Skip connections on top of pool2 and pool3 are used to combine features with different resolutions.

which is $\mathcal{M}$, which is referred as *motion flow*. For convenience of expression, we let $\mathcal{M} = (\mathbf{U}, \mathbf{V})$, where $\mathbf{U}$ and $\mathbf{V}$ denote the motion maps in the horizontal and vertical directions, respectively. For any pixel $\mathbf{p} = (i, j)$, we define $\mathcal{M}_\mathbf{p} = (\mathbf{U}(i, j), \mathbf{V}(i, j))$ with $\mathbf{U}(i, j) = u_\mathbf{p}$ and $\mathbf{V}(i, j) = v_\mathbf{p}$.

As shown in Figure 3.2 (b), given a blurred image and the estimated motion flow, we can recover the sharp image by solving an non-blind deconvolution problem

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}(\mathcal{K})\mathbf{x}\|_2^2 + \Omega(\mathbf{x})$$

with regularizer $\Omega(\mathbf{x})$ on the unknown sharp image. In practice, we use a Gaussian mixture model based regularizer as $\Omega(\mathbf{x})$ (Zoran and Weiss, 2011; Sun et al., 2015).

3.3.3 Learning for Motion Flow Estimation

The key contribution of our work is to show how to obtain the motion flow field that results in the pixel-wise motion blur. To do so we train a FCN to directly estimate the motion flow field from the blurry image.

Let $\{(\mathbf{Y}^t, \mathcal{M}^t)\}_{t=1}^T$ be a set of blurred-image and motion-flow-map pairs, which we take as our training set. Our task is to learn an end-to-end mapping function $\mathcal{M} = f(\mathbf{Y})$ from any observed blurry image $\mathbf{Y}$ to the underlying motion flow $\mathcal{M}$. In practice, the challenge is that obtaining the training ground-truth dense motion flow for sufficiently many and varied real blurry images is infeasible. Human labeling is impossible, and training from automated methods for image deblurring would defeat the purpose. To overcome this problem, we generate the training set by simulating motion flows maps. (See section 3.4.2). Specifically, we collect a set of sharp images $\{\mathbf{X}^n\}$, simulate T motion flows $\{\mathcal{M}^t\}$ in total for all images in $\{\mathbf{X}^n\}$, and then generate T blurred images $\{\mathbf{Y}^t\}$ based on the models in (3.1) and (3.4) (See Figure 3.2 (a)).

Feasible domain of motion flow To simplify the training process, we train the FCN over a discrete output domain. Interestingly, classification on discrete output space has achieved some impressive results for some similar applications, *e.g.* optical flow estimation Walker, Gupta, and Hebert (2015) and surface normal prediction Wang, Fouhey, and Gupta (2015). In our work, we adopt an integer domain for both $\mathbf{U}$ and $\mathbf{V}$, and treat the mapping $\mathcal{M} = f(\mathbf{Y})$ as a multi-class classification problem. Specifically, we uniformly discretize the motion values as integers with a 1 (pixel) interval, which provides a high-precision approximation to the latent continuous space. As a result, by assuming the maximum movements in the horizontal and vertical directions to be u_{max} and v_{max}, respectively, we have $\mathbb{D}_u = \{u|u \in \mathbb{Z}, |u| \leq u_{max}\}$ and $\mathbb{D}_v = \{v|v \in \mathbb{Z}, |v| \leq v_{max}\}$, where $\mathbb{Z}$ denotes the integer domain.

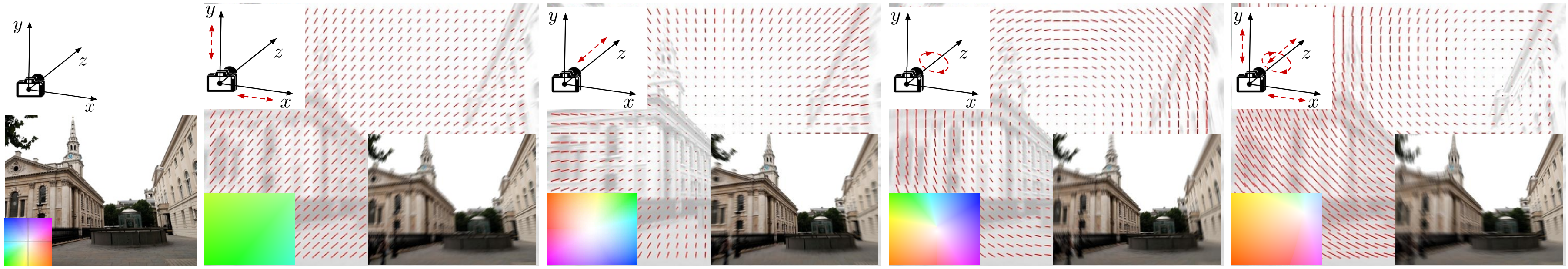
As shown in Figure 3.3 (a), any linear blur kernel is symmetric. Any two motion vectors with same length and opposite directions, *e.g.* $(u_\mathbf{p}, v_\mathbf{p})$ and $(-u_\mathbf{p}, -v_\mathbf{p})$, generate the same blur pattern, which may confuse the learning process. We thus further restrict the motion in the horizontal direction to be nonnegative as shown in Figure 3.3 (b), *i.e.* $u_\mathbf{p} \in \mathbb{D}_u^+ = \{u|u \in \mathbb{Z}_0^+, |u| \leq u_{max}\}$, by letting $(u_\mathbf{p}, v_\mathbf{p}) = \phi(u_\mathbf{p}, v_\mathbf{p})$ where

$$\phi(u_\mathbf{p}, v_\mathbf{p}) = \begin{cases} (-u_\mathbf{p}, -v_\mathbf{p}), & \text{if } u_\mathbf{p} < 0, \\ (u_\mathbf{p}, v_\mathbf{p}), & \text{otherwise.} \end{cases} \tag{3.5}$$

3.4 Dense Motion Flow Estimation

3.4.1 Network Design

The goal of this FCN network is to achieve the end-to-end mapping from a blurry image to its corresponding motion flow map. Given any RGB image with the arbitrary size $P \times Q$, the FCN is used to estimate a motion flow map $\mathcal{M} = (\mathbf{U}, \mathbf{V})$ with the same size to the input image, where $\mathbf{U}(i,j) \in \mathbb{D}_u^+$ and $\mathbf{V}(i,j) \in \mathbb{D}_v$, $\forall i, j$. For convenience, we let $D = |\mathbb{D}_u^+| + |\mathbb{D}_v|$ denote the total number of labels for both $\mathbf{U}$ and $\mathbf{V}$. Our network structure is similar to the original FCN (Long, Shelhamer, and Darrell, 2015). As shown in Figure 5.2, we use 7 convolutional (conv)



(a) Sharp Image (b) x and y-axis translation (c) z-axis translation (d) z-axis rotation (e) Arbitrary sampled motion

FIGURE 3.5. Demonstration of the motion flow simulation. (a) A sharp example image and the coordinate system of camera. (b)-(c) The sampled motion flow and the corresponding blurred image by simulating the translation along x and y-axes ($\mathcal{M}_{T_x} + \mathcal{M}_{T_y}$), translation along z-axis ($\mathcal{M}_{T_z}$) and rotation around z-axis ($\mathcal{M}_{R_z}$), respectively. (d) A sample based on the model considering all components in (3.6).

layers and 4 max-pooling (pool) layers as well as 3 uconv layers to up-sample the prediction maps. Following Wang and Gupta (2016), uconv denotes the fractionally-strided convolution, a.k.a. deconvolution. We use a small stride of 1 pixel for all convolutional layers. The uconv layers are initialized with bilinear interpolation and used to up-sample the activations. We also add skip connections which combine the information from different layers as shown in Figure 5.2.

The feature map of the last uconv layer (conv7 + uconv2) is a $P \times Q \times D$ tensor with the top $|\mathbb{D}_u^+|$ slices of feature maps ($P \times Q \times |\mathbb{D}_u^+|$) corresponding to the estimation of $\mathbf{U}$, and the remaining $|\mathbb{D}_v|$ slices of feature maps ($P \times Q \times |\mathbb{D}_v|$) corresponding to the estimation of $\mathbf{V}$. Two separate soft-max layers are applied to those two parts respectively to obtain the posterior probability estimation of both channels. Let $F_{u,i,j}(\mathbf{Y})$ represent the probability that the pixel at (i,j) having a movement u along the horizontal direction, and $F_{v,i,j}(\mathbf{Y})$ represent the probability that the pixel at (i,j) having a movement v along the vertical direction, we then use the sum of the cross entropy loss from both channels as the final loss function:

$$\begin{aligned} L(\mathbf{Y},\mathcal{M}) = & -\sum_{i=1}^{P}\sum_{j=1}^{Q}\sum_{u\in\mathbb{D}_u^+} \mathbb{1}(\mathbf{U}(i,j)=u)\log(F_{u,i,j}(\mathbf{Y})) \\ & -\sum_{i=1}^{P}\sum_{j=1}^{Q}\sum_{v\in\mathbb{D}_v} \mathbb{1}(\mathbf{V}(i,j)=v)\log(F_{v,i,j}(\mathbf{Y})), \end{aligned}$$

where $\mathbb{1}$ is an indicator function.

3.4.2 Simulate Motion Flow for Data Generation

The gist of this section is generating a dataset that contains realistic blur patterns on diverse images for training. Although an i.i.d. random sampling may generate very diverse training samples, since the realistic motion flow preserves some properties such as piece-wise smoothness, we aim to design a simulation method to generate motion flows reflecting the natural properties of the movement in imaging process.

Although the object motion (Hyun Kim, Ahn, and Mu Lee, 2013) can lead to heterogeneous motion blur in real images, our method only simulates the motion flow caused by camera motion for learning. Even so, as shown in Section 3.5.5, data generated by our method can also give the model certain ability to handle object motion.

For simplicity, we generate a 3D coordinate system where the origin at the camera's optical center, the xy-plane is aligned with the camera sensors, and the z-axis is perpendicular to the xy-plane, as shown in Figure 3.5. Since our objective is the motion flow on an image grid, we directly simulate the motion flow projected on 2D image instead of the 3D motion trajectory (Whyte et al., 2012). Considering the ambiguities caused by rotations around x and y axis (Gupta et al., 2010), we simulate a motion flow $\mathcal{M}$ by sampling four additive components:

$$\mathcal{M} = \mathcal{M}_{T_x} + \mathcal{M}_{T_y} + \mathcal{M}_{T_z} + \mathcal{M}_{R_z}, \tag{3.6}$$

where $\mathcal{M}_{T_x}$, $\mathcal{M}_{T_y}$ and $\mathcal{M}_{T_z}$ denote the motion flows associated with the translations along x, y and z axis, receptively, and $\mathcal{M}_{R_z}$ represents the motion from the rotation around z axis. We generate each element as the following.

Translation along x or y axis We describe the generation of $\mathcal{M}_{T_x}$ as an example. We first sample a central pixel $\mathbf{p}_{T_x} = (i_{T_x}, j_{T_x})$ on image plane, a basic motion value t_{T_x} and a acceleration coefficient r_{T_x}. Then $\mathcal{M}_{T_x} = (\mathbf{U}_{T_x}, \mathbf{V}_{T_x})$ can be generated as the following $\mathbf{U}_{T_x}(i,j) = (i - i_{T_x})r_{T_x} + t_{T_x}, \mathbf{V}_{T_x}(i,j) = 0$. $\mathcal{M}_{T_y}$ can be generated in a similar way.

Translation along z axis The translation along z axis usually causes radial motion blur pattern towards the vanishing point (Zheng, Xu, and Jia, 2013). By ignoring the semantic context and assuming a simple radial pattern, $\mathcal{M}_{T_z}$ can be generated by $\mathbf{U}_{T_z}(i,j) = t_{T_z} d(i,j)^{\zeta}(i - i_{T_z}), \mathbf{V}_{T_z}(i,j) = t_{T_z} d(i,j)^{\zeta}(j - j_{T_z})$ where $\mathbf{p}_{T_z}$ denotes a sampled vanishing point, $d(i,j) = \|(i,j) - \mathbf{p}_{T_z}\|_2$ is the distance from any pixel (i,j) to the vanishing point, ζ and t_{T_z} are used to control the shape of radial patterns, which reflects the moving speed.

Rotation around z axis We first sample a rotation center $\mathbf{p}_{R_z}$ and an angular velocity ω, where $\omega > 0$ denotes the clockwise rotation. Let $d(i,j) = \|(i,j) - \mathbf{p}_{R_z}\|_2$. The motion magnitude at each pixel is $s(i,j) = 2d(i,j)\tan(\omega/2)$. By letting $\theta(i,j) = \text{atan}[(i - i_{R_z})/(j - j_{R_z})] \in [-\pi, \pi]$, motion vector at pixel (i,j) can be generated as $\mathbf{U}_{R_z}(i,j) = s(i,j)\cos(\theta(i,j) - \pi/2), \mathbf{V}_{R_z}(i,j) = s(i,j)\sin(\theta(i,j) - \pi/2)$.

We place uniform priors over all the parameters corresponding to the motion flow simulation as Uniform(α, β). More details can be found in supplementary materials. Note that the four components in (3.6) are simulated in continuous domain and are then discretized as integers.

Training dataset generation We use 200 training images with sizes around 300×460 from the dataset BSD500 (Arbelaez et al., 2011) as our sharp image set $\{\mathbf{X}^n\}$. We then independently simulate 10,000 motion flow maps $\{\mathcal{M}^t\}$ with ranges $u_{max} = v_{max} = 36$ and assign each $\mathbf{X}^n$ 50 motion flow maps without duplication. The non-blurred images $\{\mathbf{X}^n\}$ with $\mathbf{U}(i,j) = 0$ and $\mathbf{V}(i,j) = 0, \forall i,j$ are used for training. As a result we have a dataset with 10,200 blurred-image-motion-flow pairs $\{\mathbf{Y}^t, \mathcal{M}^t\}$ for training.

3.5 Experiments

We implement our model based on Caffe (Jia et al., 2014) and train it by stochastic gradient descent with momentum and batch size 1. In the training on the dataset simulated on BSD, we use a learning rate of 10^{-9} and a step size of 2×10^5. The training converges after 65 epochs.

3.5.1 Datasets and Evaluation Metrics

Datasets We conduct the experiments on both *synthetic datasets* and *real-world images*. Since ground truth motion flow and sharp image for real blurry image are difficult to obtain, to perform general quantitative evaluation, we first generate two synthetic datasets, which both contain 300 blurred images, with 100 sharp images

Dataset	Metric	GT $\mathcal{K}$	Xu and Jia, 2010	Whyte et al., 2012	Xu, Zheng, and Jia, 2013	noMRF (Sun et al., 2015)	patchCNN (Sun et al., 2015)	Ours
BSD-S	PSNR	23.022	17.773	17.360	18.351	20.483	20.534	**21.947**
	SSIM	0.6609	0.4431	0.3910	0.4766	0.5272	0.5296	**0.6309**
BSD-M	PSNR	24.655	19.673	18.451	20.057	22.789	22.9683	**23.978**
	SSIM	0.7481	0.5661	0.5010	0.5973	0.6666	0.6735	**0.7249**

TABLE 3.1. Evaluation on motion blur estimation. Comparison on PSNR and SSIM of the recovered images with the estimated blur kernel. The best results are bold-faced.

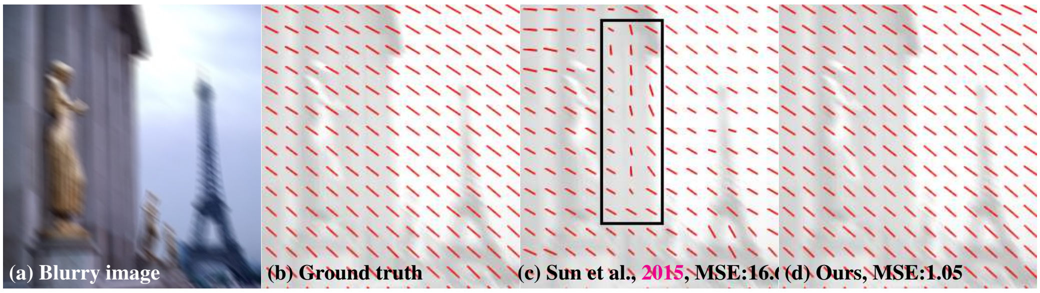


FIGURE 3.6. A motion flow estimation example on a synthetic image in BSD-M. The method of Sun et al., 2015 is more sensitive to the image content (See the black box in (c)).

randomly picked from BSD500 (Arbelaez et al., 2011)[2], and 3 different motion flow maps for each. Note that no two motion flow maps are the same. We simulate the motion flow with $u_{max} = v_{max} = 36$, which is same as in the training set. For fairness to the method of Sun et al. (2015) with a smaller output space, we also generate relative mild motion flows for the second dataset with $u_{max} = v_{max} = 17$. These two are referred as **BSD-S** and **BSD-M**, respectively. In addition, we evaluate the generalization ability of the proposed method using two synthetic datasets (**MC-S** and **MC-M**) with 60 blurry images generated from 20 sharp images from Microsoft COCO (Lin et al., 2014) and above motion flow generation setting.

Evaluation Metrics For evaluating the accuracy of estimated motion flow, we measure the mean-squared-error (**MSE**) of the motion flow map. Specifically, given an estimated motion flow $\widehat{\mathcal{M}}$ and the ground truth $\mathcal{M}$, the MSE is defined as $\frac{1}{2|M|}\sum_{i,j}((\mathbf{U}(i,j)-\widehat{\mathbf{U}}(i,j))^2+((\mathbf{V}(i,j)-\widehat{\mathbf{V}}(i,j))^2$, where $|\mathcal{M}|$ denotes the number of motion vectors in $\mathcal{M}$. For evaluation of the image quality, we adopt peak signal-to-noise-ratio (**PSNR**) and structural similarity index (**SSIM**) (Wang et al., 2004).

3.5.2 Evaluation of Motion Flow Estimation

We first compare with the method of Sun et al., 2015 ("patchCNN"), which is the only method with available code for estimating motion flow from blurry images[3]. This method performs training and testing on small image patches, and uses MRF to

[2] No overlapping with the training dataset.

[3] The code of the other motion flow based method (Kim and Lee, 2014) is unavailable.

improve the accuracy on the entire image. Its version without MRF post-processing ("noMRF") is also compared, where the soft-max output is directly used to get the motion flow as in our method. Table 3.2 shows the average MSE of the estimated motion flow maps on all images in BSD-S and BSD-M. It is noteworthy that, even without any post-processing such as MRF or CRF, the comparison manifests the high quality of our estimated motion flow maps. Furthermore, our method can still produce accurate motion flow even on the more challenging BSD-S dataset, on which the accuracies of the patch based method (Sun et al., 2015) decrease significantly. We also show an example of the the estimated motion flows in Figure 3.6, which shows that our result preserves a smooth motion flow very similar to the ground truth, and the method of Sun et al., 2015 is more sensitive to the image contents. From this example, we can see that the method of Sun et al., 2015 generally underestimates the motion values and produces errors near the strong edges, maybe because its patch-level processing is confused by the strong edges and ignores the blur pattern context in a larger area.

Dataset	patchCNN (Sun et al., 2015)	noMRF (Sun et al., 2015)	Ours
BSD-S	50.1168	54.4863	**6.6198**
BSD-M	15.6389	20.7761	**5.2051**

TABLE 3.2. Evaluation on motion flow estimation (MSE). The best results are bold-faced.

To compare with other blind deblurring methods of Xu and Jia (2010), Xu, Zheng, and Jia (2013) and Whyte et al. (2012), which do not estimate the motion flow, we directly evaluate the quality of the image recovered using their estimated blur kernel. For fairness, we use the same non-blind deconvolution method with least square loss function and a Gaussian mixture model prior (Zoran and Weiss, 2011) to recover the sharp image. As the non-blind deconvolution method may limit the recovering quality, we evaluate the images recovered using the ground truth motion flow as reference. Table 3.1 shows the average values on all images in each dataset, which shows that our method produce significantly better results than the others.

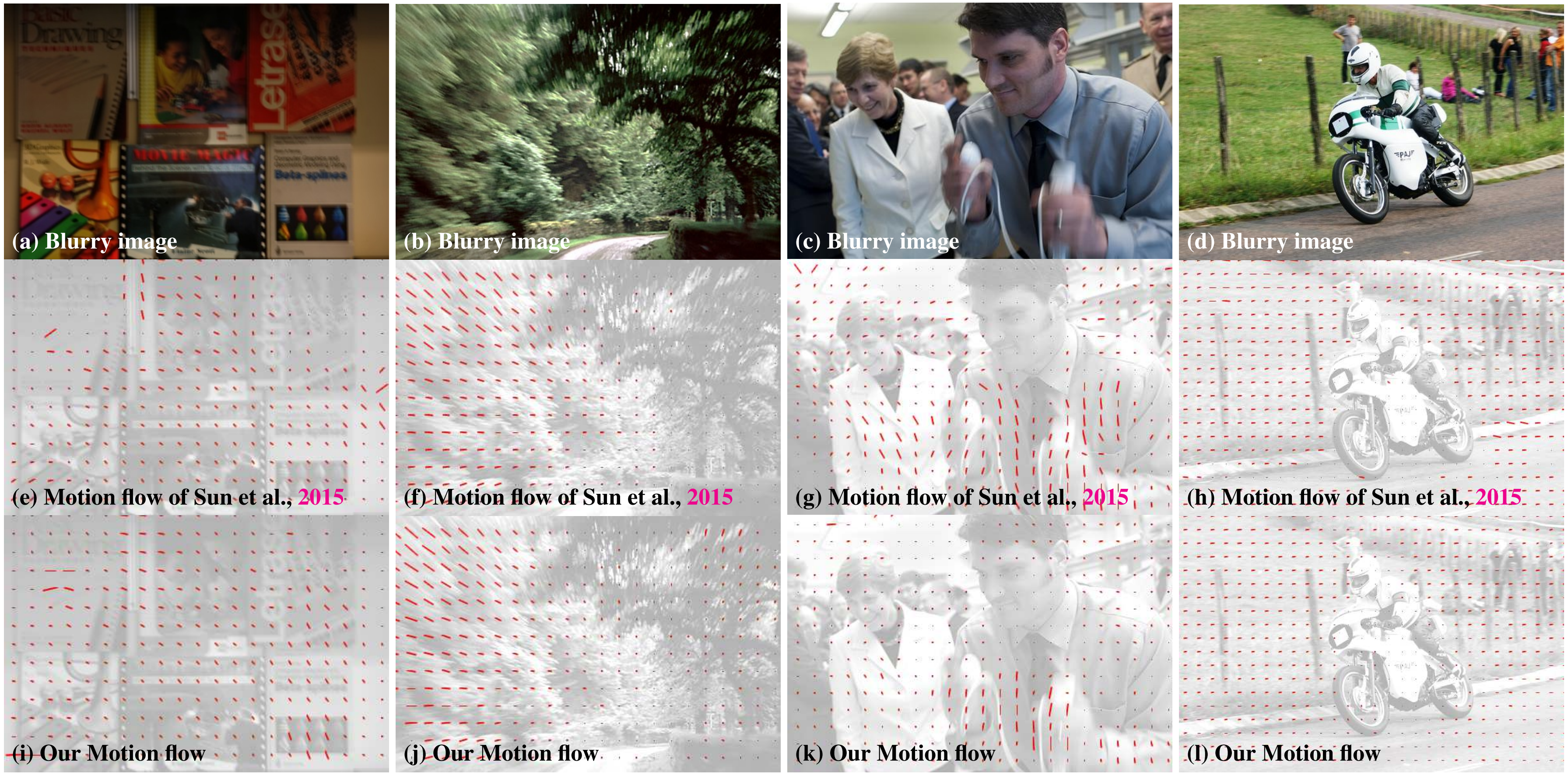

(a) Blurry image
(b) Blurry image
(c) Blurry image
(d) Blurry image
(e) Motion flow of Sun et al., 2015
(f) Motion flow of Sun et al., 2015
(g) Motion flow of Sun et al., 2015
(h) Motion flow of Sun et al., 2015
(i) Our Motion flow
(j) Our Motion flow
(k) Our Motion flow
(l) Our Motion flow

FIGURE 3.7. Examples of motion flow estimation on real-world blurry images. From top to bottom: Blurry image $\mathbf{Y}$, motion flow estimated by the patchCNN (Sun et al., 2015), and by our motion flow $\mathcal{M}$. Our results are more smooth and more accurate on moving objects.

3.5.3 Evaluation of Generalization Ability

To evaluate the generalization ability of our approach on different images, we use the datasets based on the Microsoft COCO (Lin et al., 2014) (*i.e.* MC-S and MC-M) to evaluate our model trained on the dataset based on BSD500 (Arbelaez et al., 2011). Table 3.3 shows the evaluation and comparison with the "patchCNN" (Sun et al., 2015). The results demonstrate that our method stably produces high accuracy results on both datasets. This experiment suggests that the generalization ability of our approach is strong.

Dataset	Metric	GT $\mathcal{K}$	patchCNN	noMRF (Sun et al., 2015)	Ours
MC-S	MSE	–	52.1234	60.9397	**7.8038**
	PSNR	22.620	20.172	20.217	**21.954**
	SSIM	0.6953	0.5764	0.5772	**0.6641**
MC-M	MSE	–	22.4383	31.2754	**7.3405**
	PSNR	23.827	22.186	22.028	**23.227**
	SSIM	0.7620	0.6924	0.6839	**0.7402**

TABLE 3.3. Evaluation of the generalization ability on datasets MC-S and MC-M. The best results are bold-faced.

3.5.4 Running-time Evaluation

We conduct a running-time comparison with the relevant motion flow estimation methods (Sun et al., 2015; Kim and Lee, 2014) by running motion flow estimation for 60 blurred images with sizes around 640×480 on a PC with an NVIDIA GeForce 980 graphics card and Intel Core i7 CPU. For the method in Kim and Lee (2014), we quote its running-time from the paper. Note that both the method of Sun *et.al.*and ours use the GPU to accelerate the computation. As shown in Table 3.4, the method in Kim and Lee (2014) takes very long time due to its iterative optimization scheme. Our method takes less than 10 seconds, which is more efficient than others. The patchCNN method (Sun et al., 2015) takes more time because many post-processing steps are required.

Method	Kim and Lee (2014)	patchCNN (Sun et al., 2015)	noMRF (Sun et al., 2015)	Ours
Time (s)	1500	45.2	18.5	**8.4**

TABLE 3.4. Running-time comparison.

3.5.5 Evaluation on Real-world Images

As the ground truth images of real-world blurry images are unavailable, we only present the visual evaluation and comparison against several state-of-the-art methods for spatially-varying blur removing. More results can be found in supplementary materials.

Results of motion flow estimation We first compare the proposed method with the method of Sun et al. (2015) on motion flow estimation. Four examples are shown in Figure 3.7. Since the method of Sun et al. (2015) performs on local patches, their motion flow components are often misestimated, especially when the blur pattern in a small local area is subtle or confusing, such as the areas with low illumination or textures. Thanks to the universal end-to-end mapping, our methods can generate more natural results with smooth flow and less clutters. Although we train our model on dataset with only smoothly varying motion flow, compared with Sun et al. (2015), our method can obtain better results on images with moving object.

Comparison with the method of Kim and Lee (2014) Kim and Lee (2014) use the similar heterogeneous motion blur model as ours and also estimate motion flow for deblurring. As their code is unavailable, we directly perform a comparison on their real-world data. Figure 3.8 shows the results on an example. Compared with the results of Kim and Lee (2014), our motion flow more accurately reflects the complex blur pattern, and our recovered image contains more details and less artifacts.

Images with camera motion blur Figure 3.9 shows an example containing blur mainly caused by the camera motion. The deblurred image generated by the non-uniform camera shake deblurring method (Whyte et al., 2012) suffers from heavy blur because its model ignores the blur caused by large forward motion. Compared

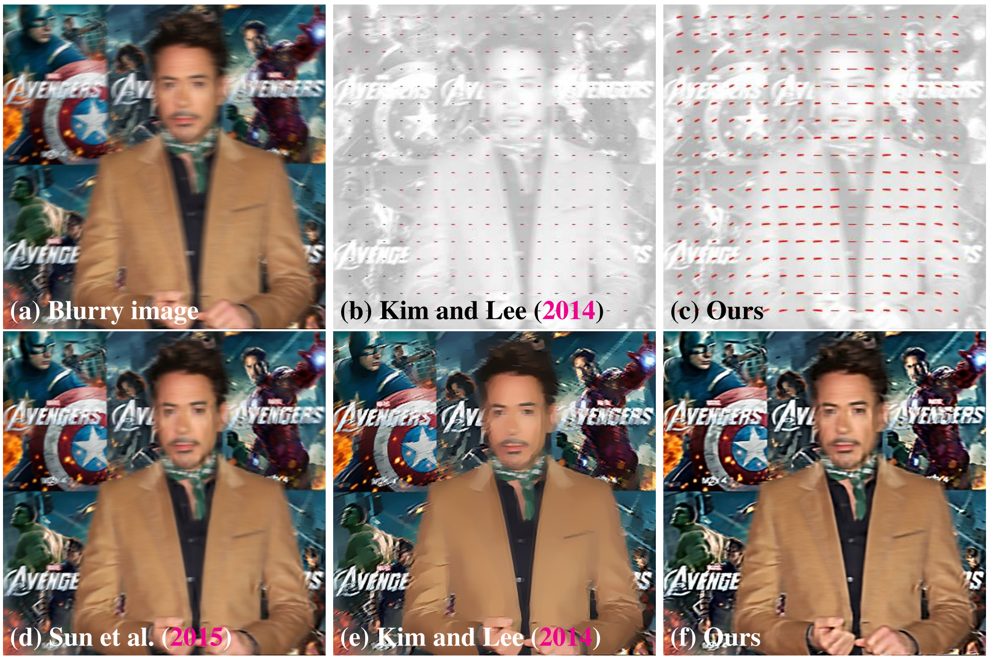
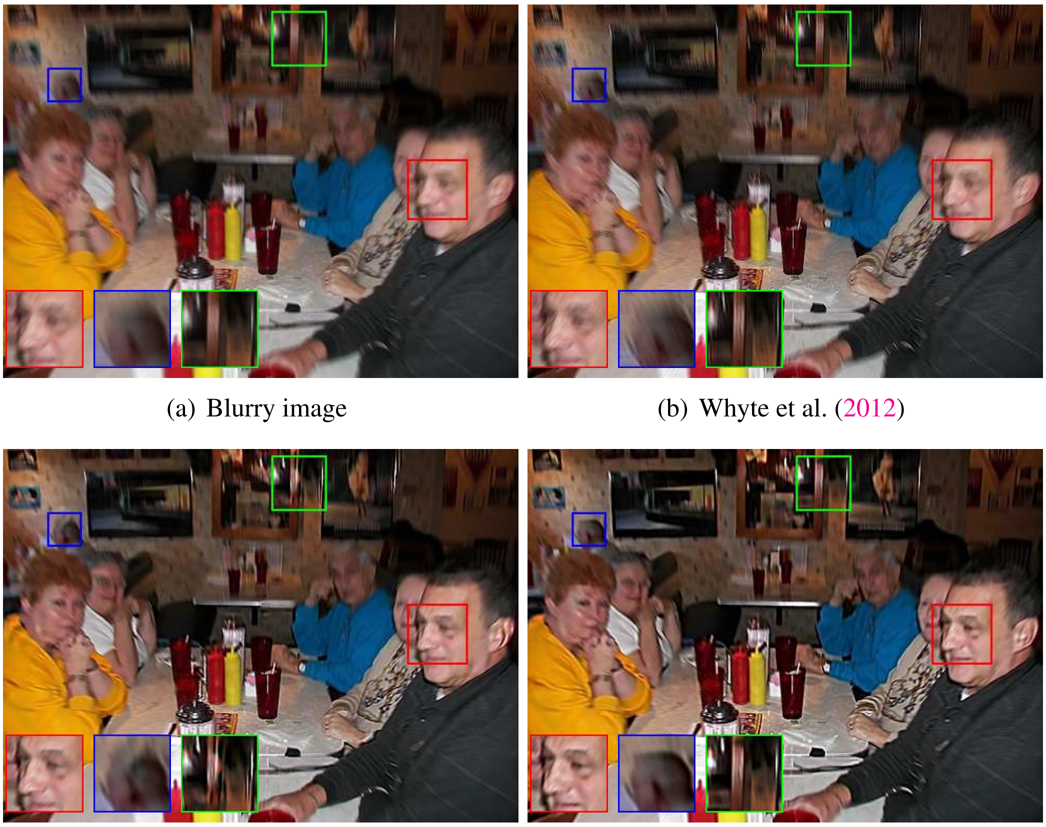


FIGURE 3.8. Comparison with the method of Kim and Lee (2014).

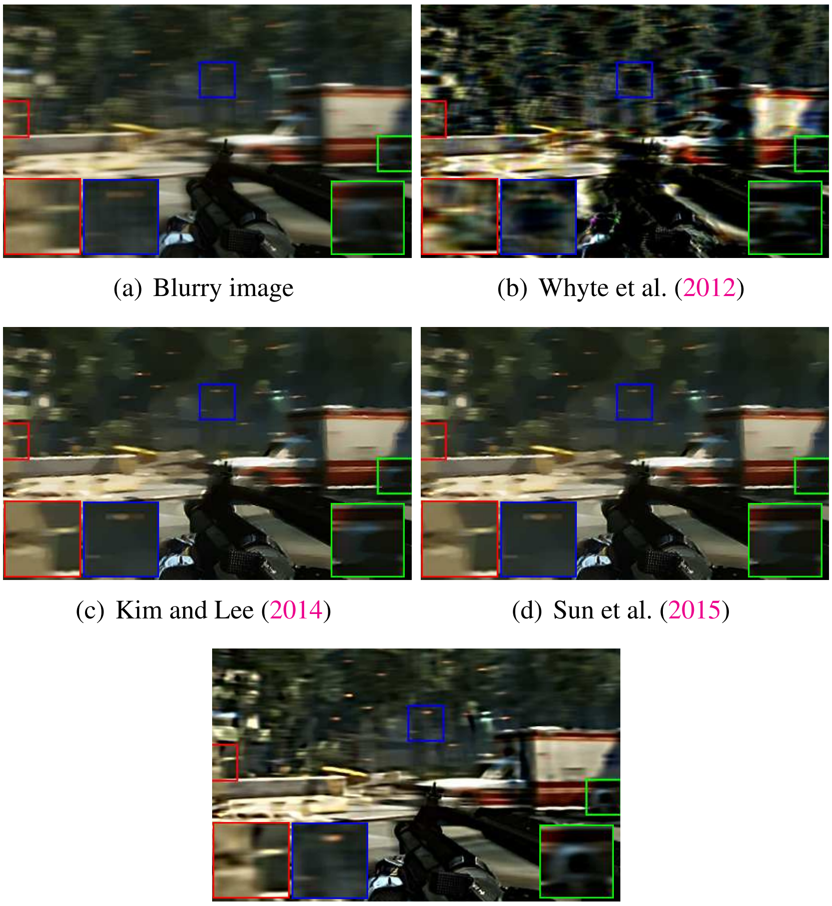

(a) Blurry image

(b) Whyte et al. (2012)

(c) Sun et al. (2015)

(d) Ours

FIGURE 3.9. Deblurring results on an image with camera motion blur.



(a) Blurry image

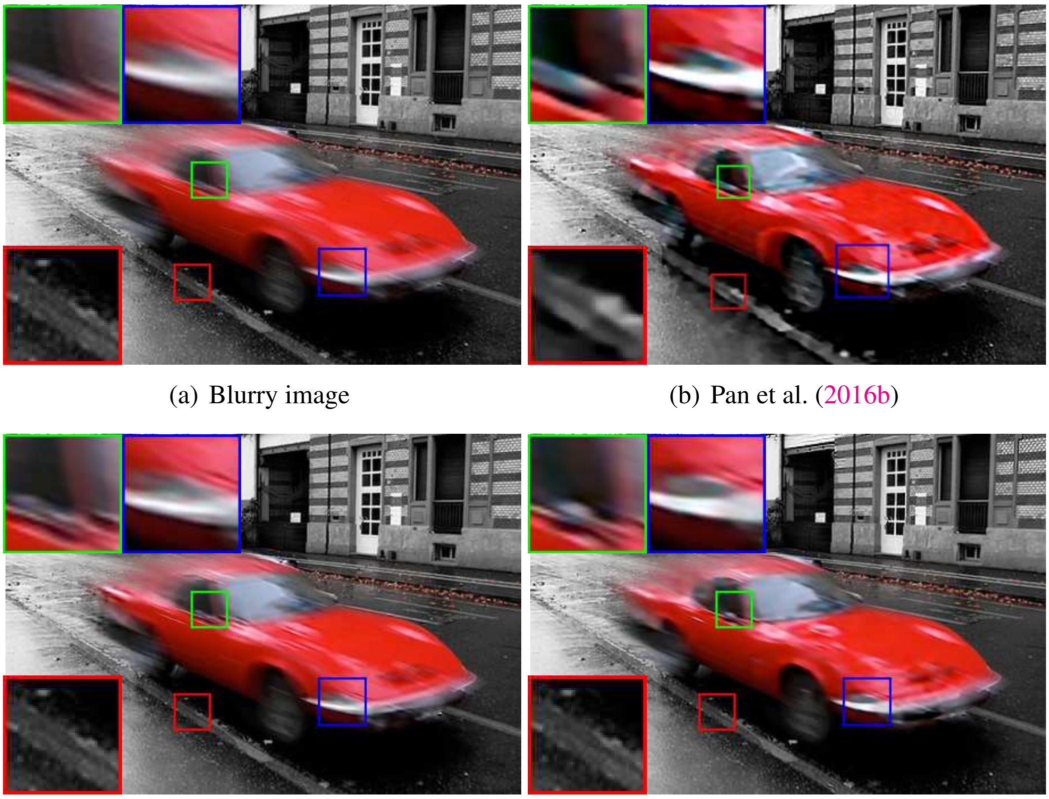
(b) Whyte et al. (2012)

(c) Kim and Lee (2014)

(d) Sun et al. (2015)

(e) Ours

FIGURE 3.10. Deblurring results on an non-uniform blur image with strong blur on background.



(a) Blurry image

(b) Pan et al. (2016b)

(c) Sun et al. (2015)

(d) Ours

FIGURE 3.11. Deblurring results on an image with large scale motion blur caused by moving object.

with the result of Sun et al. (2015), our method produces a sharper result with more details and less artifacts.

Images with object motion blur We evaluate our method on the images containing object motion blur. In Figure 3.10, the result of Whyte et al. (2012) contains heavy ringing artifacts due to the object motion. Our method can handle the strong blur in the background and generate a more natural image. We further compare with the segmentation-based deblurring method of Pan et al. (2016b) on an image with large scale blur caused by moving object on static background. As shown in Figure 3.11, the result of Sun et al. (2015) is oversmooth due to the underestimate of motion flow. In the result of Pan et al. (2016b), some details are lost due to the segmentation error. Our proposed method can recover the details on blurred moving foreground and keep the sharp background as original.

3.6 Conclusion

In this chapter, we proposed a flexible and efficient deep learning based method for estimating and removing the heterogeneous motion blur. By representing the heterogeneous motion blur as pixel-wise linear motion blur, the proposed method uses a FCN to estimate the a dense motion flow map for blur removal. Moreover, we automatically generate training data with simulated motion flow maps for training the FCN. Experimental results on both synthetic and real-world data show the excellence of the proposed method.

Chapter 4

Deep Learning for Single Image Reflection Removal

Contents

From this chapter, we aim at another computer vision problem: single image reflection removal. Reflections often obstruct the desired scene when taking photos through glass panels. Removing unwanted reflection automatically from the photos is highly desirable. Traditional methods often impose certain priors or assumptions to target particular type(s) of reflection such as shifted double reflection, thus have difficulty to generalize to other types. Very recently a deep learning approach has *been proposed. It learns a deep neural network that directly maps a reflection contaminated image to a background (target) image* (i.e. *reflection free image) in an end-to-end fashion, and outperforms the previous methods. We argue that, to remove reflection truly well, we should estimate the reflection and utilize it to estimate the background image. We propose a cascade deep neural network, which estimates both the background image and the reflection. This significantly improves reflection removal. In the cascade deep network, we use the estimated background image to estimate the reflection, and then use the estimated reflection to estimate the background image, facilitating our idea of seeing deeply and bidirectionally. This work was presented at ECCV 2018.*

4.1 Intrduction

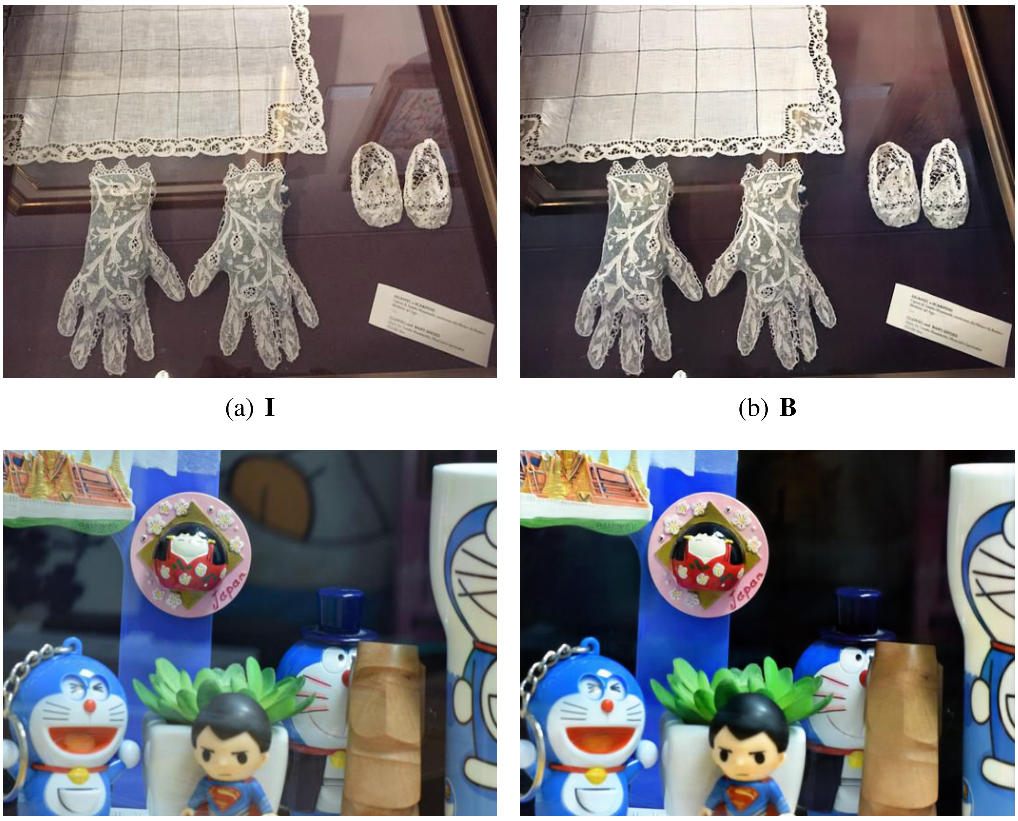
In the previous chapter, we have addressed the problem of image motion blur removal. And in this chapter, we are going to target another challenging problem, single image reflection removal.

When taking photos through windows or vitrines, reflections of the scene on the same side of the camera, often obstruct the desired scene and ruin the photos. The reflections, however, are often unavoidable due to the limitations on time and/or space. There are practical demands for image reflection removal.

To deal with the image reflection, we first assume that, without the obstruction from the reflection, we can take a clear image, $\mathbf{B} \in \mathbb{R}^{m\times n}$, and then model the reflection contaminated image $\mathbf{I} \in \mathbb{R}^{m\times n}$ as a linear combination of $\mathbf{B}$ and a reflection layer (called reflection) $\mathbf{R} \in \mathbb{R}^{m\times n}$ (Xue et al., 2015):

$$\mathbf{I} = \alpha * \mathbf{B} + (1-\alpha) * (\mathbf{K} \otimes \mathbf{R}), \tag{4.1}$$

where the real scale weight $\alpha \in (0.5, 1)$ is usually assumed as a homogeneous constant (Xue et al., 2015; Szeliski, Avidan, and Anandan, 2000; Li and Brown, 2014),



(a) **I** (b) **B**

(c) **I** (d) **B**

FIGURE 4.1. An example of single image reflection removal. (a) and (c) are images taken in front of a glass display case, which is degenerated by the reflection. (b) and (d) are the recovered background images of the proposed reflection removal method.

$\otimes$ is a convolution operator and $\mathbf{K}$ usually represents a Gaussian blurring kernel corresponding a defocus effect on the reflection. Note that $\mathbf{K}$ can also be a delta function (*i.e.* no blur on $\mathbf{R}$) to represent the case where $\mathbf{B}$ and $\mathbf{R}$ are both in-focus.

Given an image $\mathbf{I}$ contaminated by reflection $\mathbf{R}$, reflection removal aims to recover the clear background image $\mathbf{B}$. This is challenging since it is highly ill-posed (Levin and Weiss, 2007). Some methods thus require multiple images with variations in reflection and/or background as input (Xue et al., 2015; Li and Brown, 2013; Guo, Cao, and Ma, 2014; Sarel and Irani, 2004; Han and Sim, 2017) or user assistance to label the potential area of reflection and background (Levin and Weiss, 2007) to reduce the issue. Multiple images and reliable user guidance are often not easy to acquire, however. To make reflection removal practical, single image reflection removal has received increasing attentions (Li and Brown, 2014; Shih et al., 2015; Fan et al., 2017).

Solving for $\mathbf{B}$ from a single observation $\mathbf{I}$ usually requires some priors or assumptions to distinguish reflection and background. For example, the ghosting cue (Shih et al., 2015) is used to identify a special pattern of the shifted double reflection layers from two reflection surfaces. Priors on image gradients are often used to capture the different properties of the different layers (Li and Brown, 2014; Arvanitopoulos, Achanta, and Süsstrunk, 2017). These methods assume the reflection $\mathbf{K} \otimes \mathbf{R}$ is highly blurry due to out-of-focus. Relying on this, recently, a deep learning based method (Fan et al., 2017) has been proposed to achieve end-to-end single image reflection removal, which utilizes strong edges to identify the background scene, and is trained on the images synthesized with highly blurry reflection layers.

These methods have achieved state-of-the-art performance on many testing examples. However, they also exhibit some limitations in practices such as oversmoothing the image, can not handle the case when the reflections do not have strong blurry or have similar brightness and structure with the background. In this chapter, considering the success of the deep learning on image restoration (Ledig et al., 2017; Gong et al., 2017; Mao, Shen, and Yang, 2016; Gong et al., 2018), we propose to tackle the single image reflection removal by using a cascade deep neural network. Instead of training a network to estimate $\mathbf{B}$ alone from $\mathbf{I}$, we show that estimating not only $\mathbf{B}$, but also the reflection $\mathbf{R}$ (a seemingly unnecessary step), can significantly improve the quality of reflection removal. Since our network is trained to reconstruct the scenes on both sides of the reflection surface (*e.g.* glass pane), and in the cascade we use $\mathbf{B}$ to estimate $\mathbf{R}$, and use $\mathbf{R}$ to estimate $\mathbf{B}$, we call our network bidirectional network (BDN).

4.2 Related Work

Methods relying on conventional priors Single image reflection removal is a very ill-posed problem. Previous methods rely on certain priors or additional information to handle specific kinds of scenarios.

In some cases, the objects in background layer and reflection layer are approximately in the same focal plane. Some methods exploited gradient sparsity priors to decompose background and reflection with minimal gradients and local features such as edges and corners (Levin, Zomet, and Weiss, 2003; Levin, Zomet, and Weiss, 2004).

In other cases, when taking pictures of objects in the background, the objects reflected from the other side are out of focus due to the different distances to the camera, which leads to the different levels of blur in background and reflection. Li and Brown (2014) exploited the relative smoothness and proposed a probabilistic model to regularize the gradients of the two layers. In addition to ℓ_0 gradient sparsity prior, Arvanitopoulos, Achanta, and Süsstrunk (2017) proposed to impose a Laplacian data fidelity term to preserve the fine details of the original image. Wan et al. (2016) used a multi-scale Depth of Filed map to guide edge classification and used the method in Levin and Weiss (2007) for layer reconstruction afterward.

To distinguish the reflection layer from the background layer, Shih et al. (2015) studied ghosting cues, which is a specific phenomenon when the glass has a certain thickness and employed a patch-based GMM prior to model the natural image for reflection removal.

Deep learning based methods Some recent works start to employ learning based methods in reflection removal problems.

Fan et al. (2017) proposed a deep learning based methods to recover background from the image contaminated by reflections. Similar to Li and Brown (2014), it also relies on the assumption that the reflection layer is more blurry due to out of focus and they further argue that in some real-world cases, the bright lights contributes a lot to the generation of reflections. They proposed a data generation model to mimic such properties by performing additional operations on the reflection part. They proposed a two-stage framework to first predict an intrinsic edge map to guide the recovery of the background.

Zhang, Ng, and Chen (2018) used a deep neural network with a combination of perceptual loss, adversarial loss and an exclusion loss to exploit low-level and high-level image information. Wan et al. (2018) proposed to combine gradient inference and image reconstruction in one unified framework. They also employed perceptual loss to measure the difference between estimation and ground-truth in feature space.

Other related methods Many previous works use multiple observation images as additional information for the recovery of background images. Some use pairs of images in different conditions, such as flash/non-flash (Agrawal et al., 2005), different focus (Schechner, Kiryati, and Basri, 2000). Some use images from different viewpoints, such as video frames (Szeliski, Avidan, and Anandan, 2000; Sarel and Irani, 2004; Gai, Shi, and Zhang, 2012; Sinha et al., 2012; Li and Brown, 2013; Guo, Cao, and Ma, 2014; Xue et al., 2015; Yang et al., 2016), through a polarizer at multiple orientations (Schechner, Shamir, and Kiryati, 2000; Sarel and Irani, 2004; Kong, Tai, and Shin, 2014), *etc*. But in many real scenarios, we do not have the required multi-frame images for reflection removal. Some work requires manual labelling of edges belonging to reflections to distinguish between reflection and background (Levin and Weiss, 2007), which is also not suitable for general applications.

4.3 Proposed method

Focusing on reflection removal, we seek to learn a neural network which is able to recover a reflection-free image from an observation containing reflection obstruction. Specifically, our final goal is to learn a mapping function $\mathcal{F}(\cdot)$ to predict the background image $\widehat{\mathbf{B}} = \mathcal{F}(\mathbf{I})$ from an observed image $\mathbf{I}$. Instead of training only on the image pairs $(\mathbf{I}, \mathbf{B})$'s, we impose the ground truth reflection layers $\mathbf{R}$'s to boost the

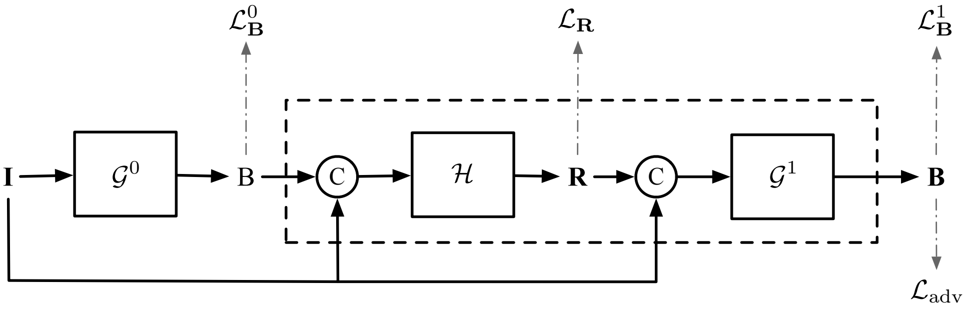

$\mathcal{L}^0_{\mathbf{B}}$
$\mathcal{L}_{\mathbf{R}}$
$\mathcal{L}^1_{\mathbf{B}}$
I
$\mathcal{G}^0$
B
C
$\mathcal{H}$
R
C
$\mathcal{G}^1$
B
$\mathcal{L}_{\text{adv}}$

FIGURE 4.2. Overview of our proposed BDN network architecture and the training objectives. Component C stands for tensor concatenation.

training of $\mathcal{F}(\cdot)$ by training on a set of triplets $\{(\mathbf{I}_t, \mathbf{B}_t, \mathbf{R}_t)\}_{t=1}^N$. Note that $\mathbf{R}_t$'s are only used in training, not in testing.

4.3.1 Bidirectional Estimation Model

To directly estimate $\mathbf{B}$ from a given $\mathbf{I}$ in an end-to-end manner, the straightforward idea is to let $\mathcal{F}(\cdot)$ be a neural network taking $\mathbf{I}$ as input and generating $\mathbf{B}$ as output. Our method also includes such a mapping function, and we call it *vanilla generator* $\mathcal{G}^0(\cdot)$. However, our solution further introduces two mapping networks $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$ to estimate the reflection image and refine the background image estimation. In the following parts, we call a composition of $\mathcal{H}$ and $\mathcal{G}^1$ as the bidirectional unit since together they provide estimates for both reflection and background images based on the output of the vanilla generator. The overall structure of the proposed network is shown in Fig. 4.2.

Vanilla generator The vanilla generator takes the observation $\mathbf{I}$ as the input and generates a background image $\mathbf{B}^0$, *i.e.* $\mathbf{B}^0 = \mathcal{G}^0(\mathbf{I})$, which is the input to the following bidirectional unit.

Bidirectional unit As shown in Fig. 4.2, the bidirectional unit consists of two components, one for predicting the reflection image and the other for predicting the background image. The first component $\mathcal{H}(\cdot)$ in the bidirectional estimates the reflection image $\mathbf{R}$ from the observation $\mathbf{I}$ and the background estimation $\mathbf{B}^0$ from $\mathcal{G}^0$, *i.e.* $\mathbf{R} = \mathcal{H}(\mathbf{B}^0, \mathbf{I})$. After that, another background estimator $\mathcal{G}^1(\cdot)$ refines the background estimation by utilizing information from the estimation of $\mathbf{R}$ and the original observation $\mathbf{I}$. Thus, the final estimation of background image is calculated by

$$\hat{\mathbf{B}} = \mathcal{G}^1(\mathcal{H}(\mathbf{B}^0, \mathbf{I}), \mathbf{I}). \tag{4.2}$$

The motivation of using the above bidirectional estimation model is the mutual dependency of the estimation of reflection images and background images. Intuitively, if a good estimation of the reflection image is provided, it will be easier to estimate the background image, vice versa. Also, including the objective of recovering the reflection image provides additional supervision signals to train the network.

Bidirectional prediction model Based on the above definition of $\mathcal{G}^0(\cdot)$, $\mathbf{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$, we can formulate the whole bidirectional prediction model as:

$$\hat{\mathbf{B}} = \mathcal{G}^1(\mathcal{H}(\mathcal{G}^0(\mathbf{I}), \mathbf{I}), \mathbf{I}), \tag{4.3}$$

which only takes the observation $\mathbf{I}$ as input. The model shown in Eq. (4.3) approaches the mapping function $\mathcal{F}(\cdot)$ from the observation $\mathbf{I}$ to the background image B via a composition of $\mathcal{G}^0(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$.

4.3.2 Network Structure for $\mathcal{G}^0(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$

The proposed BDN mainly consists of three subnetworks $\mathcal{G}^0(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$. We employ a variation of U-net (Ronneberger, Fischer, and Brox, 2015; Isola et al., 2017) to implement $\mathcal{G}^0(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$. All the three modules share the same network structure (except for the first convolutional layer) but not the same parameters. $\mathcal{G}^0(\cdot)$ has 14 layers, while $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$ has 10 layer. The structure of the network structure is illustrated in Fig. 4.3.

The U-net employed here contains an encoder part and a decoder part. For the encoder network, all convolution layers are followed by BatchNorm layer (Ioffe and

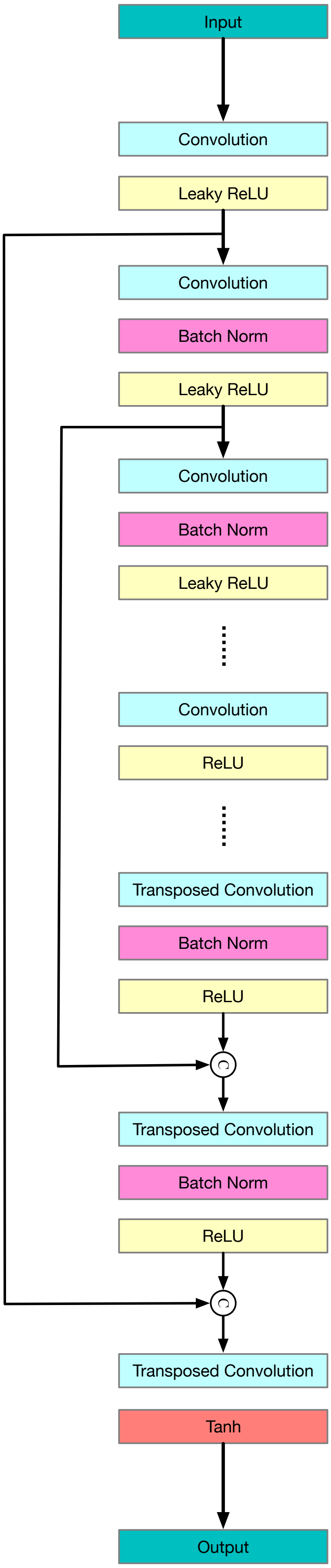


FIGURE 4.3. The network structure of $\mathcal{G}^0$, $\mathcal{H}$ and $\mathcal{G}^1$. C stands for tensor concatenation.

Szegedy, 2015) and leaky ReLU with slope 0.2, except for the first convolution, which does not have Batch-Norm. For the decoder network, each transposed convolution with stride 2 is used to upsample the feature maps by a factor of 2. The output channel is followed by a Tanh function. All convolutions are followed by a BachNorm layer and a leaky ReLU activation. The kernel size of the filters in all the convolution and transposed convolution layers is fixed to 4×4. The skip connections concatenate each channel from layer i to layer $n - i$ where n is the number of layers. The skip connections combine the information from different layers, specifically allowing low-level information to be shared between input and output. The use of skip connections doubles the number of input channels in the decoder network. The inputs of $\mathcal{H}(\cdot)$ and $\mathcal{G}^1(\cdot)$ are two images. We simply concatenate those two images to make the input have 6 channels rather than 3 color channels.

4.4 Network Training

4.4.1 Training Objective

The goal of our network is to learn a mapping function from $\mathbf{I}$ to $\mathbf{B}$ given training samples $\{(\mathbf{I}_t, \mathbf{B}_t, \mathbf{R}_t)\}_{t=1}^{N}$.

Our model consists of three mapping operations: $\mathcal{G}^0 : \mathbf{I} \to \mathbf{B}$, $\mathcal{H} : (\mathbf{I}, \mathbf{B}) \to \mathbf{R}$ and $\mathcal{G}^1 : (\mathbf{I}, \mathbf{R}) \to \mathbf{B}$. Each of the above mapping operations leads to a loss for comparing the compatibility of the estimation and the ground-truth results. In this work, we consider to minimizer the difference between the estimate and the ground truth relying on the ℓ_2-loss and the adversarial loss.

(1) ℓ_2-loss

ℓ_2-loss is widely used to measure the Euclidean distance between the estimated image and the ground-truth image. Minimizing the ℓ_2-loss favors the small mean-squared-error (MSE). Since we have three estimations from the three subnetworks in our network, three respective loss terms are defined and the summation of the three loss term will be used to train the network:

$$\mathcal{L}_2 = \mathcal{L}_{\mathbf{B}}^0 + \mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\mathbf{B}}^1, \tag{4.4}$$

where

$$\mathcal{L}_{\mathbf{B}}^0 = \sum_{t=1}^{N} ||\mathcal{G}^0(\mathbf{I}_t) - \mathbf{B}_t||_2, \tag{4.5}$$

$$\mathcal{L}_{\mathbf{R}} = \sum_{t=1}^{N} ||\mathcal{H}(\mathbf{I}_t, \mathbf{B}) - \mathbf{R}_t||_2, \tag{4.6}$$

$$\mathcal{L}_{\mathbf{B}}^1 = \sum_{t=1}^{N} ||\mathcal{G}^1(\mathbf{I}_t, \mathbf{R}) - \mathbf{B}_t||_2. \tag{4.7}$$

In (4.6) and (4.7), the $\mathbf{B}$ and $\mathbf{R}$ can be the ground truth $\mathbf{B}_t$ or $\mathbf{R}_t$ or the estimates from previous blocks, which depends on the settings in training (See Section 4.4.2).

(2) Adversarial loss

ℓ_2-loss only calculates the pixel-wise difference between two images, which may not reflect the perceptual difference between two images. Recently, there are an increasing number of works (Isola et al., 2017; Zhu et al., 2017; Ledig et al., 2017; Lettry, Vanhoey, and van Gool, 2018; Shrivastava et al., 2017) applying the adversarial loss (Goodfellow et al., 2014) to provide additional supervision for training an image mapping network. The adversarial loss was originally proposed in Generative adversarial networks (Goodfellow et al., 2014). The idea is to iteratively train a discriminator to differentiate the ground-truth images from the images generated by a generator at the certain stage of training. Then the objective becomes to encourage the generator to generate images that can confuse the current discriminator. When applying such an adversarial loss to image processing (mapping), we treat the mapping function that maps the observations to the desired output as the generator. The discriminator in the adversarial loss implicitly learns a distribution of the natural images, as an image prior. By applying adversarial loss, the implicit image prior performs as guidance for recovering the images following the natural image distribution. To simplify the training process, we only apply this adversarial loss to the last estimation of the background image, namely, the output of $\mathcal{G}^1$. Formally, the generation function is defined as $\mathcal{F}(\mathbf{I}) = \mathcal{G}^1(\mathcal{H}(\mathbf{B}^0, \mathbf{I})$ and a discriminator $\mathcal{D}$ is trained by optimizing the following objective:

$$\mathcal{L}_{\mathcal{D}} = \sum_{t=1}^{N} \log \mathcal{D}(\mathbf{B}_t) + \sum_{t=1}^{N} \log(1 - \mathcal{D}(\mathcal{F}(\mathbf{I}_t))), \tag{4.8}$$

and the adversarial loss is defined as

$$\mathcal{L}_{\text{adv}} = \sum_{t=1}^{N} -\log \mathcal{D}(\mathcal{F}(\mathbf{I}_t)) \tag{4.9}$$

Full objective Finally, we sum the ℓ_2 loss and adversarial loss as the final objective:

$$\mathcal{L} = \mathcal{L}_2 + \lambda \mathcal{L}_{\text{adv}}, \tag{4.10}$$

where λ is the hyper-parameter that controls the relative importance of the two objectives.

4.4.2 Training Strategies

Our proposed network has three cascaded modules, the vanilla generator, the reflection estimator and the refined background estimator. These components can be trained independently or jointly. In our work, we explored three ways to conduct training:

- The most straightforward way is to train the whole network end-to-end from scratch.

- Each module can also be trained independently. Specifically, we can progressively train each component until converged and then stack its output to the next component as the input. We call this training strategy as greedy training.

- We can also first train each sub-network progressively and then fine-tune the whole network, which is referred as "greedy training + fine-tuning".

In Section 4.5.1, we will present the comparison and analysis of these training strategies.

4.4.3 Implementation

Training data generation We use the model in Eq. (4.1) to simulate the images with reflections. To synthesize one image, we sample two natural images from the dataset and randomly crop the images into 256×256 patches. One patch is served as background $\mathbf{B}$ and the other is used as reflection $\mathbf{R}$. A Gaussian blur kernel of standard deviation $\sigma \in [0, 2]$ is applied on the reflection patch to simulate the defocus blur may appear on the reflection layer in reality. The two patches are blended using scale weight $\alpha \in [0.6, 0.8]$. The generated dataset contains triplets of $\{(\mathbf{I}_t, \mathbf{B}_t, \mathbf{R}_t)\}_{t=1}^{N}$.

We use images from PASCAL VOC dataset (Everingham et al., 2010) to generate our synthetic data. The dataset contains natural images in a variety of scenes, and it is suitable to represent the scenes where the reflection is likely to occur. We generate 50K training images from the training set of PASCAL VOC dataset, which contains 5717 images.

To compare with (Fan et al., 2017), which is the only available learning based method as far as we know, we also use the method introduced by Fan et al. (2017) to generate another training dataset. It subtracts an adaptively computed value followed by clipping to avoid the brightness overflow when mixing two images. We use the same setting as Fan et al. (2017) in data synthesis. The images are also from PASCAL VOC dataset and are cropped at 224×224. The training data is generated from 7643 images, and test set is generated from 850 images. We trained our network and the network of Fan et al. (2017) using both our training data and training data generated by the method of Fan et al. (2017).

Training details We implement our model using PyTorch and train the models using Adam optimizer (Kingma and Ba, 2014) using the default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate is set to be 0.001. Weights are initialized using the method of He et al. (2015). The code is available at `https://github.com/yangj1e/bdn-refremv`.

4.5 Experiments

In this section, we first present comparisons of ablations of our methods to illustrate the significance of our design decisions. Then we quantitatively and qualitatively evaluate our approach on single image reflection removal against previous methods (Li and Brown, 2014; Arvanitopoulos, Achanta, and Süsstrunk, 2017; Fan et al., 2017) and demonstrate state-of-the-art performance. For numerical analysis, we employed peak-signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) (Wang et al., 2004) as evaluation metrics.

4.5.1 Ablation Studies for the Bidirectional Network

Testing data For ablation studies, we use a dataset synthesized from PASCAL VOC (Everingham et al., 2010) validation set, which does not contain any images appeared in the training set. We generate 400 images for testing in ablation studies. The setting of testing data generation is the same as the setting in Secion 4.4.3 for training data generation.

To analyze the performance of reflection removal with respect to the scale weight of the background, which reflects relative strength between background and reflection, we generate another smaller dataset. We increment the scale weight from 0.55 to 0.85, with a step size of 0.05 and generate 10 images for each scale weight.

Analysis of the model structure To verify the importance of our bidirectional unit, we compare three model structures: vanilla generator $\mathcal{G}^0$, vanilla generator $\mathcal{G}^0$ +

reflection estimator $\mathcal{H}$, and the full bidirectional network (*i.e.* the composition of $\mathcal{G}^0$, $\mathcal{H}$ and $\mathcal{G}^1$, which is referred as $\mathcal{G}^0 + \mathcal{H} + \mathcal{G}^1$ in the following).

All networks are trained from scratch using the settings specified in Section 4.4.3. Since adding the bidirectional unit to vanilla generator will increase the depth of the network and the number of parameters, we cascade three blocks of the vanilla generator to match the depth and number of parameters of our full model. Table 4.1 shows that merely training a vanilla generator is not good enough to recover reflection free images. Increasing the number of layers of the vanilla generator (see Vanilla $\mathcal{G}^0$ (deep) in Table 4.1) to enhance the capacity of the model can slightly improve the performance, but it still underperforms our full model. Appending a reflection estimator to vanilla generator improved the performance by regularizing the reconstruction and cascading a background estimator to form a bidirectional unit improve the performance further. Fig. 4.4 shows several qualitative examples. It can be observed that adding background estimator improved the result of estimation the reflection layer, which enhances the recovery of background in reverse.

Ablation study of the objective functions In Table 4.1, we compare against ablations of our full loss. To employ adversarial loss, we need to train a discriminator network with our model. We adopt the 70×70 PatchGAN (Isola et al., 2017) for discriminator, which only penalizes structure at the scale of patches. To train the network with GAN, we pretrain our BDN without adversarial loss first for 2 epochs, and then use the pretrained network to initialize the generator. As the evaluation metrics like PSNR is directly related to MSE, adding adversarial loss has very little improvements compared to directly optimizing ℓ_2 loss in numerical analysis, but for visual appearance, we noticed improvements in restoring the fine details of the background, as shown in Fig. 4.5.

Analysis of training strategy We compare three training strategies specified in Section 4.4.2. Progressively training each module and then stacking them together, *i.e.* BDN (greedy training + fine-tuning) in Table 4.1, results in poor performance.

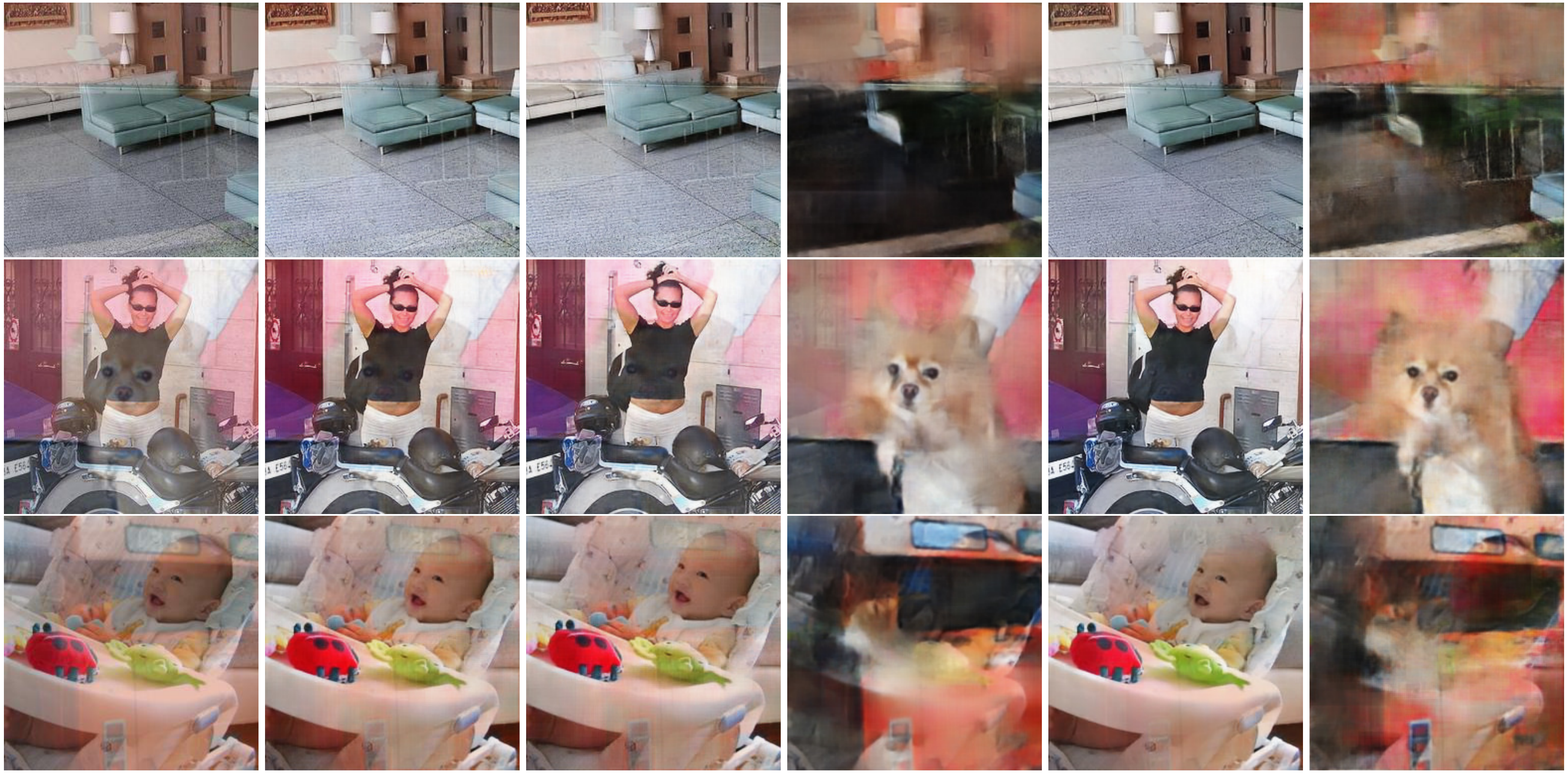


FIGURE 4.4. Visual comparison of our ablation studies on model structure. From left to right: $\mathbf{I}$, $\mathbf{B}$ ($\mathcal{G}^0$), $\mathbf{B}$ ($\mathcal{G}^0 + \mathcal{H}$), $\mathbf{R}$ ($\mathcal{G}^0 + \mathcal{H}$), $\mathbf{B}$ ($\mathcal{G}^0 + \mathcal{H} + \mathcal{G}^1$), $\mathbf{R}$ ($\mathcal{G}^0 + \mathcal{H} + \mathcal{G}^1$). Best viewed on screen with zoom.

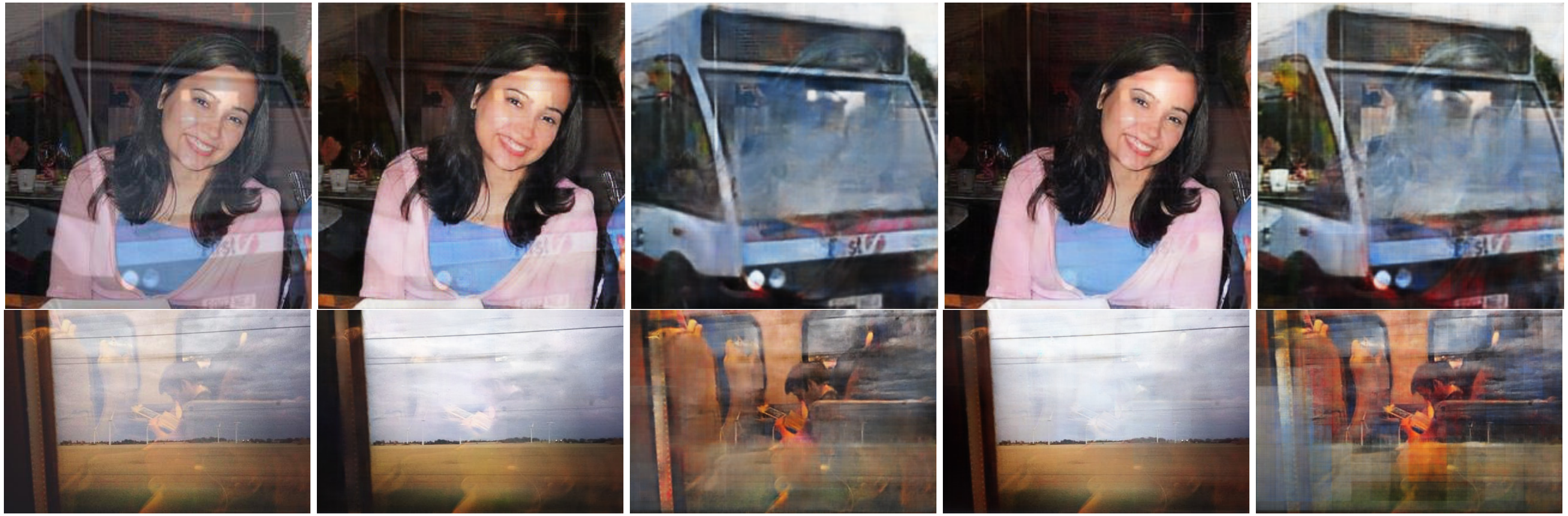


FIGURE 4.5. Visual comparison of our ablation studies on model structure on objective functions. From left to right: $\mathbf{I}$, $\mathbf{B}$ (BDN w/o adversarial loss), $\mathbf{R}$ (BDN w/o adversarial loss), $\mathbf{B}$ (BDN with adversarial loss), $\mathbf{R}$ (BDN with adversarial loss). The upper image is synthetic and the bottom image is real. Best viewed on screen with zoom.

	PSNR	SSIM
Vanilla $\mathcal{G}^0$	22.10	0.811
Vanilla $\mathcal{G}^0$ (deep)	22.16	0.817
Vanilla $\mathcal{G}^0$ + $\mathcal{H}$	22.30	0.813
BDN (greedy training)	20.82	0.792
BDN (greedy training + fine-tuning)	22.43	0.825
BDN (joint training, w/o adversarial loss)	23.06	0.833
BDN	**23.11**	**0.835**
Li and Brown (2014)	16.46	0.745
Arvanitopoulos, Achanta, and Süsstrunk (2017)	19.18	0.760
Fan et al. (2017)	19.80	0.782

TABLE 4.1. Quantitative comparison with ablation of our methods and with the state-of-the-art methods on 500 synthetic images with reflection generated using the method in Section 4.4.3, the best results are bold-faced.

The reason is that the reflection estimator and background estimator in the bidirectional unit needs to coordinate, *e.g.* if we train background estimator greedily using the ground truth pairs $\{(\mathbf{I}_t, \mathbf{B}_t)\}_{t=1}^{N}$, but when we stack it after the vanilla generator, the input of this module becomes $\{(\mathbf{I}_t, \widehat{\mathbf{B}}_t)\}_{t=1}^{N}$. Although finetuning from the progressively trained module improves performance and converges quickly, it underperforms end-to-end joint training from scratch, as the greedy initialization is more likely to converge to a bad local optima. For all the following experiments, we train our model from scratch, *i.e.* the three subnetworks are trained jointly.

4.5.2 Quantitative Evaluation

Comparison with the-state-of-the-art We perform quantitative comparison between our method and the-state-of-the-art single image reflection methods of Li and Brown (2014), Arvanitopoulos, Achanta, and Süsstrunk (2017) and Fan et al. (2017) using synthetic dataset. The numerical results shown in Table 4.1 indicates that our method outperforms the state-of-the-art.

Comparison with learning based method We specifically perform some comparisons with Fan et al. (2017) as Fan et al. (2017) is the only method of solving single image reflection removal problem using deep learning techniques so far. Both Fan

	Dataset in Fan et al., 2017		Our dataset	
	PSNR	SSIM	PSNR	SSIM
BDN (Ours)	**20.82**	0.832	**23.11**	**0.835**
Fan et al., 2017	18.29	**0.8334**	20.03	0.790

TABLE 4.2. Comparison between our method and Fan et al., 2017. Both models are trained and evaluated using the synthetic dataset of Fan et al., 2017, the best results are bold-faced.

et al. (2017) and our method require training with synthetic data, but we use different data synthesis mechanism. To compare with Fan et al. (2017), we train both our model and Fan et al. (2017) using our training data as described in Sec. 4.4.3 and a training set generated using the algorithm in Fan et al. (2017). Then we evaluate trained models on the corresponding test set, and the results are shown in Table. 4.2.

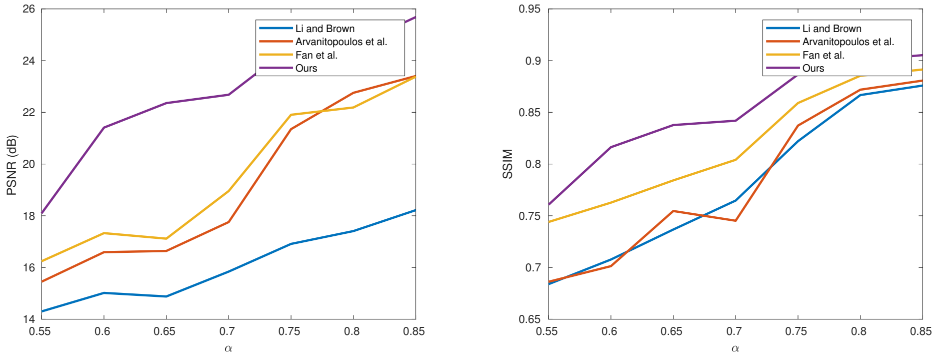
Trained on synthetic data in Fan et al. (2017), our model achieves comparable performance on the test set in Fan et al. (2017) and outperforms Fan et al. (2017) when training and testing on our synthetic dataset. Because Fan et al. (2017) explicitly utilize edge information and removes reflection by recovering the intrinsic edge of the background image, it relies more on the assumption that the reflection layer is blurry. Therefore, when training in our dataset, which is less blurry and contains a more general form of reflections, the method of Fan et al. (2017) does not perform as well as it does in Fan et al. (2017). By contrast, our model has a stronger capacity to learn from data directly and dealing with less blurry reflections.

Learning based methods train models on synthetic data due to the lack of real labeled data. Since we choose different methods to generate training data and it is difficult to tell which data synthesis method fits the real data the best, we use SIR dataset (Wan et al., 2017) to evaluate the generational ability of our model on real data with reflections. SIR dataset (Wan et al., 2017) contains 454 triplets of images shot under various capture settings, *e.g.* glass thickness, aperture size and exposure time, to cover various types of reflections. The dataset contains three scenarios: postcards, solid objects, and wild scenes. The images in this dataset are in size 540×400.

Sensitivity to the reflection level Considering the weight α in model (4.1) reflects

	Postcard		Solid objects		Wild scenes	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Fan et al. (2017)	**21.0829**	0.8294	**23.5324**	**0.8843**	22.0618	0.8261
BDN (Ours)	20.4076	**0.8548**	22.7076	0.8627	**22.1082**	**0.8327**

TABLE 4.3. Numerical study of the learning based methods on SIR benchmark dataset (Wan et al., 2017), the best results are bold-faced.



(a) PSNR

(b) SSIM

FIGURE 4.6. Evaluation of PSNR and SSIM with the change of scale weight α for the background.

the strength of the reflection level, to study the sensitivity of the proposed method to the reflection, we conduct and experiments to evaluate the performance of different methods on the images with different α's. As shown in Fig. 4.6, with the scale weight of background decreases, it is increasingly difficult to separate reflection from the background. Actually when the background layer and reflection layer have similar brightness and structure, sometimes it is even painful for humans to distinguish them apart. Also, note that the range of α exceeds the range we used in data synthesis, and our methods are robust in different levels of scale weights.

4.5.3 Qualitative Evaluation

We compare with the previous works using real images collected from previous works (Arvanitopoulos, Achanta, and Süsstrunk, 2017; Fan et al., 2017; Li and Brown, 2013) and collected from the Internet and wild scenes. Since these images have no ground truth, we can only perform the visual comparison.

Comparison with the method only estimating background Arvanitopoulos, Achanta, and Süsstrunk (2017) focus on suppressing the reflections, *i.e.* they do not recover the reflection layer. Therefore, we can only show the comparison with $\mathbf{I}$ and $\mathbf{B}$ in Fig. 4.7. It can be seen that our method better preserves the details in the background and has fewer artifacts, while Arvanitopoulos, Achanta, and Süsstrunk (2017) tends to oversmooth the image and lose too much information details. For example, in the image of clouds, our result keeps more details of cloud than Arvanitopoulos, Achanta, and Süsstrunk (2017) and in the image of the bag, our result looks more realistic.

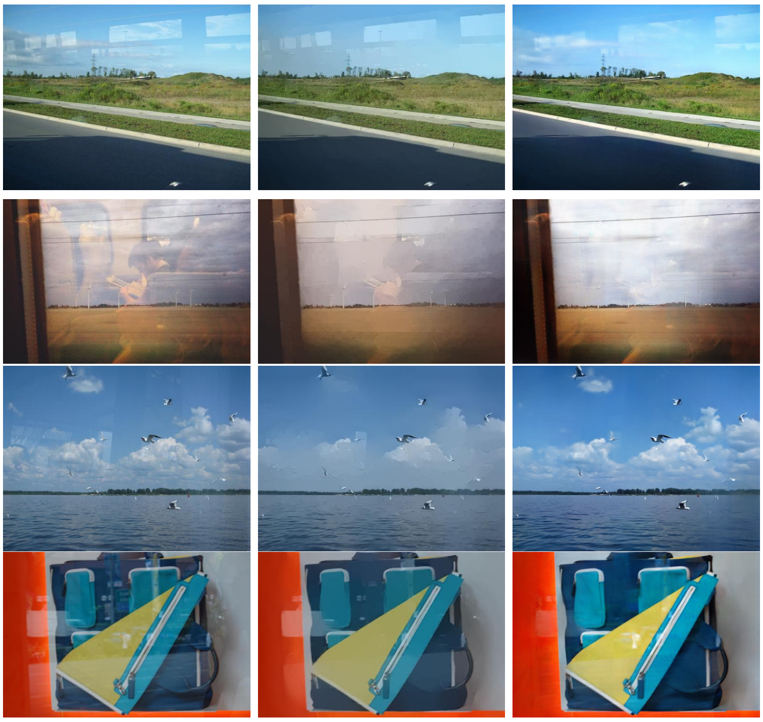


FIGURE 4.7. Comparison with the method of Arvanitopoulos, Achanta, and Süsstrunk (2017) on real images. From left to right: $\mathbf{I}$, $\mathbf{B}$ (Arvanitopoulos, Achanta, and Süsstrunk, 2017), $\mathbf{B}$ (Ours). Arvanitopoulos, Achanta, and Süsstrunk (2017) tends to be oversmooth and our results look more natural. Best viewed on screen with zoom.

Comparison with methods separating two layers We compare our methods with Li and Brown (2014), and Fan et al. (2017), which generate a reflection layer along with the background layer. Although our method focuses on recovering the background rather than separating two layers, our estimation of reflection contains more meaningful information compared to previous methods by looking bidirectional. The quality of the reflection layer reconstructed helps boost our recovery of background in our case. Fig. 4.8 shows the qualitative comparison results. Our methods outperform the state-of-the-art in recovering the clear background in real scenes with obstructive reflections. Compared to Fan et al. (2017), our method better recovers the color of the original image. Because a portion of the light will be reflected back to the side of the background, the objects in the background usually look pale compared to the observation directly without glass. This is reflected by the scale operation when generating our training data.

In Fig. 4.9, we show an examples of failure case. The image, which is from Wan et al. (2017), is taken using two postcards through a thick glass. The reflection is very strong and contains ghosting artifacts, while the background is very blurry, and the interactions between reflections have very complex structure. None of the methods works well in this case.

4.6 Conclusion

In this chapter, we studied the single image reflection removal problem. Motivated by an idea that one can estimate the reflection and use it to boost the estimation of the background, we propose a deep neural network with a cascade structure for single image removal, which is referred as the bidirectional network (BDN). Benefiting from the powerful supervision, the proposed BDN can recover the background image effectively. Extensive experiments on synthetic data and the real-world data demonstrate that the proposed methods work well in diverse scenarios.

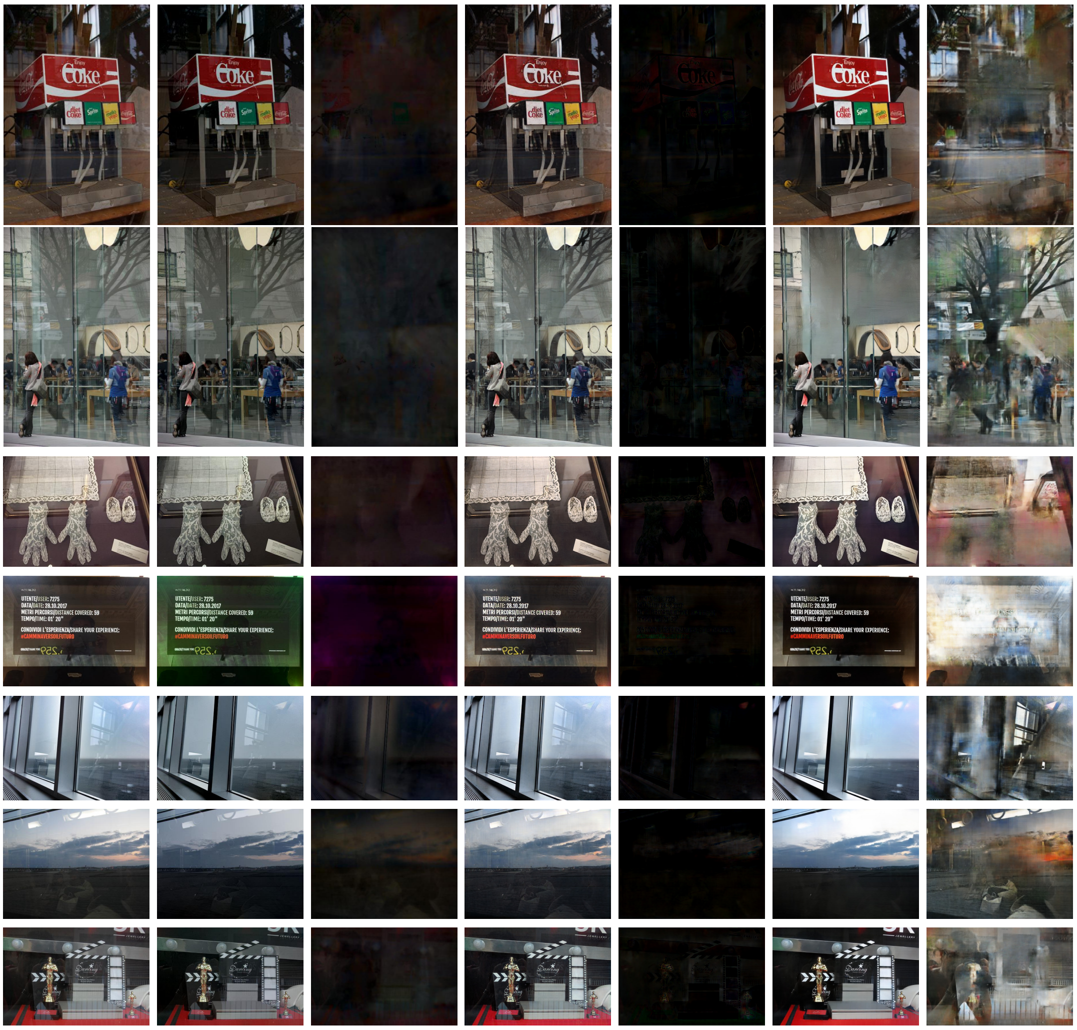


FIGURE 4.8. Comparision of our method with the-stat-of-the-art on real images. From left to right: $\mathbf{I}$, $\mathbf{B}$ (Li and Brown, 2014), $\mathbf{R}$ (Li and Brown, 2014), $\mathbf{B}$ (Fan et al., 2017), $\mathbf{R}$ (Fan et al., 2017), $\mathbf{B}$ (Ours), $\mathbf{R}$ (Ours). Our networks has clearer background estimation and better color recovery. Best viewed on screen with zoom.

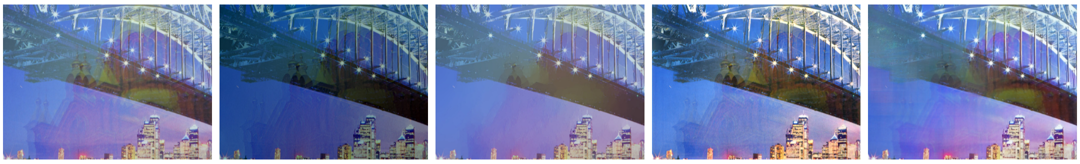


FIGURE 4.9. An example of failure case. From left to right: $\mathbf{I}$, $\mathbf{B}$ (Li and Brown, 2014), $\mathbf{B}$ (Arvanitopoulos, Achanta, and Süsstrunk, 2017, $\mathbf{B}$ (Fan et al., 2017), $\mathbf{B}$ (Ours)

Chapter 5

Self-supervised Image Reflection Removal

Contents

This chapter introduces a self-supervised image reflection removal framework. Recent learning-based methods have achieved significant improvements over traditional optimization-based methods. Still, they usually require synthetic data for training as it is impractical and time-consuming to collect real datasets with ground truth. However, the reflection synthesis process cannot fully simulate the real-world images with reflection. Considering that the taking real-world reflection images are much more convenient than taking aligned ground truth, we introduce a reflection removal approach relying on learning from the real-world image pairs with reflection *taken from multiple views. We propose a self-supervised method, only relying on the supervision from the geometry correspondence and consistency between the multi-view reflection images. Specifically, we use the multi-view reflection image pairs to train a deep network with a series of novel consistency losses that are effective and robust to utilize the* imperfect cues *derived from the multi-view consistency. The trained network achieves state-of-the-art performance in handling the real-world reflection images without supervision from the ground truth. Moreover, the proposed method can further optimize the network in the testing phase while given the multi-view images for a scene. This chapter is an early version of the work prepared for submission.*

5.1 Introduction

In chapter 4, we introduce a supervised learning method for single image reflection removal, and in this chapter, we aim at designing a self-supervised framework which do not require training data to have ground truth.

Generally, the representation for image formation with reflections can be formulated as:

$$\mathbf{I} = \mathbf{B} + \mathbf{R}, \tag{5.1}$$

where the image observed through the glass $\mathbf{I} \in \mathbb{R}^{m \times n}$ can be decomposed into a background layer $\mathbf{B} \in \mathbb{R}^{m \times n}$ and a reflection layer $\mathbf{R} \in \mathbb{R}^{m \times n}$. Recovering the background image $\mathbf{B}$ from a single observation $\mathbf{I}$ is a severely ill-posed problem as there are infinite possible decompositions. And the problem is more challenging considering that $\mathbf{B}$ and $\mathbf{R}$ may have overlapping appearance distributions and the ambiguity of reflections sometimes is difficult even for human to distinguish.

Conventional methods usually apply additional priors or assumptions to restrict the solution space and solve the problem through optimization (Levin, Zomet, and Weiss, 2003; Levin, Zomet, and Weiss, 2004; Li and Brown, 2014; Shih et al., 2015; Arvanitopoulos, Achanta, and Süsstrunk, 2017). Some methods require multiple images (Szeliski, Avidan, and Anandan, 2000; Sarel and Irani, 2004; Guo, Cao, and Ma, 2014; Xue et al., 2015; Han and Sim, 2017) or leverage manual intervention (Levin and Weiss, 2007). The methods requiring additional priors often can only handle some typical kinds of reflections, and the methods relying on multiple images or additional manual processing is not ideal for practical usage.

Deep learning-based methods have recently received increasing attention in solving the reflection removal problem (Fan et al., 2017; Yang et al., 2018; Zhang, Ng, and Chen, 2018; Wei et al., 2019; Wen et al., 2019). Compared to conventional methods, deep learning-based methods are more capable of capturing the underlying pattern of complex reflections. Most learning-based methods to solve the reflection removal problem are supervised, which requires a large amount of labeled training data, *i.e.* , pairs of real-world $\mathbf{I}$, and the corresponding $\mathbf{B}$. Although taking images with reflection is not difficult, it is usually difficult to collect the well-aligned pairs of $\mathbf{I}$ and the ground truth $\mathbf{B}$. Some data collection procedures (Zhang, Ng, and Chen, 2018; Wei et al., 2019) are proposed to collect real data relying on restricted laboratory environments. A commonly used procedure is taking $\mathbf{I}$ with a manually fixed glass in front of a fixed camera and then taking the aligned $\mathbf{B}$ and $\mathbf{R}$ by carefully removing the glass and putting a black cloth behind the glasses, respectively. The process is labor-intensive and sensitive, which makes collecting a large number of real training data infeasible. And the data collection is restricted to some specific scenarios that are easy to manipulate. Though Wei et al. (2019) proposed alignment-invariant loss functions to allow for misalignment between mixture image and background, it is still cumbersome to carry the big glass to take photos.

Due to the difficulty of collecting a large amount of real data, synthetically generated images are most commonly used as training data. Even for methods accepting some real data for training (Zhang, Ng, and Chen, 2018; Wei et al., 2019), they still use synthetic data to enrich the training dataset. Since the reflection can be very complex, existing image synthesis procedures are either simply linear combination or based on heuristic method, and the domain gap may hinder the generalization

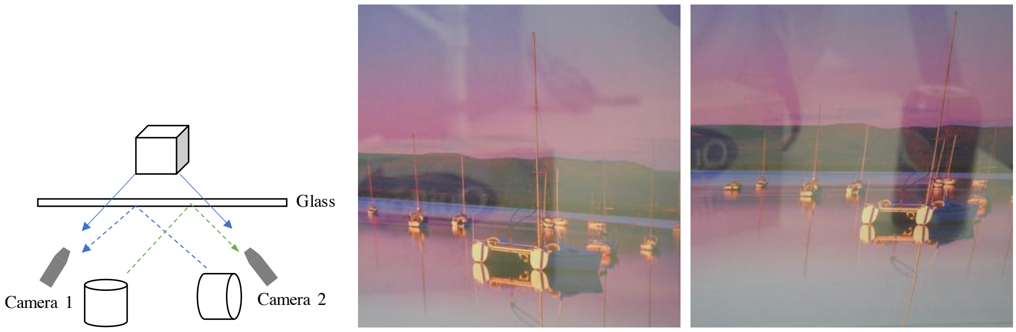


FIGURE 5.1. Demonstration of a pair of multi-view images with the same background. The reflection captured is inconsistent across the two images due to the different viewpoint the pictures are taken.

ability of the model to real data.

To address the challenges above, considering that taking $\mathbf{I}$ is much easier than taking the aligned ground truth $\mathbf{B}$ (and $\mathbf{R}$), we introduce a single image reflection removal approach that is free from the ground truth labels and only relies on the multi-view image pairs contaminated by reflections. As shown in Figure 5.1, if we observe the same background from different views, the reflections tend to be different, which helps distinguish the background from reflections. To use the multi-view cues for training the model, we propose a novel self-supervised learning method that trains the single image reflection removal model relying on the multi-view consistency. During training, the proposed method simultaneously takes image pairs targeting the same background scene captured from different views and trains the model according to the consistency between the background images estimated from the multi-view images. Although the multi-view consistency properly defines the relationship between the desired background images in the ideal case, it is imperfect to solely use the basic consistency as the supervision, due to the ambiguity of the image separation process. Thus, we introduce a series of complementary losses and integrate them to utilize the imperfect cues effectively and robustly. Since the multi-view cues are only used in training, the model is flexible to handle single image reflection removal. Moreover, as a byproduct, the proposed multi-view-based self-supervised learning approach can also be used to handle the multi-view observations seen in testing phase.

In summary, the main contributions are as follows:

- We propose a self-supervised method for single image reflection removal, in training phase only relying on the supervision from the geometry correspondence and consistency between the multi-view images contaminated by reflections, free from the ground truth labels. We design a series of novel consistency losses that are effective and robust to utilize the imperfect cues derived from the multi-view consistency. The proposed method is the first self-supervised learning approach to handle the real-world reflection removal considering the multi-view setting.

- We collected a real-world datasets containing multi-view image pairs contaminated by reflections for training the self-supervised model in conjunction with synthetic multi-view data. The proposed method can produce high-quality results on the real-world images. It achieves the best performance on existing benchmark datasets in the comparison with the methods free from the labor-intense data acquisition (similar to ours) and produces the results comparable to the methods trained with the image pairs captured in the way same to the testing data.

- Apart from training the single image reflection removal, as a byproduct, the proposed multi-view-based self-supervised learning method can also be used to further optimize the performance in testing phase when multi-view observations are available (via fine tuning on the testing data).

5.2 Related Work

Reflection removal is a very challenging problem in image processing and has been extensively studied in recent years. Previous works can be classified into the following two categories.

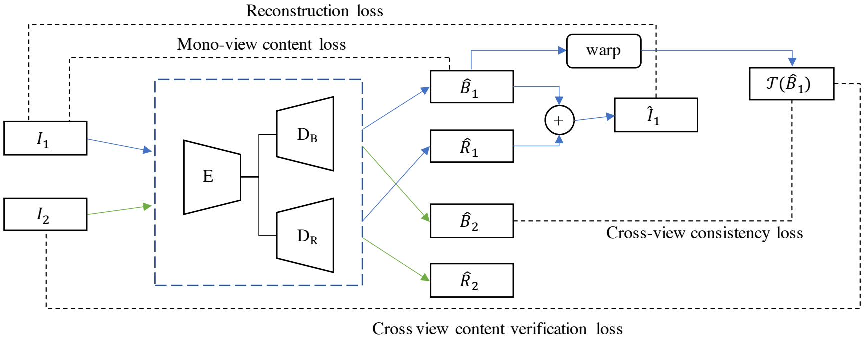


FIGURE 5.2. Overview of our proposed framework and the training objectives. For the training objectives, we only illustrate half of the loss functions as the other half is symmetric.

Methods relying on conventional priors. Conventional optimization based methods address the reflection removal problem by imposing various hand crafted priors. For example, Levin, Zomet, and Weiss (2003) and Levin, Zomet, and Weiss (2004) exploits the gradient sparsity prior to decompose layers with minimal gradients and local features such as edges and corners, Li and Brown (2014) relies on the smooth gradient priors which assumes the reflections are often less in focus, Arvanitopoulos, Achanta, and Süsstrunk (2017) is based on the Laplacian data fidelity term, and Shih et al. (2015) deals with the ghosting effects, a special phenomenon when the glass is thick and reflection is double-layered, *etc*.

There are also some methods working on multiple images, which requires the target background to be captured from different viewpoints (Sarel and Irani, 2004; Guo, Cao, and Ma, 2014; Xue et al., 2015) or camera settings (Schechner, Kiryati, and Basri, 2000; Agrawal et al., 2005; Kong, Tai, and Shin, 2014).

The hand crafted priors may work well on specific tasks, but usually do not generalize well to different types of reflections. And the methods that require multiple images are not practical to use in real scenarios when a user wants to remove the reflections after taking a single photo or from the photo downloaded from the Internet.

Learning based methods. Recent learning based methods learn to separate reflections from data driven approaches. Fan et al. (2017) proposed a deep network with the estimation of the edge map of the background as the additional cue to guide the layer separation. Wan et al. (2018) associate the estimation of the gradient of the background and the separated images in a unified multi-scale framework. Yang et al. (2018) proposed a cascade bidirectional network, which use the predicted background estimate the reflection and then use the predicted reflection to estimate the background. Zhang, Ng, and Chen (2018) proposed to use perceptual loss and exclusion loss to exploit both low-level and high-level information during the separation.

These methods are all based on supervised learning which relies on labeled data. For image reflection removal, it requires image triplets $\{\mathbf{I}, \mathbf{B}, \mathbf{R}\}$, where $\mathbf{B}$ and $\mathbf{R}$ are the background and reflection components of the image $\mathbf{I}$. The image triplets can be collected following the methods introduced by Wan et al. (2017) using a transparent glass and a piece of black sheet. However, it is infeasible to collect large scale dataset for training. Wei et al. (2019) proposed to collect misaligned data as training data, which does not require the camera to be at a fixed position, thus reduces the difficulty of data collection. But the usage of a removable glass is still unavoidable which makes the data collection inconvenient. Therefore, to get sufficient training data, one common practice is to use synthetic data to simulate the reflection in the real-world. The most commonly used data simulation model is the linear combination model (Wan et al., 2018; Yang et al., 2018; Wen et al., 2019) and saturation model (Fan et al., 2017; Zhang, Ng, and Chen, 2018). However, these models may not generalize well in real-world images due to the complexity of the real-world reflections. Ma et al. (2019) proposed a weekly supervised method which is free from the external glass for data collection, while it still needs to collect a group of background and reflection images for training.

Self-supervised learning. Recently, there has been significant interest in self-supervised learning for computer vision, including low-level vision tasks. For example, Pathak et al. (2016) for image inpainting, Laine et al. (2019) for image denosing, Menon et al. (2020) for image super resolution, *etc*. Various frameworks and loss functions are carefully designed to exploit supervision from unlabeled data alone. However, there is no self-supervised learning method for reflection removal so far to our knowledge. One work that is closely related to our proposed method is from Chen et al. (2018), which introduces a self-consistency loss for supervision in image deblurring. It relies on the temporal frames to provide supervision, while our method make use of multi-view information. The self-supervised network is used to finetune existing deblurring neural networks by enforcing that the output, when blurred based on the optical flow between subsequent frames, matches the input blurry image.

5.3 The Proposed Approach

5.3.1 Overview of the Proposed Framework

The overall framework of our proposed method is illustrated in Figure 5.2. The reflection removal network learns a mapping operation on a single image $\mathbf{I} \rightarrow (\mathbf{B}, \mathbf{R})$ and it is a two-stream structure network with one encoder and two decoders sharing the same structure.

Although the network itself takes a single image as input, it takes paired multi-view images (with the same background scene) as training samples during training stage for utilizing the multi-view correspondence as the supervision. We denote the paired training samples as $\{\mathbf{I}_1, \mathbf{I}_2\}$ and the corresponding separation of background and reflection as $\{(\mathbf{B}_1, \mathbf{R}_1), (\mathbf{B}_2, \mathbf{R}_2)\}$. Each pair of images are targeting at the same background, and discrimination between background and reflection relies on the consistency between the background and the inconsistency between the reflection layers of the paired training samples $\{\mathbf{I}_1, \mathbf{I}_2\}$. The inconsistency of reflections can be caused by the movement of objects in the reflection layer and change of camera position. In either case, the consistency and inconsistency provide cues to guide the network to separate apart background and reflection.

Geometry correspondence for multi-view consistency. In general cases, the multi-view pairs $\{\mathbf{I}_1, \mathbf{I}_2\}$ are taken targeting the same background from different views. The corresponding background images $\mathbf{B}_1$ and $\mathbf{B}_2$ contain the same contents but suffer from misalignment due to the different views. For the most real cases, we can assume that the target scenes in the background lie on a (nearly) planar surface in the scene. Then there exists 2D homographs Γ_{ij} to model the transformation from $\mathbf{B}_i$ to $\mathbf{B}_j$ and a corresponding warping operator $\mathcal{T}_i(\cdot)$ that align $\mathcal{T}_i(\mathbf{B}_i)$ with the $\mathbf{B}_j$. Considering that the reflection scene usually lies on the different depth to the background scene, the transformations between $\mathbf{R}_i$ and $\mathbf{R}_j$ are different to Γ_{ij}, resulting in cues for the model to distinguish the elements from the background and reflection (Guo, Cao, and Ma, 2014; Yang et al., 2016). During training, the transformation Γ_{ij} (with the operator $\mathcal{T}_i(\cdot)$) is obtained for each image pair and used to build the multi-view consistency loss functions. To avoid calculating the transformation on the predicted $\mathbf{B}_i$ and $\mathbf{B}_j$ (which causes additional computation in each iteration), by assuming that the background components dominate the observed image, we can obtain Γ_{ij} by estimating on the observed image pairs $\mathbf{I}_i$ and $\mathbf{I}_j$. We employ a RANSAC-based algorithm (Fischler and Bolles, 1981) to estimate the homography matrix Γ.

According to the basic multi-view geometry correspondence, we can design the loss functions to supervise the reflection removal network's learning while the ground truth $\mathbf{B}$ and $\mathbf{R}$ are absent in realistic scenarios. The network learns to separate the background and reflection in a self-supervised manner with the paired data guidance during the training stage. As a byproduct, when the multi-view observations are available in the testing phase (similar to the setting of multi-view reflection removal (Guo, Cao, and Ma, 2014)), the proposed self-supervised learning method can be used to further optimize the performance of the network on specific observations on-the-fly (See Figure 5.6).

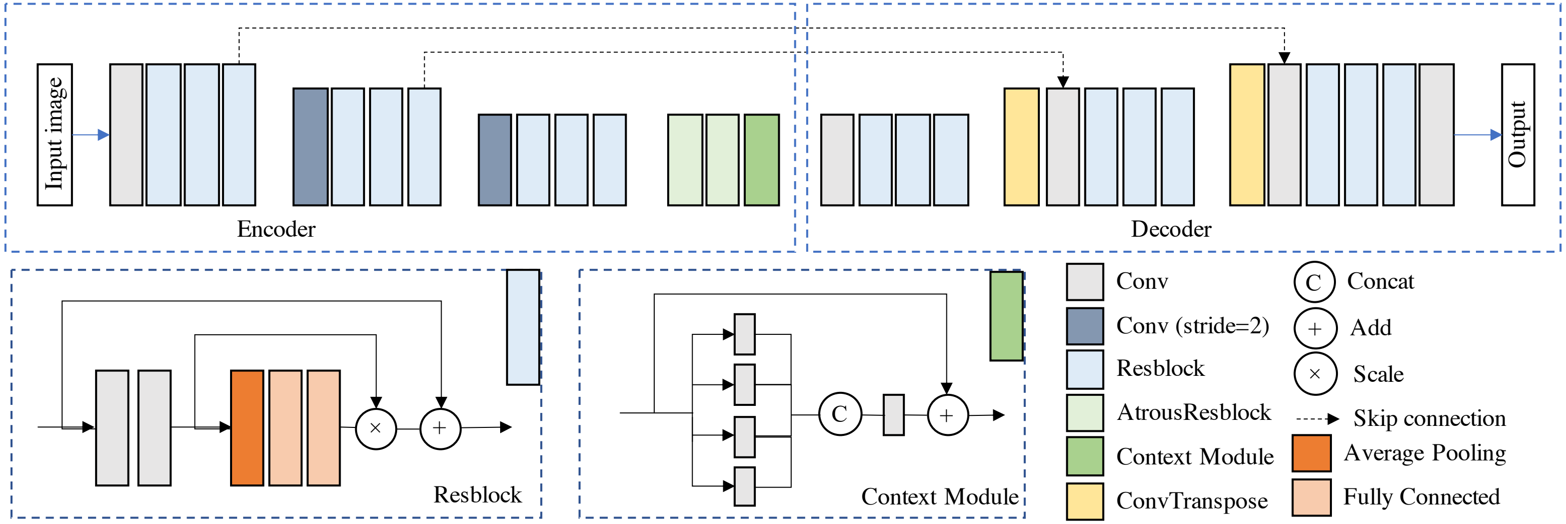


FIGURE 5.3. The proposed network is a two-stream network. The two decoders share the same structure, thus it is illustrated using a single decoder in the figure. The bottom left shows the structure of the resblock with channel attention module and the bottom middle shows the structure of context module.

5.3.2 Network Structure

The architecture of the proposed network is shown in Figure 5.3. It is a two-stream structure with one encoder and two decoders with the same structure. In the encoder, the features are downsampled twice using convolution with $stride = 2$. Afterwards, the decoders reconstruct the image with full resolution via two transposed convolutions. Each downsampling layer of the encoder have skip-connections to the corresponding upsampling layer of all the two decoders, which forms a U-net (Ronneberger, Fischer, and Brox, 2015) based structure. The two decoders estimate the background and reflection images.

Although our framework is able to accommodate different networks, a good design of network is beneficial to the performance. We incorporate the channel attention module (Hu, Shen, and Sun, 2018; Wei et al., 2019) into residual blocks to recalibrate feature maps using global summary statistics. Compared to original residual blocks, this module introduces global contextual information across channels without dramatically increasing the number of parameters. Large receptive field is proven to be effective in reflection removal as it helps to consider more long-range information across the image. To enlarge the receptive field of the network, we employ two atrous residual blocks and a modified version of Atrous Spatial Pyramid Pooling module (ASPP) (Chen et al., 2017) at the end of the encoder to obtain richer features. The ASPP module contains parallel dilated convolutions with different dilated rates and the four dilated rates are set to: 1, 2, 3, 4.

5.3.3 Objective Functions for Self-supervised Learning

The learning of the mapping $\mathbf{I} \rightarrow (\mathbf{B}, \mathbf{R})$ is guided by the following objective functions, with training samples $\{(\mathbf{I}_1, \mathbf{I}_2)\}$ and the estimated background $\mathbf{B} = \{(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2)\}$ and reflection $\mathbf{R} = \{(\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2)\}$. It is fully self-supervised and do not require ground-truth B and R.

Reconstruction loss. For each single observation $\mathbf{I}$, given the estimated background image $\hat{\mathbf{B}}$ and reflection image $\hat{\mathbf{R}}$, we can reconstruct the mixture image as $\hat{\mathbf{I}} = \hat{\mathbf{B}} + \hat{\mathbf{R}}$ and the reconstructed image should be similar to the original input image through a well-trained network. The reconstruction loss is defined as

$$\mathcal{L}_{\text{recon}} = ||\hat{\mathbf{I}}_1 - \mathbf{I}_1||_1 + ||\hat{\mathbf{I}}_2 - \mathbf{I}_2||_1, \tag{5.2}$$

which is a combination of the reconstruction errors on the paired inputs and irrelevant to the multi-view consistency.

Cross-view consistency loss. Since the multi-view images contain the same background, the background images estimated from each input should be consistent with each other. The most straight forward idea to restrict the consistency between the estimated $\hat{\mathbf{B}}_1$ and $\hat{\mathbf{B}}_2$. In order to compare those two backgrounds, we transform the backgrounds to the same plane using homography transformation, as discussed in Section 5.3.1. Since transforming an image to the targeting plane while retaining the shape of the original image may result in some invalid regions, we compute a mask map $\mathbf{M}$ to indicate the valid regions and only involve the valid pixels for computing the consistency loss.

To simplify the model, we calculate the homography transformation to model the correspondence between the observation pairs. The homography model might not strictly apply in all the multi-view real-world image pairs. Therefore, the warped background might not be perfectly aligned with the target background, making simple pixel-wise loss function unsatisfactory. We thus leverage the perceptual-based loss function to encourage the multi-view consistency on the background images and use the masks $\mathbf{M}_1$ and $\mathbf{M}_2$ (for the two images) to exclude the problematic regions out of the loss calculation, which is similar to the loss functions used by Wei et al.

(2019). The cross-view consistency loss can be defined as:

$$\begin{aligned}\mathcal{L}_{\text{cons}} =& ||\phi_h(\mathcal{T}_1(\hat{\mathbf{B}}_1)) - \phi_h(\hat{\mathbf{B}}_2)||_1 \odot \mathbf{M}_1 + \\ & ||\phi_h(\mathcal{T}_2(\hat{\mathbf{B}}_2)) - \phi_h(\hat{\mathbf{B}}_1)||_1 \odot \mathbf{M}_2,\end{aligned} \tag{5.3}$$

where $\mathcal{T}$ is the transformation to warp the background to the other plane, $\mathbf{M}_i$ is the mask maps to select the regions where the warped pixels are valid, and ϕ_h denotes the ‘conv5_2’ feature of the pretrained VGG-19 network.

If the warped region is well aligned with the target, *e.g.* when synthetic data is used to augment the training data, ℓ_1-norm based loss function can also be defined as:

$$\begin{aligned}\mathcal{L}_{\text{cons-pix}} =& ||\mathcal{T}_1(\hat{\mathbf{B}}_1) - \hat{\mathbf{B}}_2||_1 \odot \mathbf{M}_1 + \\ & ||\mathcal{T}_2(\hat{\mathbf{B}}_2) - \hat{\mathbf{B}}_1||_1 \odot \mathbf{M}_2.\end{aligned} \tag{5.4}$$

Mono-view content loss. Although the cross-view consistency loss can properly define the relationship between the ideally-estimated of the multi-view background image pairs, there is still ambiguities in the solution space defined by the above objectives. For example, there exists a trivial solution with $\hat{\mathbf{B}} = \mathbf{0}$ and $\hat{\mathbf{R}} = \mathbf{I}$ (i.e., the elements in $\hat{\mathbf{B}}$ are almost zeros), which can bypass both the reconstruction loss and the cross-view consistency loss. Considering that $\hat{\mathbf{B}}$ should have the content similar to the observation **I**, we apply the mono-view content loss as in the following:

$$\mathcal{L}_{\text{cont-mono}} = ||\hat{\mathbf{B}}_1 - \mathbf{I}_1||_1 + ||\hat{\mathbf{B}}_2 - \mathbf{I}_2||_1, \tag{5.5}$$

which can stabilize the training process. The content loss and the cross-view consistency loss work together and against each other to avoid the trivial solutions.

Cross-view content verification loss. Although we already have multiple losses to define the solution space, there is still ambiguities from the nature of the layer separation task. In practice, we observed that the estimated $\hat{\mathbf{B}}_i$ often contains some residuals from the observation $\mathbf{I}_j$ and the same for $\hat{\mathbf{B}}_j$. To further verify the content in

$\mathbf{B}_i$ by using the observation $\mathbf{I}_j$ according to the multi-view consistency, we introduce the following cross-view content verification loss:

$$\mathcal{L}_{\text{cont-cx}} = \sum_{i \neq j} ||\mathcal{F}(\mathcal{T}_i(\hat{\mathbf{B}}_i) - \mathbf{I}_j)||_1 \odot \mathbf{M}_i \tag{5.6}$$

where

$$\mathcal{F}(\mathbf{X})_k = \begin{cases} x_k, & \text{if } x_k \geq 0 \\ \alpha x_k, & \text{otherwise.} \end{cases} \tag{5.7}$$

In (5.7), $\alpha \in (0, 1)$ is a scalar to decay the influence of value, $\mathcal{F}(\mathbf{X})_k$ denotes the k-th element of $\mathcal{F}(\mathbf{X})$ and x_k denotes the k-th element of $\mathbf{X}$. Here $\mathcal{F}(\cdot)$ could be seen as a LeakyReLU activation on the error map $\mathcal{T}_i(\hat{\mathbf{B}}_i) - \mathbf{I}_j$.

Although the most straightforward way for cross-view content verification is to apply a loss similar to $\mathcal{L}_{\textbf{cont-mono}}$, it may increase the risk of involving the reflection components in $\mathbf{I}_j$ into $\mathbf{B}_i$. Based on the observation that the salient reflection components tend to increase the pixels' intensities, the pixels in $\mathcal{T}_i(\hat{\mathbf{B}}_i)$ with lower intensity than that in $\mathbf{I}_j$ tend to be pixels containing the remaining reflection residuals. On the other hand, the locations in $\mathbf{I}_j$ with $\mathcal{T}_i(\hat{\mathbf{B}}_i)_k < (\mathbf{I}_j)_k$ may contain reflection components with higher probabilities, which may be excluded from the loss functions. We thus use the activation function $\mathcal{F}(\cdot)$ and weight α to decay the influence of the elements with $\mathcal{T}_i(\hat{\mathbf{B}}_i)_k < (\mathbf{I}_j)_k$. We set $\alpha = 0.1$ in all experiments.

Full objective The final objective function is the weighted combination of the aforementioned losses.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{cons}} + \lambda_3 \mathcal{L}_{\text{cont-mono}} + \lambda_4 \mathcal{L}_{\text{cont-cx}}, \tag{5.8}$$

where the weights are empirically set as $\lambda_1 = 15$, $\lambda_2 = 20$, $\lambda_3 = 5$, $\lambda_4 = 1$.

Self-supervised model tuning in testing phase In real-world practice, it is hard to guarantee the testing images always following the distribution of the training images, but it is easy to capture target scenes multiple times from different views. Since the proposed method is free from the ground truth, the model can be used to further optimize the testing performance via tuning the model on the testing multi-view observations directly.

5.3.4 Implementation Details

Our implementation is based on PyTorch. The model is trained for 100 epochs using the Adam optimizer (Kingma and Ba, 2014). The base learning rate is set to 10^{-3} and decays to 10^{-4} after 50 epochs. We use a combination of synthetic and real data as our training dataset.

To synthesize a pair of synthetic images, we randomly sample three different image from PASCAL VOC dataset (Everingham et al., 2010), using one as the background $\mathbf{B}_1$ and two as reflections $\{\mathbf{R}_1, \mathbf{R}_2\}$. The images are randomly cropped to size 256×256. To simulate the different appearance between two views, we generate a homography matrix Γ based on random shift, rotation, shear *etc*, and use it to warp the background to another plane as $\mathbf{B}_2 = \mathbf{B}_1 \circ \Gamma$. Then we use the method of Fan et al. (2017) to generate mixture images $\{\mathbf{I}_1, \mathbf{I}_2\}$ from $\{\mathbf{B}_1, \mathbf{B}_2\}$ and $\{\mathbf{R}_1, \mathbf{R}_2\}$. Figure shows some examples of synthetic data.

The real images are collected from wild scenes using both DSLR camera and smartphone. We collected 50 pairs of images in the wild. For each background, we shoot two images from different angles to capture different reflections. The images are collected under various conditions, *e.g.* sunny days and overcast, outdoor and indoor, *etc*.

5.4 Experiments

5.4.1 Comparison to State-of-the-art Methods

Quantitative evaluations. We compare our self-supervised method against state-of-the-art methods including CEILNet (Fan et al., 2017), BDN (Yang et al., 2018),

	PSNR↑	SSIM↑	NCC↑	E-LPIPS↓
CEILNet	20.84	0.815	0.904	0.0207
BDN	22.01	0.825	0.931	0.0189
Wen19	21.24	0.826	0.891	0.0233
Baseline (sup. & syn.)	20.57	0.828	0.924	0.0194
Ours (self-sup. & syn.)	20.75	0.846	0.924	0.0179
Ours (self-sup. & fusion)	21.60	0.852	0.935	0.0165

TABLE 5.1. Quantitative evaluation results on SIR^2 dataset with the state-of-the-art supervised methods trained on synthetic data.

Zhang18 (Zhang, Ng, and Chen, 2018), ERRNet (Wei et al., 2019) and Wen19 (Wen et al., 2019). The comparison is conducted on the 'Wild' subset of real-world benchmark dataset SIR^2 (Wan et al., 2017). The quality metrics include PSNR, SSIM (Wang et al., 2004), NCC (Wan et al., 2017) and E-LPIPS (Kettunen, Härkönen, and Lehtinen, 2019). Larger values of PSNR, SSIM and NCC indicate better result, while a smaller value of E-LPIPS implies better perceptual similarity based on a pre-trained network.

The methods compared are split into two categories, the ones that only use synthetic data for training and the ones use a fusion of synthetic and real data. The results are shown in Table 5.1 and Table 5.2 respectively. For BDN, we retrain the network with the same data generation method. For Wen19, they have three different models corresponding to different type of reflections, and we pick the one with best performance.

The results in Table 5.1 and Table 5.2 indicate that our self-supervised method is comparable to best performing supervised methods. ERRNet also uses a fusion of synthetic data and real data and the real data they collected have ground truth and are more similar to the scenes in the SIR^2 dataset, thus leads to a better result.

Qualitative comparisons. Figure 5.4 displays visual results on real-world images. It contains image examples from Fan et al. (2017), Zhang, Ng, and Chen (2018) and Wan et al. (2017). It can be seen that our self-supervised method can handle some reflections even better than the state-of-the-art supervised methods, especially for the ones that are trained only on synthetic dataset.

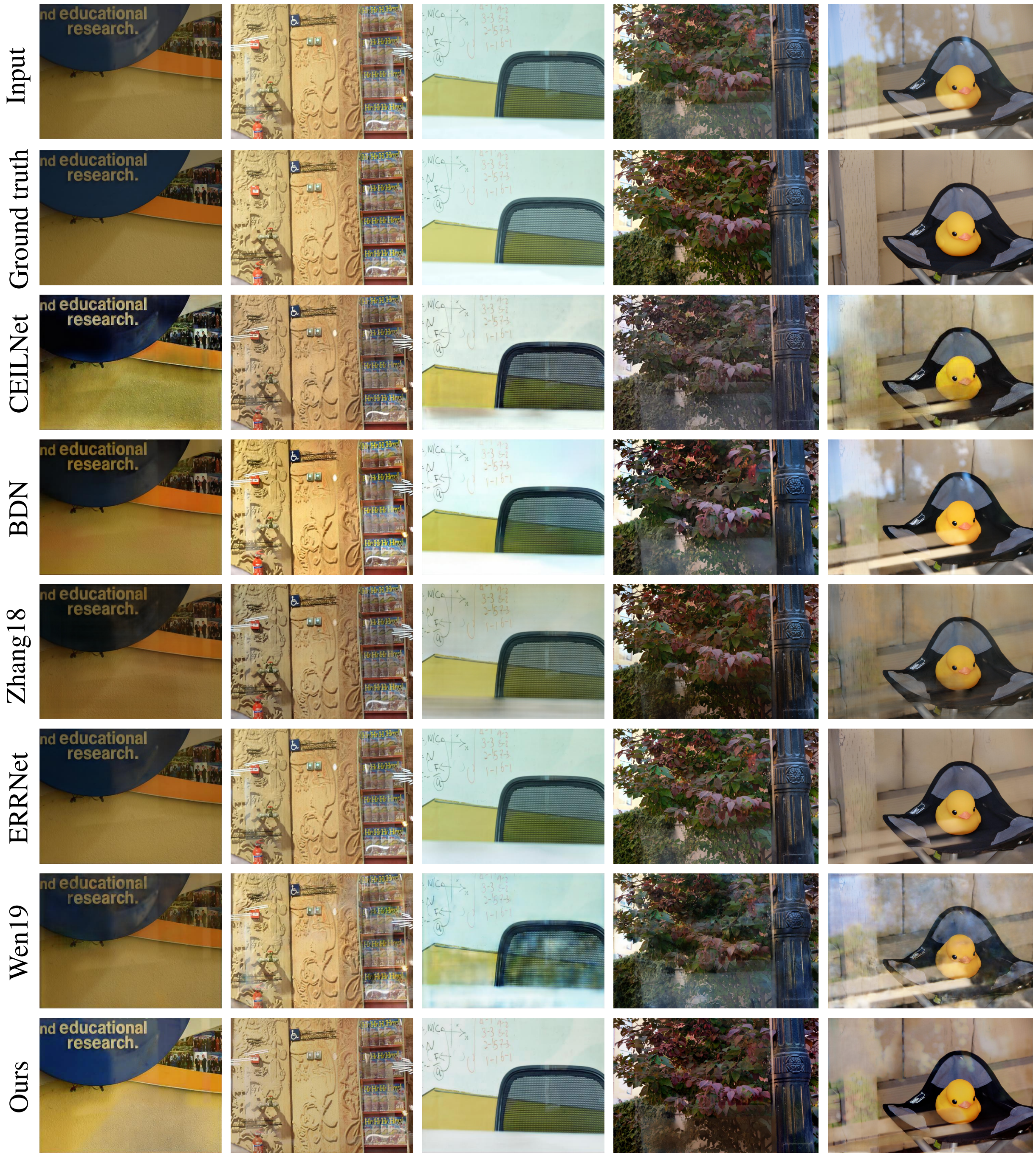


FIGURE 5.4. Examples of the reflection removal results on real-world images. The first three columns are from SIR^2 dataset, and the last two columns are from Zhang, Ng, and Chen (2018). Best viewed with zoom.

	PSNR↑	SSIM↑	NCC↑	E-LPIPS↓
Zhang18	21.11	0.835	0.907	0.0176
ERRNet	23.87	0.854	0.915	0.0137
Ours	21.60	0.852	0.935	0.0165

TABLE 5.2. Quantitative evaluation results on SIR^2 dataset with the state-of-the-art methods trained on a fusion of synthetic and real data.

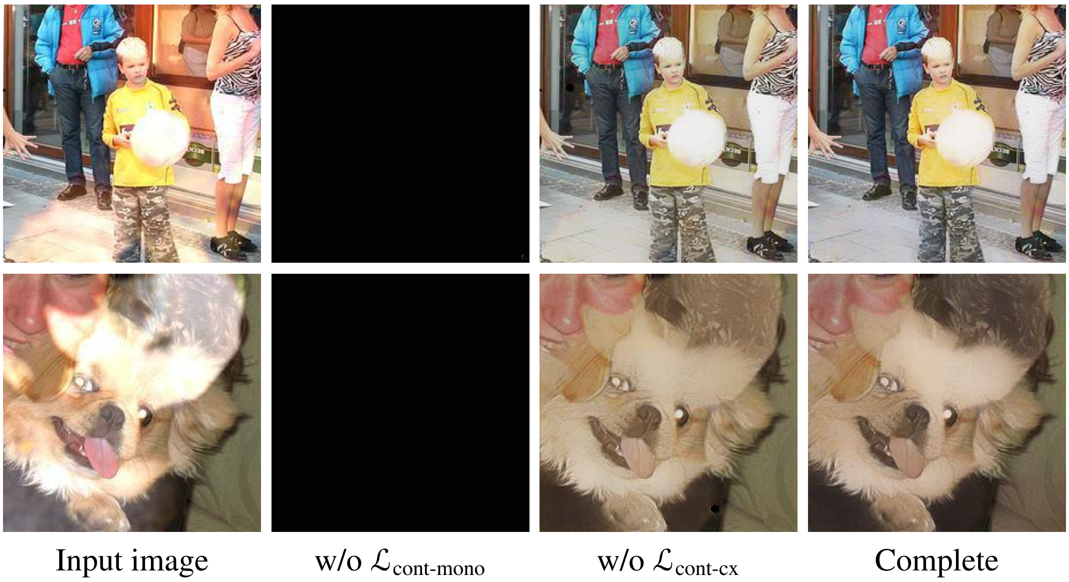

Input image
w/o $\mathcal{L}_{\text{cont-mono}}$
w/o $\mathcal{L}_{\text{cont-cx}}$
Complete

FIGURE 5.5. Ablation study of loss terms.

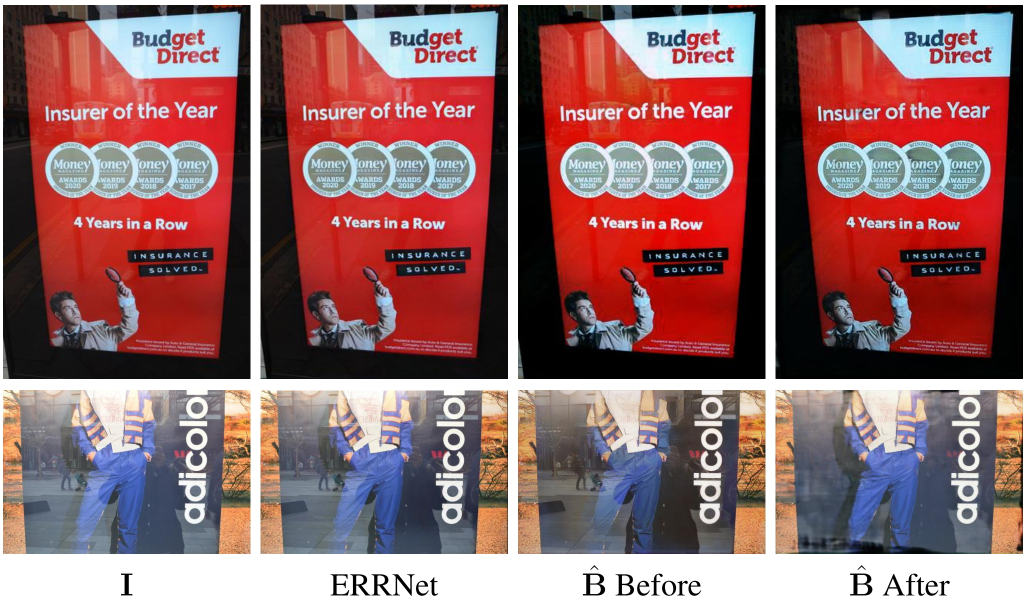


FIGURE 5.6. Examples of using a pair of multi-view images to optimize the network and report the results before and after optimization. Only one view is displayed.

	PSNR↑	SSIM↑	NCC↑	E-LPIPS↓
w/o channel attention	19.05	0.826	0.901	0.0253
w/o $\mathcal{L}_{\text{cont-cx}}$	20.01	0.841	0.917	0.0223
Complete	20.97	0.847	0.927	0.0200

TABLE 5.3. Ablation study for loss terms and network components on synthetic testing dataset.

5.4.2 Ablation Study

We conduct ablation study for our method on 50 synthetic testing images generated using the same generation method as our synthetic training dataset.

Among the proposed loss functions, $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{cons}}$ are the essential constrains to the framework. Figure 5.5 shows that removing $\mathcal{L}_{\text{cont-mono}}$ leads to all black background estimation, which means without $\mathcal{L}_{\text{cont-mono}}$, the network tends to naively treat the original image as the reflection to satisfy the basic constrains. It can be observed from Table 5.3 that introducing $\mathcal{L}_{\text{cont-cx}}$ improves the quality of reflection removal. These results indicate the necessity in including those loss functions. Table 5.3 also indicates that utilizing the channel attention module is beneficial to the performance.

5.4.3 Self-supervised Optimization in Testing Phase

Although our proposed method aims at estimating the reflection-free image from a single image, it can also be used to improve the quality of reflection removal if there are images from multiple views. For each pair of images, we finetune the network with 100 iterations. After optimization, more reflections are removed while the background are better preserved. Figure 5.6 shows some examples of real-world images containing strong reflections which is difficult to handle with previous methods based on single image, exploiting information from another view is able to improve the performance.

5.5 Conclusion

We introduce a self-supervised reflection removal approach which only relies on the supervision from the geometry correspondence and consistency between the multi-view reflection images. Specifically, we use the multi-view reflection image pairs to train a deep network with consistency losses that are effective and robust to utilize the imperfect cues derived from the multi-view consistency. The trained network achieves state-of-the-art performance in handling the real-world reflection images without supervision from the ground truth. Moreover, the proposed method can to further optimize the network in the testing phase while given the multi-view images for a scene.

Chapter 6

Conclusion

Contents

This chapter summarizes the contents covered in this thesis and discusses possible future works.

6.1 Summary of the Thesis

In this thesis, we mainly discussed two challenging problems in low-level vision, *i.e.* image deblurring and reflection removal, using deep learning based approaches. Addressing these problems will enhance the image quality and improves the performance of relevant computer vision systems accepting these images as input. Our proposed methods on these issues contribute to existing approaches and made them the more completed and practical solution to real-world applications. This framework is free from the dependance on ground truth labels and achieves better generalization on real images.

In chapter 3, we present a flexible and efficient deep learning based method for the estimation and removal of the heterogeneous motion blur. We represent the heterogeneous motion blur as pixel-wise linear motion blur, which is able to represent an extensive range of possible motions. And we propose a fully convolutional network to estimate the dense motion flow map and use automatically generated data to train the neural network. The proposed method outperforms the state-of-the-art image deblurring method. Since our blur model is heterogeneous, it is more capable to keep the non-blurry part of the image sharp after deblurring. The range of motion flow is pretty large in our data simulation, as a result, our trained network is able to handle a wider range of motion blur compared to previous work, especially some strong motion blur that is difficult to process. However, it still requires additional deconvolution process to recover latent sharp image from estimated motion flow and this process is very time-consuming. And since there is a lack of multi-scale components in our network, it may not perform very well when the area of blur is either too small or too big compared to the training data.

In chapter 4 and 5, we have addressed the image reflection removal problem using supervised and self-supervised methods respectively. In chapter 4, we present a cascade neural network for supervised single image reflection removal. The network utilizes the estimation of reflection to boost the recovery of the background. We show that the correlation between reflection and background is essential to the good recovery of the two layers. By considering the recovery of reflections, we achieve a significant performance boost in the estimation of background image. Since there is few method considering the recovery of the reflections, and different reflection model may result in different definition of reflection components in an image, it is difficult to compare different methods regarding reflection recovery. In chapter 5 we propose a self-supervised learning framework for reflection removal. The framework leverages the supervision from the geometry correspondence and consistency between the multi-view reflection image pairs. Instead of designing various method to approximate the real reflections through simulation, either manually or using a neural network, or through a rendering engine, our framework is free from dependence on the reflection-free target images. However, the self-supervised method is not as efficient compared to supervised learning as in some cases the supervision is weak due to the degradation of image, such as occlusions *etc*, which introduces noise to the training process.

6.2 Future Work

We have presented some solutions to the targeted problems, but the some issues remain unsolved and there are still room for improvement in the performance of existing solutions, and there are various directions to extend the current work. We illustrate some future directions for these two problems.

Current learning-based reflection removal approaches mainly focus on single image solutions. Due to the complexity and ambiguity of reflections, it is sometimes difficult to address the ill-posedness of the problem. Zhou et al. (2019) proposed to work on binocular stereo images for image deblurring. Considering the latest smartphones usually have multiple cameras, methods based on multiple images becomes practical. Although we have exploited the multi-view information for self-supervised training, the network still works on single image. In our method, the alignment of images from two views may be inaccurate. Using binocular stereo images, instead of arbitrary multi-view images, may provide more reliable geometric correspondence between the two images. It is also possible to use multiple images with different apertures for reflection removal.

Blur and reflections may only occur in certain part of the image, while the remaining part of the image is sharp and dominated by background layer. According to our observation, the methods that are more capable of handling strong reflections is likely to oversmooth the regions without reflections. The attention mechanism might be useful in attend to the regions according the level of blur and reflection.

Recently, transformer network proposed by Vaswani et al. (2017), which was originally designed for machine translation, has obtained promising results in the field of computer vision as well. It is worth investigating its application to computer photography problems.

Bibliography

Agrawal, Amit, Ramesh Raskar, Shree K Nayar, and Yuanzhen Li (2005). “Removing Photography Artifacts Using Gradient Projection and Flash-Exposure Sampling”. In: *ACM Trans. on Graphics* 24.3, pp. 828–835.

Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik (2011). “Contour Detection and Hierarchical Image Segmentation”. In: *IEEE Trans. on PAMI* 33.5, pp. 898–916.

Arvanitopoulos, Nikolaos, Radhakrishna Achanta, and Sabine Süsstrunk (2017). “Single Image Reflection Suppression.” In: *CVPR*. IEEE, pp. 1752–1760.

Bar, Leah, Nahum Kiryati, and Nir Sochen (2006). “Image deblurring in the presence of impulsive noise”. In: *IJCV* 70.3, pp. 279–298.

Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester (2000). “Image inpainting”. In: *SIGGRAPH*. ACM, pp. 417–424.

Brusius, Florian, Ulrich Schwanecke, and Peter Barth (2011). “Blind Image Deconvolution of Linear Motion Blur”. In: *VISIGRAPP*. Springer, pp. 105–119.

Buades, Antoni, Bartomeu Coll, and J-M Morel (2005). “A non-local algorithm for image denoising”. In: *CVPR*. Vol. 2. IEEE, pp. 60–65.

Chakrabarti, Ayan (2016). “A Neural Approach to Blind Motion Deblurring”. In: *ECCV*.

Chakrabarti, Ayan, Todd Zickler, and William T Freeman (2010). “Analyzing Spatially-Varying Blur”. In: *CVPR*. IEEE, pp. 2512–2519.

Chan, Tony F and Chiu-Kwong Wong (1998). “Total Variation Blind Deconvolution”. In: *IEEE Trans. on Image Processing* 7.3, pp. 370–375.

Chen, Huaijin, Jinwei Gu, Orazio Gallo, Ming-Yu Liu, Ashok Veeraraghavan, and Jan Kautz (2018). “Reblur2deblur: Deblurring videos via self-supervised learning”. In: *ICCP*. IEEE, pp. 1–9.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille (2017). “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Trans. on PAMI* 40.4, pp. 834–848.

Cho, Sunghyun and Seungyong Lee (2009). “Fast Motion Deblurring”. In: *SIGGRAPH ASIA*.

Couzinie-Devy, Florent, Jian Sun, Karteek Alahari, and Jean Ponce (2013). “Learning to Estimate and Remove Non-Uniform Image Blur”. In: *CVPR*. IEEE, pp. 1075–1082.

Dai, Shengyang and Ying Wu (2009). “Removing Partial Blur in a Single Image”. In: *CVPR*. IEEE, pp. 2544–2551.

Debevec, Paul E and Jitendra Malik (2008). “Recovering high dynamic range radiance maps from photographs”. In: *SIGGRAPH classes*, pp. 1–10.

Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang (2015). “Image super-resolution using deep convolutional networks”. In: *IEEE Trans. on PAMI* 38.2, pp. 295–307.

Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2010). “The Pascal Visual Object Classes (Voc) Challenge”. In: *IJCV* 88.2, pp. 303–338.

Fan, Qingnan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David P Wipf (2017). “A Generic Deep Architecture for Single Image Reflection Removal and Image Smoothing.” In: *ICCV*, pp. 3258–3267.

Farid, Hany and Edward H Adelson (1999). “Separating Reflections and Lighting Using Independent Components Analysis”. In: *CVPR*. Vol. 1. IEEE, pp. 262–267.

Fergus, Rob, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman (2006). “Removing Camera Shake from a Single Photograph”. In: *ACM Trans. on Graphics*.

Fischler, Martin A and Robert C Bolles (1981). “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6, pp. 381–395.

Freeman, William T, Thouis R Jones, and Egon C Pasztor (2002). “Example-based super-resolution”. In: *IEEE Computer graphics and Applications* 22.2, pp. 56–65.

Gai, Kun, Zhenwei Shi, and Changshui Zhang (2012). “Blind Separation of Superimposed Moving Images Using Image Statistics”. In: *IEEE Trans. on PAMI* 34.1, pp. 19–32.

Glasner, Daniel, Shai Bagon, and Michal Irani (2009). “Super-resolution from a single image”. In: *ICCV*. IEEE, pp. 349–356.

Gong, Dong, Mingkui Tan, Yanning Zhang, Anton van den Hengel, and Qinfeng Shi (2016). “Blind Image Deconvolution by Automatic Gradient Activation”. In: *CVPR*. IEEE.

Gong, Dong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi (2017). “From Motion Blur to Motion Flow: A Deep Learning Solution for Removing Heterogeneous Motion Blur”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Gong, Dong, Zhen Zhang, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, and Yanning Zhang (2018). “Learning an Optimizer for Image Deconvolution”. In: *arXiv preprint arXiv:1804.03368*. arXiv: 1804.03368.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). “Generative Adversarial Nets”. In: *NIPS*, pp. 2672–2680.

Granados, Miguel, Boris Ajdin, Michael Wand, Christian Theobalt, Hans-Peter Seidel, and Hendrik PA Lensch (2010). “Optimal HDR reconstruction with linear digital cameras”. In: *CVPR*. IEEE, pp. 215–222.

Gu, Shuhang, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng (2014). “Weighted nuclear norm minimization with application to image denoising”. In: *CVPR*. IEEE, pp. 2862–2869.

Guo, Xiaojie, Xiaochun Cao, and Yi Ma (2014). “Robust Separation of Reflection from Multiple Images”. In: *CVPR*. IEEE, pp. 2187–2194.

Gupta, Ankit, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless (2010). “Single Image Deblurring Using Motion Density Functions”. In: *ECCV*. IEEE, pp. 171–184.

Han, Byeong-Ju and Jae-Young Sim (2017). “Reflection Removal Using Low-Rank Matrix Completion”. In: *CVPR*. Vol. 2. IEEE.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification”. In: *ICCV*. IEEE, pp. 1026–1034.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *CVPR*, pp. 770–778.

Hirsch, Michael, Suvrit Sra, Bernhard Schölkopf, and Stefan Harmeling (2010). “Efficient Filter Flow for Space-Variant Multiframe Blind Deconvolution.” In: *CVPR*. Vol. 1. IEEE, p. 2.

Hirsch, Michael, Christian J Schuler, Stefan Harmeling, and Bernhard Schölkopf (2011). “Fast Removal of Non-Uniform Camera Shake”. In: *ICCV*. IEEE.

Hu, Jie, Li Shen, and Gang Sun (2018). “Squeeze-and-excitation networks”. In: *CVPR*. IEEE, pp. 7132–7141.

Hu, Zhe, Li Xu, and Ming-Hsuan Yang (2014). “Joint Depth Estimation and Camera Shake Removal from Single Blurry Image”. In: *CVPR*. IEEE, pp. 2893–2900.

Hyun Kim, Tae, Byeongjoo Ahn, and Kyoung Mu Lee (2013). “Dynamic Scene Deblurring”. In: *CVPR*. IEEE, pp. 3160–3167.

Ioffe, Sergey and Christian Szegedy (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *ICML*, pp. 448–456.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2017). “Image-to-Image Translation with Conditional Adversarial Networks”. In: *CVPR*. IEEE, pp. 5967–5976.

Jia, Jiaya (2007). “Single image motion deblurring using transparency”. In: *CVPR*. IEEE, pp. 1–8.

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell (2014). “Caffe: Convolutional Architecture for Fast Feature Embedding”. In: *arXiv*.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). “Perceptual losses for real-time style transfer and super-resolution”. In: *ECCV*. IEEE, pp. 694–711.

Kettunen, Markus, Erik Härkönen, and Jaakko Lehtinen (2019). “E-LPIPS: Robust Perceptual Image Similarity via Random Transformation Ensembles”. In: *arXiv preprint arXiv:1906.03973*.

Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee (2016). “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *CVPR*. IEEE.

Kim, Soomin, Yuchi Huo, and Sung-Eui Yoon (2020). “Single Image Reflection Removal With Physically-Based Training Images”. In: *CVPR*. IEEE, pp. 5164–5173.

Kim, Tae Hyun and Kyoung Mu Lee (2014). “Segmentation-Free Dynamic Scene Deblurring”. In: *CVPR*. IEEE.

Kingma, Diederik and Jimmy Ba (2014). “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980*. arXiv: 1412.6980.

Kong, Naejin, Yu-Wing Tai, and Joseph S Shin (2014). “A Physically-Based Approach to Reflection Separation: From Physical Modeling to Constrained Optimization”. In: *IEEE Trans. on PAMI* 36.2, pp. 209–221.

Krishnan, Dilip and Rob Fergus (2009). "Fast image deconvolution using hyper-Laplacian priors". In: *NIPS*, pp. 1033–1041.

Krishnan, Dilip, Terence Tay, and Rob Fergus (2011). "Blind Deconvolution Using a Normalized Sparsity Measure". In: *CVPR*. IEEE, pp. 233–240.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet Classification with Deep Convolutional Neural Networks". In: *NIPS*, pp. 1097–1105.

Kupyn, Orest, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas (2018). "Deblurgan: Blind motion deblurring using conditional adversarial networks". In: *CVPR*. IEEE, pp. 8183–8192.

Laine, Samuli, Tero Karras, Jaakko Lehtinen, and Timo Aila (2019). "High-Quality Self-Supervised Deep Image Denoising". In: *NIPS*. Vol. 32.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. (2017). "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." In: *CVPR*. Vol. 2. IEEE, p. 4.

Lehtinen, Jaakko, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila (2018). "Noise2Noise: Learning Image Restoration without Clean Data". In: *ICML*. Vol. 80, pp. 2965–2974.

Lettry, Louis, Kenneth Vanhoey, and Luc van Gool (2018). "DARN: A Deep Adversarial Residual Network for Intrinsic Image Decomposition". In: *WACV*. IEEE, pp. 1359–1367.

Levin, A, A Zomet, and Y Weiss (2004). "Separating Reflections from a Single Image Using Local Features". In: *CVPR*. Vol. 1. IEEE, pp. 306–313.

Levin, Anat (2006). "Blind Motion Deblurring Using Image Statistics". In: *NIPS*, pp. 841–848.

Levin, Anat and Yair Weiss (2007). “User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior”. In: *IEEE Trans. on PAMI* 29.9.

Levin, Anat, Assaf Zomet, and Yair Weiss (2003). “Learning to Perceive Transparency from the Statistics of Natural Scenes”. In: *NIPS*, pp. 1271–1278.

Levin, Anat, Yair Weiss, Fredo Durand, and William T Freeman (2009). “Understanding and Evaluating Blind Deconvolution Algorithms”. In: *CVPR*. IEEE, pp. 1964–1971.

Levin, Anat, Yair Weiss, Fredo Durand, and William T Freeman (2011). “Efficient Marginal Likelihood Optimization in Blind Deconvolution”. In: *CVPR*. IEEE, pp. 2657–2664.

Li, Yu and Michael S Brown (2013). “Exploiting Reflection Change for Automatic Reflection Removal”. In: *ICCV*, pp. 2432–2439.

Li, Yu and Michael S Brown (2014). “Single Image Layer Separation Using Relative Smoothness”. In: *CVPR*. IEEE, pp. 2752–2759.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). “Microsoft Coco: Common Objects in Context”. In: *ECCV*, pp. 740–755.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). “Fully Convolutional Networks for Semantic Segmentation”. In: *CVPR*. IEEE, pp. 3431–3440.

Ma, Daiqian, Renjie Wan, Boxin Shi, Alex C. Kot, and Ling-Yu Duan (2019). “Learning to Jointly Generate and Separate Reflections”. In: *ICCV*. IEEE.

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). “Rectifier nonlinearities improve neural network acoustic models”. In: *ICML*. Vol. 30. 1, p. 3.

Mairal, Julien, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman (2009). “Non-local sparse models for image restoration”. In: *ICCV*. IEEE, pp. 2272–2279.

Mao, Xiaojiao, Chunhua Shen, and Yu-Bin Yang (2016). “Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections”. In: *NIPS*, pp. 2802–2810.

Menon, Sachit, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin (2020). “PULSE: Self-supervised photo upsampling via latent space exploration of generative models”. In: *CVPR*. IEEE, pp. 2437–2445.

Nah, Seungjun, Tae Hyun Kim, and Kyoung Mu Lee (2017). “Deep multi-scale convolutional neural network for dynamic scene deblurring”. In: *CVPR*. IEEE, pp. 3883–3891.

Nair, Vinod and Geoffrey E Hinton (2010). “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *ICML*, pp. 807–814.

Pan, Jinshan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang (2014). “Deblurring Text Images via L_0-Regularized Intensity and Gradient Prior”. In: *CVPR*. IEEE, pp. 2901–2908.

Pan, Jinshan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang (2016a). “Blind Image Deblurring Using Dark Channel Prior”. In: *CVPR*. IEEE.

Pan, Jinshan, Zhe Hu, Zhixun Su, Hsin-Ying Lee, and Ming-Hsuan Yang (2016b). “Soft-Segmentation Guided Object Motion Deblurring”. In: *CVPR*.

Pathak, Deepak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros (2016). “Context encoders: Feature learning by inpainting”. In: *CVPR*. IEEE, pp. 2536–2544.

Perrone, Daniele and Paolo Favaro (2014). “Total Variation Blind Deconvolution: The Devil Is in the Details”. In: *CVPR*. IEEE, pp. 2909–2916.

Reinhard, Erik, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski (2010). *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann.

Ren, Wenqi, Jiawei Zhang, Lin Ma, Jinshan Pan, Xiaochun Cao, Wangmeng Zuo, Wei Liu, and Ming-Hsuan Yang (2018). “Deep non-blind deconvolution via generalized low-rank approximation”. In: *NIPS*, pp. 297–307.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *MICCAI*. Springer, pp. 234–241.

Rudin, Leonid I, Stanley Osher, and Emad Fatemi (1992). “Nonlinear total variation based noise removal algorithms”. In: *Physica D: nonlinear phenomena* 60.1-4, pp. 259–268.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). “Imagenet large scale visual recognition challenge”. In: *IJCV* 115.3, pp. 211–252.

Sarel, Bernard and Michal Irani (2004). “Separating Transparent Layers through Layer Information Exchange”. In: *ECCV*. IEEE, pp. 328–341.

Schechner, Yoav Y, Nahum Kiryati, and Ronen Basri (2000). “Separation of Transparent Layers Using Focus”. In: *IJCV* 39.1, pp. 25–39.

Schechner, Yoav Y, Joseph Shamir, and Nahum Kiryati (2000). “Polarization and Statistical Analysis of Scenes Containing a Semireflector”. In: *JOSA A* 17.2, pp. 276–284.

Schmidt, Uwe, Carsten Rother, Sebastian Nowozin, Jeremy Jancsary, and Stefan Roth (2013). “Discriminative Non-Blind Deblurring”. In: *CVPR*, pp. 604–611.

Schuler, Christian J, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf (2013). “A Machine Learning Approach for Non-Blind Image Deconvolution”. In: *CVPR*. IEEE, pp. 1067–1074.

Schuler, Christian J, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf (2015). “Learning to deblur”. In: *IEEE Trans. on PAMI* 38.7, pp. 1439–1451.

Shih, YiChang, Dilip Krishnan, Fredo Durand, and William T Freeman (2015). “Reflection Removal Using Ghosting Cues”. In: *CVPR*. IEEE, pp. 3193–3201.

Shrivastava, Ashish, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb (2017). “Learning from Simulated and Unsupervised Images through Adversarial Training”. In: *CVPR*. Vol. 2. IEEE, p. 5.

Simonyan, Karen and Andrew Zisserman (2015). “Very deep convolutional networks for large-scale image recognition”. In:

Sinha, Sudipta N, Johannes Kopf, Michael Goesele, Daniel Scharstein, and Richard Szeliski (2012). “Image-Based Rendering for Scenes with Reflections”. In: *ACM Trans. on Graphics* 31.4, pp. 100–1.

Sun, Jian, Wenfei Cao, Zongben Xu, and Jean Ponce (2015). “Learning a Convolutional Neural Network for Non-Uniform Motion Blur Removal”. In: *CVPR*. IEEE, pp. 769–777.

Sun, Libin, Sunghyun Cho, Jue Wang, and James Hays (2013). “Edge-Based Blur Kernel Estimation Using Patch Priors”. In: *ICCP*, pp. 1–8.

Szeliski, Richard, Shai Avidan, and P Anandan (2000). “Layer Extraction from Multiple Images Containing Reflections and Transparency”. In: *CVPR*. Vol. 1. IEEE, pp. 246–253.

Tai, Yu-Wing, Ping Tan, and Michael S Brown (2011). “Richardson-Lucy Deblurring for Scenes under a Projective Motion Path”. In: *IEEE Trans. on PAMI* 33.8, pp. 1603–1618.

Tao, Xin, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia (2018). “Scale-recurrent network for deep image deblurring”. In: *CVPR*, pp. 8174–8182.

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). “Instance normalization: The missing ingredient for fast stylization”. In: *arXiv preprint arXiv:1607.08022*.

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2018). “Deep image prior”. In: *CVPR*. IEEE, pp. 9446–9454.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *NIPS*.

Walker, Jacob, Abhinav Gupta, and Martial Hebert (2015). “Dense Optical Flow Prediction from a Static Image”. In: *ICCV*, pp. 2443–2451.

Wan, Renjie, Boxin Shi, Tan Ah Hwee, and Alex C Kot (2016). “Depth of Field Guided Reflection Removal”. In: *ICIP*. IEEE, pp. 21–25.

Wan, Renjie, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot (2017). "Benchmarking Single-Image Reflection Removal Algorithms". In: *ICCV*. IEEE, pp. 3942–3950.

Wan, Renjie, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot (2018). "CRRN: Multi-Scale Guided Concurrent Reflection Removal Network". In: *CVPR*. IEEE, pp. 4777–4785.

Wang, Xiaolong, David Fouhey, and Abhinav Gupta (2015). "Designing Deep Networks for Surface Normal Estimation". In: *CVPR*. IEEE, pp. 539–547.

Wang, Xiaolong and Abhinav Gupta (2016). "Generative Image Modeling Using Style and Structure Adversarial Networks". In: *ECCV*. IEEE.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004). "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Trans. on Image Processing* 13.4, pp. 600–612.

Wei, Kaixuan, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang (2019). "Single Image Reflection Removal Exploiting Misaligned Training Data and Network Enhancements". In: *CVPR*. IEEE, pp. 8178–8187.

Wen, Qiang, Yinjie Tan, Jing Qin, Wenxi Liu, Guoqiang Han, and Shengfeng He (2019). "Single Image Reflection Removal Beyond Linearity". In: *CVPR*. IEEE.

Whyte, Oliver, Josef Sivic, Andrew Zisserman, and Jean Ponce (2012). "Non-Uniform Deblurring for Shaken Images". In: *IJCV* 98.2, pp. 168–186.

Wieschollek, P., B. Schölkopf, H. P. A. Lensch, and M. Hirsch (2016). "End-to-End Learning for Image Burst Deblurring". In: *ACCV*.

Xu, Li and Jiaya Jia (2010). "Two-Phase Kernel Estimation for Robust Motion Deblurring". In: *ECCV*. IEEE, pp. 157–170.

Xu, Li, Shicheng Zheng, and Jiaya Jia (2013). "Unnatural L0 Sparse Representation for Natural Image Deblurring". In: *CVPR*, pp. 1107–1114.

Xu, Li, Jimmy SJ Ren, Ce Liu, and Jiaya Jia (2014). "Deep Convolutional Neural Network for Image Deconvolution". In: *NIPS*, pp. 1790–1798.

Xue, Tianfan, Michael Rubinstein, Ce Liu, and William T Freeman (2015). “A Computational Approach for Obstruction-Free Photography”. In: *ACM Trans. on Graphics* 34.4, p. 79.

Yang, Jianchao, John Wright, Thomas S Huang, and Yi Ma (2010). “Image super-resolution via sparse representation”. In: *IEEE Trans. on Image Processing* 19.11, pp. 2861–2873.

Yang, Jiaolong, Hongdong Li, Yuchao Dai, and Robby T Tan (2016). “Robust Optical Flow Estimation of Double-Layer Images under Transparency or Reflection”. In: *CVPR*. IEEE, pp. 1410–1419.

Yang, Jie, Dong Gong, Lingqiao Liu, and Qinfeng Shi (2018). “Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal”. In: *ECCV*. IEEE, pp. 654–669.

Zhang, Haichao and David Wipf (2013). “Non-Uniform Camera Shake Removal Using a Spatially-Adaptive Sparse Penalty”. In: *NIPS*, pp. 1556–1564.

Zhang, Haichao, David Wipf, and Yanning Zhang (2013). “Multi-Image Blind Deblurring Using a Coupled Adaptive Sparse Prior”. In: *CVPR*. IEEE.

Zhang, Jiawei, Jinshan Pan, Wei-Sheng Lai, Rynson WH Lau, and Ming-Hsuan Yang (2017a). “Learning fully convolutional networks for iterative non-blind deconvolution”. In: *CVPR*. IEEE, pp. 3817–3825.

Zhang, Jiawei, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang (2018). “Dynamic scene deblurring using spatially variant recurrent neural networks”. In: *CVPR*. IEEE, pp. 2521–2529.

Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang (2017b). “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising”. In: *IEEE Trans. on Image Processing* 26.7, pp. 3142–3155.

Zhang, Kai, Wangmeng Zuo, Shuhang Gu, and Lei Zhang (2017c). “Learning deep CNN denoiser prior for image restoration”. In: *CVPR*. IEEE, pp. 3929–3938.

Zhang, Xuaner, Ren Ng, and Qifeng Chen (2018). “Single Image Reflection Separation with Perceptual Losses”. In: *CVPR*. IEEE.

Zheng, Shicheng, Li Xu, and Jiaya Jia (2013). “Forward Motion Deblurring”. In: *CVPR*. IEEE, pp. 1465–1472.

Zhou, Shangchen, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy S Ren (2019). “Davanet: Stereo deblurring with view aggregation”. In: *CVPR*. IEEE, pp. 10996–11005.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A Efros (2017). “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *ICCV*. IEEE, pp. 2242–2251.

Zoran, Daniel and Yair Weiss (2011). “From Learning Models of Natural Image Patches to Whole Image Restoration”. In: *ICCV*. IEEE, pp. 479–486.