TOWARDS PRACTICAL DEEP LEARNING BASED IMAGE RESTORATION MODEL

BY

YUQIAN ZHOU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Mark A. Hasegawa-Johnson, Chair
Professor Zhi-Pei Liang
Assistant Professor Suma Bhat
Adjunct Assistant Professor Humphrey Shi

ABSTRACT

Image Restoration (IR) is a task of reconstructing the latent image from its degraded observations. It has become an important research area in computer vision and image processing, and has wide applications in the imaging industry. Conventional methods apply inverse filtering or optimization-based approaches to restore images corrupted in ideal cases. The limited restoration performance on ill-posed problems and the low-efficient iterative optimization processes prevents such algorithms from being deployed to more complicated industry applications. Recently, the advanced deep Convolutional Neural Networks (CNNs) begin to model the image restoration as learning and inferring the posterior probability in a regression model, and successfully achieved remarkable performance. However, due to the data-driven nature, the models trained with simple synthetic paired data (*e.g,* bicubic interpolation or Gaussian noises) cannot be well adapted to more complicated inputs from real data domains. Besides, acquiring real paired data for training such models is also very challenging.

In this dissertation, we discuss the data manipulation and model adaptability of the deep learning based image restoration tasks. Specifically, we study improving the model adaptability by understanding the domain difference between its training data and its expected testing data. We argue that the cause of image degradation can be various due to multiple imaging and transmission pipelines. Though complicated to analyze, for some specific imaging problems, we can still improve the performance of deep restoration models on unseen testing data by resolving the data domain differences implied in the image acquisition and formation pipeline. Our analysis focuses on *digital image denoising*, *image restoration from more complicated degradation types beyond denoising* and *multi-image inpainting*. For all tasks, the proposed training or adaptation strategies, based on the physical principle of the degradation formation or based on geometric assumption of the image, achieve a

reasonable improvement on the restoration performance.

For image denoising, we discuss the influence of the Bayer pattern of the Camera Filter Array (CFA) and the image demosaicing process on the adaptability of the deep denoising models. Specifically, for the task of denoising RAW sensor observations, we find that unifying and augmenting the data Bayer pattern during training and testing is an efficient strategy to make the well-trained denoising model Bayer-invariant. Additionally, for the RGB image denoising, demosaicing the noisy RAW images with Bayer patterns will result in the spatial-correlation of pixel noises. Therefore, we propose the pixel-shuffle down-sampling approach to break down this spatial correlation, and make the Gaussian-trained denoiser more adaptive to real RGB noisy images.

Beyond denoising, we explain a more complicated degradation process involving diffraction when there are some occlusions on the imaging lens. One example is a novel imaging model called Under-Display Camera (UDC). From the perspective of optical analysis, we study the physics-based imaging processing method by deriving the forward model of the degradation, and synthesize the paired data for both conventional and deep denoising pipeline. Experiments demonstrate the effectiveness of the forward model and the deep restoration model trained with synthetic data achieves visually similar performance to the one trained with real paired images.

Last, we further discuss reference-based image inpainting to restore the missing regions in the target image by reusing contents from the source image. Due to the color and spatial misalignment between the two images, we first initialize the warping by using multi-homography registration, and then propose a content-preserving Color and Spatial Transformer (CST) to refine the misalignment and color difference. We designed the CST to be scale-robust, so it mitigates the warping problems when the model is applied to testing images with different resolution. We synthesize realistic data while training the CST, and it suggests the inpainting pipeline achieves a more robust restoration performance with the proposed CST.

To my parents and Leigh, for their love and support.

ACKNOWLEDGMENTS

I will always remember what Prof. Thomas Huang said to us, "Every single student is my best student". I am feeling honored to be one of his students. Tom brought me to the U.S., held my hand and guided me to the world of computer vision and image processing. His kindness and humility have always encouraged me to be an honest, humble, and ambitious researcher. I would like to express my deepest respect and sincere thoughts to our beloved Prof. Huang and Margaret Huang.

Great thanks to my thesis advisor Prof. Mark Hasegawa-Johnson. He gave me great support during the hardest time, and enlightened impressive research ideas. This dissertation will become impossible without his patient help. I will also extend my thanks to the doctoral committee: Prof. Zhi-Pei Liang, Prof. Suma Bhat and Prof. Humphrey Shi, for their critically important advice to improve the work.

I enjoyed working with my colleagues from the Image Formation and Processing (IFP) family. They are talented, kind and greatly helpful: Dr. Yuchen Fan, Dr. Kuangxiao Gu, Dr. Wei Han, Dr. Jianbo Jiao, Dr. Zengming Shen, Dr. Zhiqiang Shen, Dr. Zilong Huang, Prof. Xinchao Wang, Prof. Zhangyang Wang, Prof. Yunchao Wei, Dr. Ning Xu, Dr. Jiahui Yu, Haichao Yu, Hanchao Yu, Xiaolin Zhang, Bowen Cheng, Yang Fu, Zhonghao Wang, Jiachen Li, Xingqian Xu, Haoming Lu, Mang Tik Chiu, Rui Qian, Jiachen Li, and many others.

I would like to thank my mentors and colleagues during my internship: Dr. Jue Wang, Dr. Xue Bai, Dr. Haibin Huang, Dr. Kai Li, Jiaming Liu, Tong He, and Yang Wang from Megvii Research; Tim Large, Dr. Sehoon Lim, Dr. Neil Emerton, and Dr. David Ren from Microsoft Applied Science Group; Dr. Connelly Barnes, Dr. Eli Shechtman, Dr. Zhe Lin, and Sohrab Amirghodsi from Adobe. They gave me great support and made contributions to my publications. I feel lucky to have a chance working with all of them.

I am thankful for receiving grant support from the Thomas and Margaret Huang Research Award, and the recommendation from Prof. Florin Dolcos. I learned cross-disciplinary knowledge while collaborating the neuroscience project with Paul Bogdan.

Many thanks to my family and Leigh for their love and support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The image restoration problem has been well studied for decades, and is becoming a more important research area in low-level and physics-based vision. Common image restoration tasks include image denoising, deblurring, super-resolution, and inpainting, etc. Suppose we model the degradation process as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{x}$ is the latent clean image, $\mathbf{y}$ is the corrupted image with degradation, $\mathbf{H}$ is the degradation matrix, and $\mathbf{n}$ is the additive noise function. For example, in an image denoising task, $\mathbf{H}$ is an identity matrix, and $\mathbf{n}$ is the real complicated noise model in practice. For a more complicated degradation model, $\mathbf{H}$ becomes the matrix representing the degradation operator. For image inpainting tasks, $\mathbf{H}$ is the occlusion mask removing pixels from the images. The goal of the image restoration task is to estimate the latent clean or complete image by observing the corrupted input. Due to its ill-posed nature, image restoration becomes challenging, especially when the image degradation is severe and complicated.

Conventionally, image restoration tasks can be addressed by image filtering, or modeled by a Maximum Likelihood (ML) or a Maximum a Posteriori (MAP) problem. Recently, embracing the large-scale training data and powerful representation of Deep Neural Networks (DNNs), deep learning-based image restoration models have achieved very competitive performance on most image restoration applications such as de-noising [1, 2, 3, 4, 5, 6], de-blurring [7, 8], de-raining [9, 10], de-hazing [11, 12], super-resolution [13, 14], light-enhancement [15] , and inpainting. A well-trained deep model forcibly learns the mapping from a specific set of the corrupted images $\mathbf{y}$ to the degradation-free or complete image $\mathbf{x}$. The advantages of these methods are fast during inference without iterative optimization steps, and high performance resulting

from flexible non-linear transformation of the deep network layers. However, they also suffer from *poor adaptation and generalization ability* issues and *data hungry bottleneck*. Specifically, most existing deep models are trained on paired data with the degraded images corrupted by simple Gaussian noise, bi-cubic down-sampling, or manually-defined blur kernels. Such models cannot be successfully applied to real inputs with complicated or combined degradation factors. Models trained with paired data $(\hat{y}, x)$ can not be easily used to test on inputs y from another domain relatively different from the training domain.

In this thesis, we focus on addressing three problems deep image restoration models face while being applied to practical real-world problems: training-testing domain shift, real complicated training data insufficiency, and training-testing image scale difference. Specifically, we present three types of image restoration tasks: RAW and RGB image denoising, image restoration with complicated degradation beyond denoising, and reference-based image inpainting with real color and spatial difference between the target and source images.

The contributions of this thesis focus on the specific strategies designed for specific tasks:

- First, for RAW image denoising, we study the influence of Bayer patterns of the Camera Filter Array (CFA) on the model adaptability. For a real-collected dataset containing RAW images with a single or different Bayer pattern, we propose Bayer Unification (BayerUnify) and Bayer Augmentation (BayerAug) strategies to train the network. Applying the adaptor greatly improves the denoising performance on unseen noisy data.

- Second, for real RGB image denoising, we analyze demosaicing and argue that the demosaicing process interpolates the pixels to make the image noisy patterns spatially-correlated. Therefore, we propose to break down the spatial correlation using a pixel-shuffle down-sampling method. Experiments show the effectiveness of the strategy by verifying the performance improvement of a model trained with pixel-independent Gaussian noises.

- Third, for a more complicated degradation type related to diffraction

effects caused by the occlusions on the imaging lens, we propose a physics-based image formation approach based on optical analysis to synthesize training data. The synthesized Point Spread Function (PSF) yields reasonable restoration performance using a traditional deconvolution pipeline. The model trained with paired synthesized data also achieves comparable visual restoration to the model trained with real paired data.

- Fourth, for another type of image restoration task named image inpainting, we introduce a reference-based image inpainting problem. Given a target image and a source image capturing a similar scene, we restore and complete the target image by reusing the contents from the source image. To address the real spatial and color differences between them, we propose a pipeline consisting of multi-homography registration and a deep Color-Spatial Transformation (CST) module. To adapt the CST to practical inputs with multiple scales, complicated color differences and spatial misalignment, we design the deep models to be scale-robust, and synthesize the training dataset with a diversity of color and spatial differences. The model then works well on real user images.

In this thesis, we organize the contents in the following ways. In chapter 2, we present the RAW image denoising networks and the proposed Bayer pattern-related adaptation methods. In chapter 3, the pixel-shuffle down-sampling method is introduced to adapt an AWGN-trained denoiser to real RGB noises. In chapter 4, a physics-based image processing approach is presented for more complidated degradation related to diffraction caused by lens occlusions. In chapter 5, we present Transfill, a reference-based image inpainting model addressing real color and spatial difference between the source and target images. Finally, conclusions are drawn in chapter 7. We will discuss the limitations of the current methods and future work of each task.

CHAPTER 2

BAYER PATTERN MANIPULATION FOR REAL RAW IMAGE DENOISING

In this chapter, we mainly discuss the training-testing data domain mismatching problem related to Bayer patterns for real RAW image denoising tasks. We present Bayer Unification (BayerUnify) and Bayer-preserving Augmentation (BayerAug) strategies to train the networks adaptive to testing images with arbitrary Bayer patterns.

2.1 Introduction

Image denoising is one of the fundamental problems in image processing and computer vision, and restoring high quality images from extremely noisy ones remains challenging. This can be even worse when it comes to images taken from mobile devices. Due to the use of relatively low-cost sensors and lenses, images captured by mobile cameras can be severely corrupted by high level noise, especially in low-light scenarios. Many denoising methods have been proposed to address this problem, including traditional methods such as NLM [16] and BM3D [17] as well as more recent deep neural network (DNN) based denoising models [18, 19, 2, 20, 21, 22, 23], but their performances are still far from satisfactory on mobile devices.

Recently, thanks to public noisy image datasets [24, 15, 25], denoising RAW image data and real sRGB data has received more and more attention and has shown promising results [15, 26, 27]. Specifically, RAW images are direct readings from images sensors, with camera filter arrays (CFAs) arranged in specific patterns such as the Bayer pattern [28]. These digital signals are further post-processed to obtain RGB images through a complex pipeline including lens shading correction, white balancing, demosaicing, gamma correction, etc. [29]. Therefore, original noise properties that exist in RAW images are often distorted in RGB images, making the noise harder

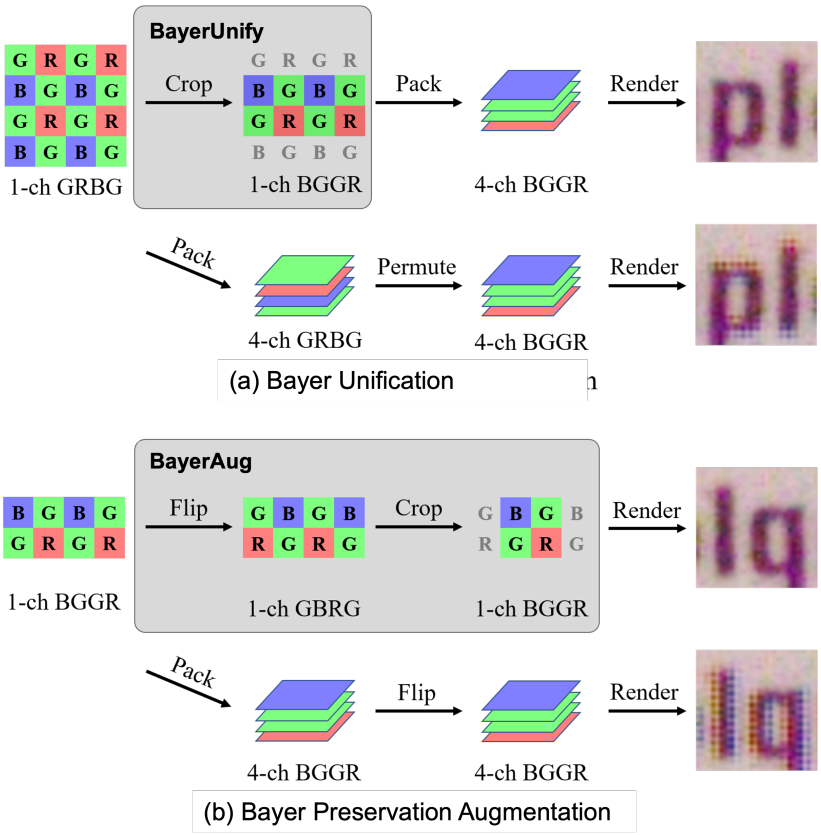


Figure 2.1: Demonstration of the proposed Bayer pattern unification (a) and Bayer preserving augmentation (b). The shown method converts and augments the Bayer RAW images without affecting the content, while improper unification or augmentation would disturb the spatial relationship of the RAW images and therefore result in artifacts.

to remove afterwards. This means that there are potentially better denoising methods that can be developed on the RAW image data [15], compared with many works done in the RGB domain.

In this chapter, we mainly study the problem of RAW image denoising using Deep Neural Networks (DNNs). We propose to use a UNet-based structure and review and explore the performance of DNNs on image denoising task. For RAW image denoising, we introduce a novel data unification and data augmentation method to efficiently train the network.

2.2 Bayer Pattern Unification and Augmentation

Background In this section, we introduce the intuitive idea and methods of Bayer pattern unification and augmentation specifically designed for RAW image denoising networks. To perform RAW image denoising with DNNs, it is a common practice to pack a Bayer raw image into a 4-channel RGGB image, and feed it into neural networks [15]. With data collected from cameras with different Bayer patterns, a simple solution is to train one model for each pattern. However, this decreases the size of the effective training set and thereby hurts the performance. To fully utilize all training data to achieve better performance, we introduce a Bayer pattern unification (BayerUnify) technique to eliminate the differences among Bayer patterns. Flipping and cropping operations are employed to turn a specific CFA pattern into another one, with which I can convert all training images into the same pattern. As a result, all the training data can be used together to optimize a single model to achieve the best possible result.

Data augmentation is also a common approach in deep learning to improve model performance by increasing the diversity of a training dataset. However, data augmentation of RAW images is not as straightforward as that of RGB images. An example is shown in Fig. 2.1 (b). Simply flipping the packed 4-channel RAW images is erroneous because it results in an image that is impossible in real world. This phenomenon can also be found in other types of augmentation operations such as cropping, transposition, . To tackle this problem, we introduce a Bayer preserving augmentation (BayerAug) technique that allows proper augmentation for raw images. As shown in Fig. 2.1 (b), extra operations are required to correctly flip a raw image.

Both BayerUnify and BayerAug techniques are simple, yet effective ways for increasing the training data size and diversity for raw image denoising. We apply these techniques to train models based on our modified U-Net [30].

Bayer Pattern Unification (BayerUnify) The Bayer patterns of RAW images fall into different categories. To apply a single CNN to denoise raw images with different Bayer patterns, it is essential to align the order of the channels since different channels capture different regions of wavelength. In the meantime, the structural information laid in adjacent pixels from different channels has to be maintained. Based on these principles, we utilize multiple

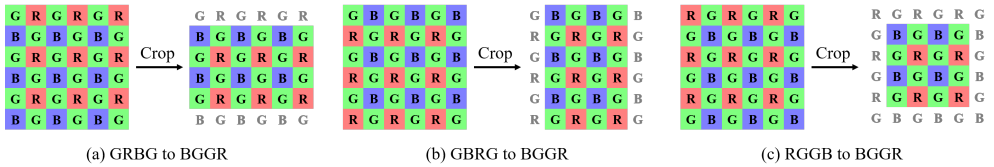


Figure 2.2: Unify Bayer pattern via cropping in the training phase.

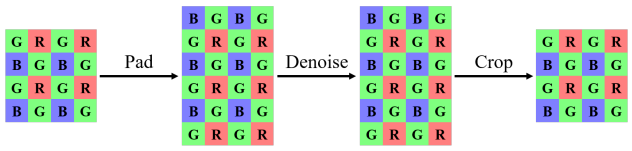


Figure 2.3: Unify Bayer pattern via padding and de-unify via cropping in the testing phase.

ways to convert a RAW image from one Bayer pattern to another, which are applicable to different scenarios.

In the training stage, we unify raw images with different Bayer patterns via cropping. By scarifying a minor number of pixels, it enables us to use RAW images from different cameras to train a single denoising model, and thereby increases the number of available training samples.

Suppose we represent each pattern by the sequence of its channels within each 2×2 block, in the order of top-left, top-right, bottom-left, and bottom-right. Typically, there are four possible formats, namely RGGB, BGGR, GRBG, and GBRG. For clarity, I use BGGR as the target format to illustrate the method.

Cropping an odd number of rows or columns creates offsets which alter the Bayer pattern. As shown in Fig. 2.2, cropping the first row and the last row changes an $C_1C_2C_3C_4$ image into a $C_3C_4C_1C_2$ image (e.g. GRBG to BGGR). Likewise, cropping the first and the last column alters $C_1C_2C_3C_4$ into $C_2C_1C_4C_3$ (e.g. GBRG to BGGR). These two operations together convert $C_1C_2C_3C_4$ into $C_4C_3C_2C_1$ (e.g. RGGB to BGGR). Hence, one can convert any Bayer pattern to another by cropping.

It has shown that one can train a Bayer-pattern-specific network with RAW images of different patterns. Moreover, it is possible to denoise images of different patterns with the trained network. Due to the fact that every pixel of the input images needs to be processed, instead of cropping some pixels

from the input images, we unify their Bayer patterns via padding some pixels. After network denoising, we simply remove these extra pixels to convert the output images. This process is illustrated in Fig. 2.3.

Padding alters the Bayer pattern in a similar way to cropping. Padding one row of pixels to the top and the bottom changes an $C_1C_2C_3C_4$ image into a $C_3C_4C_1C_2$ image (e.g. GRBG to BGGR); padding one column to the left and to the right turns $C_1C_2C_3C_4$ into $C_2C_1C_4C_3$ (e.g. GBRG to BGGR); padding to all four edges converts $C_1C_2C_3C_4$ into $C_4C_3C_2C_1$ (e.g. RGGB to BGGR).

Hence, we can apply padding to unify any pattern to the desired one. As a straightforward de-unification, removing the padded pixels reverses the conversion. Note that I apply reflection padding to make sure the additional pixels come from the correct channel.

Bayer Preserving Augmentation (BayerAug) When training a neural network for vision and graphic tasks on RGB images, it is common to apply flipping and cropping as data augmentation methods. Flipping and cropping increase the effective number of samples dramatically while being very concise. However, for Bayer raw images, flipping operations may affect the Bayer pattern. As illustrated in Fig. 2.4 (a) and (b), a horizontal flip switches the Bayer pattern from $C_1C_2C_3C_4$ to $C_2C_1C_4C_3$, and a vertical one switches the pattern from $C_1C_2C_3C_4$ to $C_3C_4C_1C_2$.

Therefore, we combine both flipping and cropping to perform data augmentation while preserving the Bayer pattern of the image. After flipping an image, we apply cropping to reverse the change of Bayer pattern. We illustrate this process in Fig. 2.4 (c).

As another type of flipping, a transposition has different effects on different patterns, depending on the channels of the diagonal components. Generally, the transpose of an $C_1C_2C_3C_4$ image would be in the pattern of $C_1C_3C_2C_4$. For a RGrGbB input, its transpose would be in RGbGrB, which is roughly the same format (assuming the different between Gr channel and Gb channel is subtle). However, for a GRBG input, its transpose would be in GBRG, a totally different pattern. For this reason, we can safely perform transposition to augment RGGB and BGGR images, but not in GRBG or GBRG.

Training with patches instead of the entire images is another common trick used in model training. Different from the cropping operations in BayerUnify,

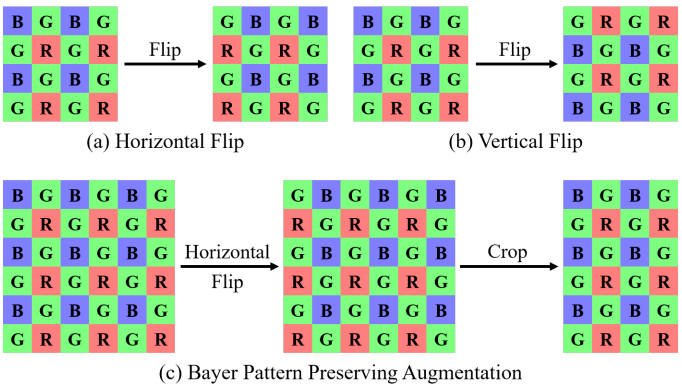


Figure 2.4: An example of Bayer preserving augmentation. Since flipping RAW image may affect its Bayer pattern, we first perform a horizontal flipping (BGGR→GBRG), and then crop its first and last column (GBRG→BGGR) to obtain a horizontally flipped BGGR image from a BGGR image.

Set	Scene	# GRBG	# BGGR	# RGGB	Total
Train	2-10	40	126	98	264
Test	1	30	16	10	56

Table 2.1: The train/test split of the SIDD dataset.

to correctly obtain patches from the entire Bayer raw image without changing its Bayer pattern, we need to avoid any offset. This could be done by simply cropping even numbers of rows (columns).

With combinations of the discussed three flipping methods and one cropping method, one will be able to perform data augmentation on Bayer RAW images without any flaw. Note that they can be applied on both homogeneous datasets [15] and heterogeneous datasets [24], enhancing the generalizability of the obtained model.

2.3 Experiments on RAW Image Denoising

The method is evaluated on the Smartphone Image Denoising Dataset (SIDD) [24]. The statistics of the Bayer pattern and scene number are shown in Table 2.1. Its training set contains 320 pairs of noise-free images and noisy images, which cover three different Bayer patterns and 10 different scenes. Its

validation set and testing set consist of 40 pairs of image from eight different scenes. Both RAW images and sRGB images are available.

A modified U-Net [30] architecture is used in the experiments. The U-Net structure contains four-level downsampling layer blocks, and for each block, there are two convolutional layers. As proposed by [15], we packed the raw images into four channels as the network input. Differently, we trained the networks to produce four-channel outputs, and unpacked them to obtain denoised raw images. In the experiments, all the networks were trained with L_1 loss and AdamW optimizer [31] with initial learning rate of $2e-4$ and weight decay of $2e-5$. Patch size and mini-batch size were set to 512 and 4 respectively. Each model is trained for $200,000$ iterations, and the learning rate is divided by 10 on plateaus. We detected the plateaus and selected the best models using the PSNR scores on the scene 1 patches of the official validation set. For testing, the entire images are fed into the network.

Ablation Study The ablation study on the proposed strategy in terms of PSNR is illustrated in Table 2.2. As the baseline, we trained one network for each Bayer pattern. Since the number of samples available for training each model is insufficient, this method resulted in a limited performance. We also compared it with NaïveNorm method, which is directly permuting the order of the packed 4-channel input. To evaluate, we ran a model with training and testing data converted (to BGGR) with this method. Compared to BayerUnify, this method obtains a lower performance, which shows the importance of our valid raw data unification method. Besides, in the NaïveAug method, it is plausible to flip the packed 4-channel images as I do to 3-channel RGB images. However, flpping a 4-channel image disarrays the spatial signal and generates images that are very different from the original dataset. We validated this augmentation method based on the correctly unified dataset (BayerUnify).

Another network was trained with the proposed Bayer pattern unification. In the training phase, we applied cropping to unify all 264 training pairs to BGGR format. In the testing phase, we employed padding to unifying and cropping to convert the test cases. Thanks to the increase of training samples, the method outperforms the previous baselines.

We further trained a network with both Bayer pattern unification and Bayer preserving augmentation. In the training phase, after pattern unification, we

augmented the data via flipping and cropping. The result shows that the data augmentation boosted the generalization of the obtained model, and consequently improved its performance on the unseen scene.

Table 2.2: PSNR of different unification and augmentation methods. As shown, normalizing and augmenting raw images in problematic methods (NaïveUnify and NaïveAug) result in degradation of the network performance.

Method	GRBG	BGGR	RGGB
GRBG Only	43.46	-	-
BGGR Only	-	49.50	-
RGGB Only	-	-	51.59
NaïveUnify	42.78	49.74	51.83
BayerUnify	**43.92**	**49.88**	**51.85**
BayerUnify+NaïveAug	43.83	49.76	51.81
BayerUnify+BayerAug	**44.02**	**49.92**	**51.95**

2.4 Conclusions

In this chapter, we present the influence of Bayer pattern on the training of RAW image denoising models. We proposed effective data pre-processing and augmentation methods specifically designed for Bayer RAW images, namely BayerUnify and BayerAug. By applying them, the model trained with images belonging to the unified Bayer patterns is adapted to testing inputs with arbitrary Bayer patterns, and the model itself is better generalized to unseen testing data by training on more data of different Bayer patterns.

CHAPTER 3

PIXEL-SHUFFLE DOWNSAMPLING: APPLYING AWGN-TRAINED DENOISER TO REAL RGB DENOISING

In this chapter, we mainly discuss the real RGB image blind denoising. We analyze the influence of demosaicing on spatial-correlated patterns in the RGB noisy images and propose an adapter to break down the spatial correlation. Specifically, we propose a pixel-shuffle down-sampling adaptation strategy for a well-trained Gaussian Denoiser to be applied to spatial-correlated real RGB noises.

3.1 Introduction

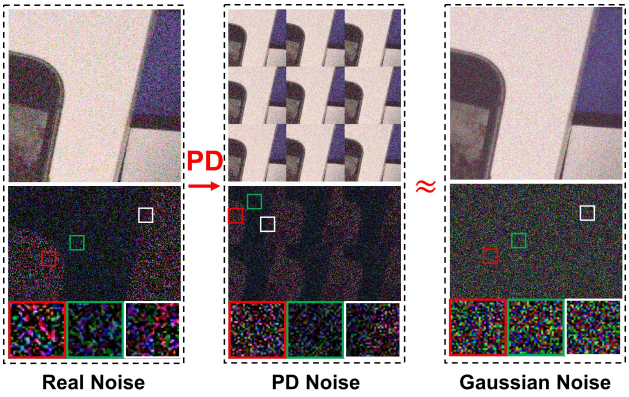


Figure 3.1: Basic idea of the proposed adaptation method: Pixel-shuffle Down-sampling (PD). Spatially-correlated real noise (left) is broken into spatially-variant pixel-independent noise (middle) to approximate spatially-variant Gaussian noise (right). Then an AWGN-based denoiser can be applied to such real noise accordingly.

As a fundamental task in image processing and computer vision, image denoising has been extensively explored in the past several decades even for downstream applications [32, 33]. Traditional methods including the ones

based on image filtering [34], low rank approximation [35, 36], sparse coding [37], and image prior [38] have achieved satisfactory results on synthetic noise such as Additive White Gaussian Noise (AWGN). Recently, deep CNN has been applied to this task, and discriminative-learning-based methods such as DnCNN [1] outperform most traditional methods on AWGN denoising.

Unfortunately, while these learning-based methods work well on the same type of synthetic noise that they are trained on, their performance degrades rapidly on real images, showing poor generalization ability in real world applications. This indicates that these data-driven denoising models are highly domain-specific and non-flexible to transfer to other noise types beyond AWGN. To improve model flexibility, the recently-proposed FFDNet [5] trains a conditional non-blind denoiser with a manually adjusted noise-level map. By giving high-valued uniform maps to FFDNet, only over-smoothed results can be obtained in real image denoising. Therefore, blind denoising of real images is still very challenging due to the lack of accurate modeling of real noise distribution. These unknown real-world noises are much more complex than pixel-independent AWGN. They can be spatially-variant, spatially-correlated, signal-dependent, and even device-dependent.

To better address the problem of real image denoising, current attempts can be roughly divided into the following categories: (1) realistic noise modeling [39, 40, 6], (2) noise profiling such as multi-scale [41, 36], multi-channel [35] and regional based [42] settings, and (3) data augmentation techniques such as the adversarial-learning-based ones [43]. Among them, CBDNet [39] achieves good performance by modeling the realistic noise using the in-camera pipeline model proposed in [44]. It also trains an explicit noise estimator and sets a larger penalty for under-estimated noise. The network is trained on both synthetic and real noises, but it still cannot fully characterize real noises. Brooks et al. [40] used prior statistics stored in the RAW data of DND to augment the synthetic RGB data, but it does not prove the generalization of the model on other real noises.

In this chapter, from a novel viewpoint of real image blind denoising, we seek to adapt a learning-based denoiser trained on pixel-independent synthetic noises to unknown real noises. As shown in Figure 3.1, we assume that real noises differ from pixel-independent synthetic noises dominantly in *spatial/channel-variance and correlation* [45]. This difference results from in-camera pipeline like demosaicing [3]. Based on this assumption, we first

propose to train a basis denoising network using mixed AWGN and RVIN. Our flexible basis net consists of an explicit noise estimator folloId by a conditional denoiser. We demonstrate that these fully-convolutional nets are actually efficient in coping with pixel-independent spatially/channel-variant noises. Second, we propose a simple yet effective adaptation strategy, Pixel-shuffle Down-sampling (PD), which employs the divide-and-conquer idea to handle real noises by breaking down the spatial correlation.

In summary, the main contributions include:

- We propose a new flexible deep denoising model (trained with AWGN and RVIN) for both blind and non-blind image denoising. We also demonstrate that such fully convolutional models trained on spatially-invariant noises can handle *spatially-variant noises*.
- We adapt the AWGN-RVIN-trained deep denoiser to real noises by applying a novel strategy called Pixel-shuffle Down-sampling (PD). *Spatially-correlated noises* are broken down to *pixel-wise independent noises*. We examine and overcome the proposed domain gap to boost real denoising performance.
- The proposed method achieves state-of-the-art performance on DND benchmark and other real noisy RGB images among models trained only with synthetic noises. Note that our model does not use any images or prior meta-data from real noise datasets. We also show that with the proposed PD strategy, the performance of some other existing denoising models can also be boosted.

3.2 Related Work

3.2.1 Deep Learning Based Image Restoration Model

Adopting deep-CNNs for image restoration has shown evident improvements by embracing their representative power. In the early work, Vincent et al [46] proposed to use stacked auto-encoder for image denoising. Later, ARCNN was introduced by Dong et al. [47] for compression artifacts reduction. Zhang et al. [1] proposed DnCNN for image denosing, which uses advanced techniques

like residual learning and batch normalization to boost performance. In IRCNN [48], a learned set of CNNs are used as denoising prior for other image restoration tasks. For image super resolution, extensive efforts have been spent into designing advanced architectures and learning methods, such as progressive super resolution [49], residual [13] and dense connection [50], back-projection [51], scale-invariant convolution, [52] and channel attention [53]. Recently, most state-of-the-art approaches [54, 55, 56] incorporate non-local attention into networks to further boost representation ability. Although extensive efforts have been made in architectural engineering, existing methods relying on convolution and non-local operation can only exploit information at a same scale.

3.2.2 Real Data Acquisition and Realistic Data Synthesis

Real-world restoration[6, 57] is becoming a new concept in low-level vision. In the past decades, low-level vision works on synthetic data (denoising on AWGN and SR on Bicubic), but the models are not efficient for images with real degradation such as real noises or arbitrary blur kernels. Making models perform better on real-world inputs usually requires new problem analysis and a more challenging data collection. In the previous literature, there have been two common ways to prepare adaptive training data for real-world problems: real data collection and near-realistic data synthesis.

Recently, more real noise datasets such as DND [58], SIDD [59, 8], and RENOIR [25], have been proposed to address practical denoising problems. Abdelrahman et al. [6] proposed to estimate ground truth from captured smartphone noise images, and utilized the paired data to train and evaluate the real denoising algorithms. In addition to noises, Chen et al. first proposed the SID dataset [15] to resolve extreme low-light imaging. In the area of Single Image Super Resolution (SISR), researchers considered collecting optical zoom data [57, 60] to learn better computational zoom. Other restoration problems including reflection removal [61, 62] also follow the trend of real data acquisition. Collecting real data suffers from limitation of scene variety since most previous models acquire images of postcards, static objects, or color boards.

A realistic dataset can be synthesized if the degradation model is fully

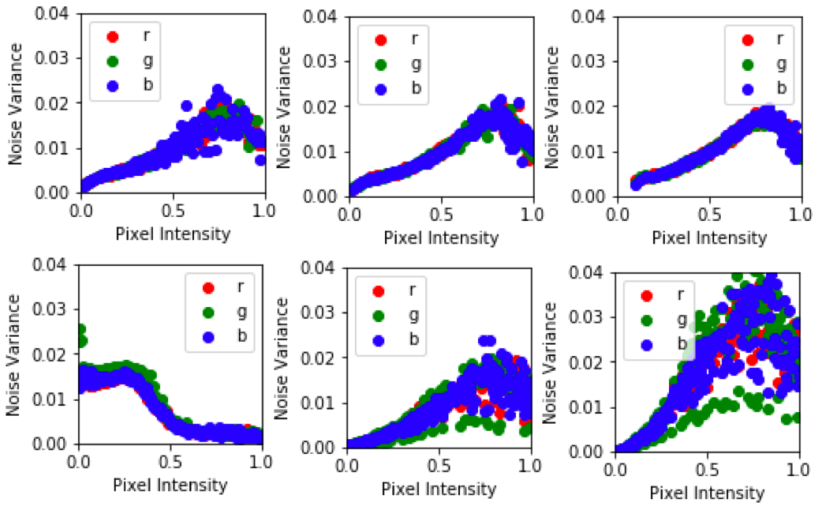


Figure 3.2: Noise Level Function (NLFs) (noise variance as a function of image intensity) before (first row) and after (second row) Gamma transform and demosaicing. Gamma factor is 0.39, 1.38, and 2.31 from the left to right column.

understood and resolved. One good practice of data synthesis is generating real noises on raw sensors or RGB images. CBDNet [63] and Tim et al. [40] synthesized real noises by unfolding the in-camera pipeline, and Abdelhamed et al. [64] better fitted the real noise distribution with flow-based generative models. Other physics-based synthesis was also explored in blur [65] or hazing [66]. In this chapter, we adapt the AWGN-RVIN noises into real RGB noises by analyzing the demosacing process.

3.3 Baseline Model and Structures

Basis Noise Model The basis noise model is mixed AWGN-RVIN. Noises in sRGB images are no longer approximated Gaussian-Poisson Noises as in the raw sensor data mainly due to gamma transform, demosaicing, and other interpolations etc. In Figure 3.2, we follow [44] pipeline to synthesize noisy images, and plot the Noise Level Functions (NLFs) (noise variance as a function of image intensity) before (first row) and after (second row) the Gamma Correction transform and demosaicing. From left to right, the Gamma factor increases. It shows that in RGB images, clipping effects, and other non-linear transforms will greatly influence the originally linear

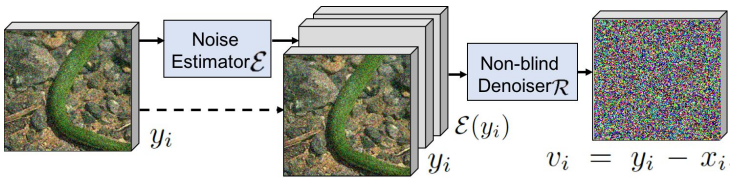


Figure 3.3: Structure of the proposed blind denoising model. It consists of a noise estimator $\mathcal{E}$ and a follow-up non-blind denoiser $\mathcal{R}$. The model aims to jointly learn the image residual.

noise variance-intensity relationship in raw sensor data, even changing the noise mean. Though complicated, for a more general case than Gaussian-Poisson noises of modeling different nonlinear transforms, real noises in RGB can still be locally approximated as AWGN [5, 67, 68]. In this chapter, we thus assume the RGB noises to be approximated as spatially-variant and spatially-correlated AWGN.

Adding RVIN for training aims at explicitly resolving the defective pixels caused by dead pixels of camera hardware or long exposure frequently appearing in most night-shot images. We generate AWGN, RVIN, and mixed AWGN-RVIN following PGB[69].

Basis Model Structure The architecture of the proposed basis model is illustrated in Figure 3.3. The proposed blind denoising model $\mathcal{G}$ consists of a noise estimator $\mathcal{E}$ and a follow-up non-blind denoiser $\mathcal{R}$. Given a noisy observation $y_i = \mathcal{F}(x_i)$, where $\mathcal{F}$ is the noise synthetic process, and x_i is the noise-free image, the model aims to jointly learn the residual $\mathcal{G}(y_i) \approx v_i = y_i - x_i$, and it is trained on paired synthetic data (y_i, v_i). Specifically, the noise estimator outputs $\mathcal{E}(y_i)$ consisting of six pixel-wise noise-level maps that correspond to two noise types, i.e., AWGN and RVIN, across three channels (R, G, B). Then y_i is concatenated with the estimated noise level maps $\mathcal{E}(y_i)$ and fed into the non-blind denoiser $\mathcal{R}$. The denoiser then outputs the noise residual $\mathcal{G}(y_i) = \mathcal{R}(y_i, \mathcal{E}(y_i))$. Three objectives are proposed to supervise the network training, including the noise estimation ($\mathcal{L}_e$), blind ($\mathcal{L}_b$), and non-blind ($\mathcal{L}_{nb}$) image denoising objectives, defined as,

$$\mathcal{L}_e = \frac{1}{2N} \sum_{i=1}^{N} ||\mathcal{E}(y_i; \Theta_E) - e_i||_F^2, \tag{3.1}$$

$$\mathcal{L}_b = \frac{1}{2N}\sum_{i=1}^{N}||\mathcal{R}(y_i, \mathcal{E}(y_i;\Theta_E);\Theta_R) - v_i||_F^2, \tag{3.2}$$

$$\mathcal{L}_{nb} = \frac{1}{2N}\sum_{i=1}^{N}||\mathcal{R}(y_i, e_i;\Theta_R) - v_i||_F^2, \tag{3.3}$$

where Θ_E and Θ_R are the trainable parameters of $\mathcal{E}$ and $\mathcal{R}$. e_i is the ground truth noise level maps for y_i, consisting of e_{iAWGN} and e_{iRVIN}. For AWGN, e_{iAWGN} is represented as the even maps filled with the same standard deviation values ranging from 0 to 75 across R,G,B channels. For RVIN, e_{iRVIN} is represented as the maps valued with the corrupted pixels ratio with upper-bound set to 0.3. I further normalize e_i to range [0,1]. Then the full objective can be represented as a weighted sum of the above three losses,

$$\mathcal{L} = \alpha\mathcal{L}_e + \beta\mathcal{L}_b + \gamma\mathcal{L}_{nb}, \tag{3.4}$$

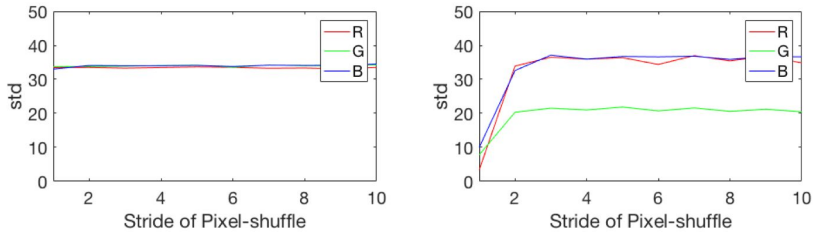
in which α, β and γ are hyper-parameters to balance the losses, and I set them to be equal for simplicity.

The proposed model structure can perform both blind and non-blind denoising simultaneously, and the model is more flexible in interactive denoising and result adjustment. Explicit noise estimation also benefits noise modeling and disentanglement.

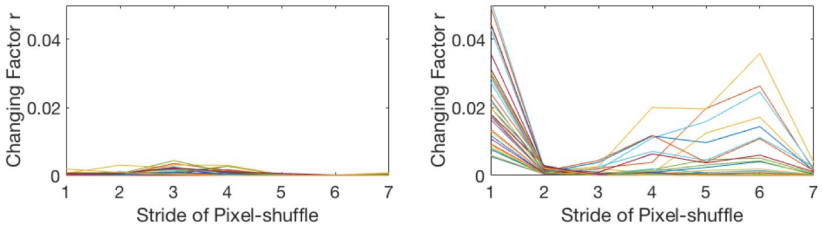
3.4 Pixel-shuffle Down-sampling (PD) Adaptation

Pixel-shuffle Down-sampling. Pixel-shuffle [70] down-sampling is defined to create the mosaic by sampling the images with stride s. Compared to other down-sampling methods like linear interpolation, bi-cubic interpolation, and pixel area relation, the pixel-shuffle, and nearest-neighbour down-sampling on noisy image would not influence the real noise distribution. Besides, pixel-shuffle also benefits image recovery by preserving the original pixels from the images compared to others. These two advantages yield the two stages of PD strategy: adaptation and refinement.

Adaptation. A Learning-based denoiser trained on AWGN is not robust enough to real noises because of domain-dependent differences in the local noise co-variance. To adapt the noise model to real noise, here we briefly



(a) As the stride increases, left: Estimated noise level on AWGN-corrupted image. right: Estimated noise level on real noisy images.



(b) left: Changing factor r_s on AWGN-corrupted images of CBSD68 and right: on real noisy images of DND. Different color lines represent different image samples.

Figure 3.4: Influence of Pixel-shuffle on noise patterns and noise estimation algorithms.

analyze and justify our assumption on the difference between real noises and Gaussian noise: spatial/channel variance and correlation.

Suppose a noise estimator is robust, which means it can accurately estimate the exact noise level, for a single AWGN-corrupted image, pixel-shuffle down-sampling will neither influence the AWGN variance nor the estimation values, when the sample stride is small enough to preserve the textural structures. When extending it to the real noise case, we have an interesting hypothesis: as we increase the sample stride of pixel-shuffle, the estimation values of specific noise estimators will first fluctuate and then keep steady for a couple of stride increment. This assumption is feasible because pixel-shuffle will break down the spatial-correlated noise patterns to pixel-independent ones, which can be approximated as spatial-variant AWGN and adapted to those estimators.

We justify this hypothesis on both [71] and our proposed pixel-wise estimator. As shown in Figure 3.1, I randomly cropped a patch of size 200×200 from a random noisy image y in SIDD[59]. We add AWGN with $std = 35$ to its noise-free ground truth x. After pixel-shuffling both y and AWGN-corrupted x, starting from stride $s = 2$, the noise pattern of y demonstrates expected pixel independence. Using [71], the estimation result for x is unchanged in Figure 3.4 (a) (left), but the one for y in Figure 3.4 (a) (right) first increases

and begins to keep steady after stride $s = 2$. It is consistent with the visual pattern and our hypothesis.

One assumption of [71] is that the noise is additive and evenly distributed across the image. For spatial-variant signal-dependent real noises, our pixel-wise estimator has its superiority. To make statistics of spatial-variant noise estimation values, we extract the three AWGN channels of noise map $\mathcal{E}_{AWGN}(y_i) \in R^{W \times H \times 3}$, where W and H are width and height of the input image, and compute the normalized 10-bin histograms $h_s \in R^{10 \times 3}$ across each channel when the stride is s. We introduce the changing factor r_s to monitor the noise map distribution changes as the stride s increases,

$$r_s = E_c ||h_{sc} - h_{(s+1)c}||_2^2, \tag{3.5}$$

where c is the channel index. I then investigate the difference of r_s sequence between AWGN and realistic noises. Specifically, we randomly select 50 images from CBSD68 [72] and add random-level AWGN to them. For comparison, we randomly pick up 50 image patches of 512×512 from DND benchmark. In Figure 3.4 (b), r_s sequence remains closed to zero for all AWGN-currupted images (Left figure), while for real noises r_α demonstrates an abrupt drop when $s = 2$. It indicates that the spatial-correlation has been broken from $s = 2$.

The above analysis inspires the proposed adaptation strategy based on pixel-shuffle. Intuitively, we aim at finding the smallest stride s to make the down-sampled spatial-correlated noises match the pixel-independent AWGN. Thus we keep increasing the stride s until r_s drops under a threshold τ. We run the above experiments on CBSD68 for 100 iterations to select the proper generalized threshold τ. After averaging the maximum r of each iteration, we empirically set $\tau = 0.008$.

PD Refinement. Figure 3.5 shows the proposed Pixel-shuffle Down-sampling (PD) refinement strategy: (1) Compute the smallest stride s, which is 2 in this example and more digital camera image cases, to match AWGN following the adaptation process, and pixel-shuffle the image into mosaic y_s; (2) Denoise y_s using $\mathcal{G}$; (3) Refill each sub-image with noisy blocks separately and pixel-shuffle upsample them; (4) Denoise each refilled image again using $\mathcal{G}$ and average them to obtain the "texture details" T; (5) Combine the

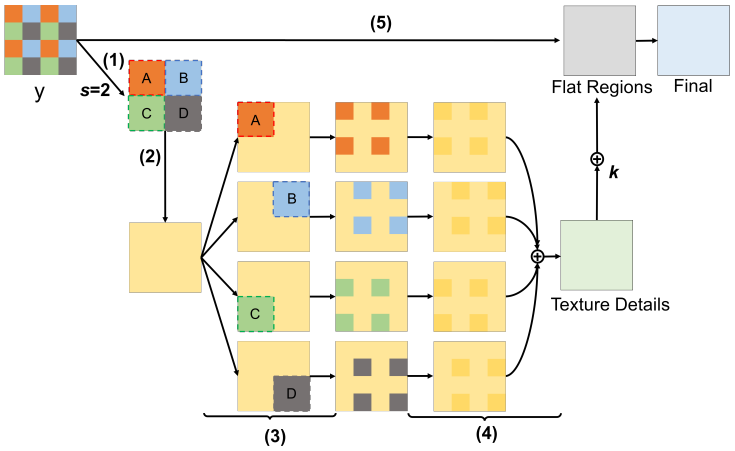


Figure 3.5: Pixel-shuffle Down-sampling (PD) refinement strategy with $s = 2$.

over-smoothed "flat regions" F to refine the final result.

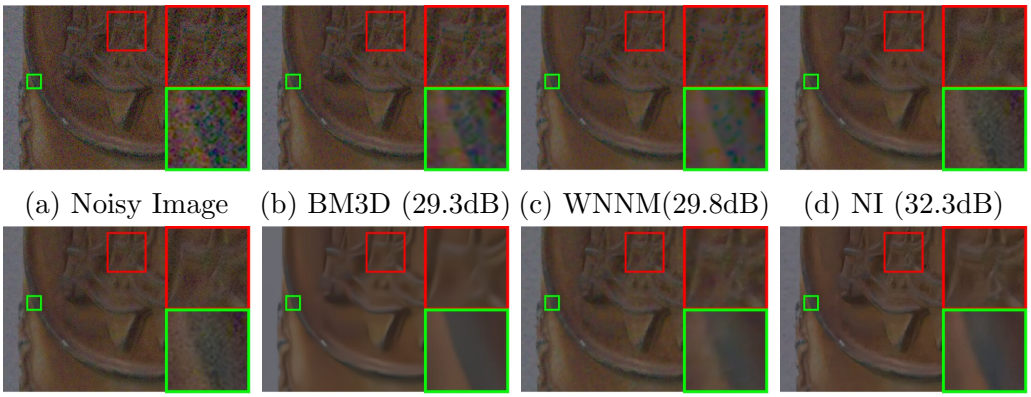
As summarized in [44], the goals of noise removal include preserving texture details and boundaries, smoothing flat regions, and avoiding generating artifacts. Therefore, in the above step-(5), we propose to further refine the denoised image with the combination of "texture details" T and "flat regions" F. "Flat regions" can be obtained from over-smoothed denoising results generated by lifting the noise estimation levels. In this work, given a noisy observation y, the refined noise maps are defined as,

$$\mathcal{E}(P\hat{D}(y))(i,j) = \max_{i,j} \mathcal{E}(PD(y))(i,j), i \in [1,W], j \in [1,H], \tag{3.6}$$

Consequently, the "flat region" is defined as $F = PU(\mathcal{R}(PD(y), \mathcal{E}(P\hat{D}(y))))$, where PD and PU are pixel-shuffle downsampling and upsampling. The final result is obtained by $kF + (1-k)T$.

3.5 Experiments

Implementation Details In this work, the structures of the sub-network $\mathcal{E}$ and $\mathcal{R}$ follow DnCNN [1] of 5 layers and 20 layers. For grayscale image experiments, we also follow DnCNN to crop 50×50 patches from 400 images of size 180×180. For color image model, we crop 50×50 patches with stride 10 from 432 color images in the Berkeley segmentation dataset (BSD) [72].



(a) Noisy Image (b) BM3D (29.3dB) (c) WNNM(29.8dB) (d) NI (32.3dB)

(e) NC (32.3dB) (f) FFD (34.5dB) (g) CBD (34.5dB) (h) Ours (36.1dB)

Figure 3.6: Denoising results on DND Benchmark. Red box indicates texture details while the green box background or edge.

The training data ratio of single-type noises (either AWGN or RVIN) and mixed noises (AWGN and RVIN) is 1:1. During training, Adam optimizer is utilized and the learning rate is set to 10^{-3}, and batch size is 128. After 30 epochs, the learning rate drops to 10^{-4} and the training stops at epoch 50.

To evaluate the algorithm on synthetic noise (AWGN, mixed AWGN-RVIN and spatially-variant Gaussian), we utilize the benchmark data from BSD68, Set20 [69] and CBSD68 [72]. For realistic noise, we test it on RNI15 [73], DND benchmark [58], and self-captured night photos. We evaluate the performance of the algorithm in terms of PSNR and SSIM. Qualitative performance for denoising is also presented, with comparison to other state-of-the-art algorithms.

3.5.1 Evaluation with Synthetic Noise

Table 3.1: Comparison of PSNR results on mixture of Gaussian noise (AWGN) and Impulse noise (RVIN) removal performance on Set20.

(σ, r)	BM3D	WNNM	PGB	DnCNN-B	Ours-NB	Ours-B
(10, 0.15)	25.18	25.41	27.17	32.09	**32.43**	32.37
(10, 0.30)	21.80	21.40	22.17	29.97	**30.47**	30.32
(20, 0.15)	25.13	23.57	26.12	29.52	**29.82**	29.76
(20, 0.30)	21.73	21.40	21.89	27.90	**28.41**	28.16

Mixed AWGN and RVIN. The model follows similar structure of DnCNN and FFDNet [5], so its performance on single-type AWGN removal is also

Table 3.2: Comparison of PSNR results on Signal-dependent Noises on CBSD68.

(σ_s, σ_c)	BM3D	FFDNet	DnCNN-B	CBDNet	Ours-B
(20, 10)	29.09	28.54	34.38	33.04	**34.75**
(20, 20)	29.08	28.70	**31.72**	29.77	31.32
(40, 10)	23.21	28.67	32.08	30.89	**32.12**
(40, 20)	23.21	28.80	30.32	28.76	**30.33**

similar to them. We thus evaluate our model on eliminating mixed AWGN and RVIN on Set20 as in [69]. We also compare our method with other baselines, including BM3D [74] and WNNM [75] which are non-blind Gaussian denoisers anchored with a specific noise level estimated by the approach provided in [71]. Besides, we include the PGB [69] denoiser that is designed for mixed AWGN and RVIN. The result of the blind version of DnCNN-B, trained by the same strategy as our model, is also presented for reference. The comparison results are shown in Table 3.1, from which we can see the proposed method achieves the best performance. Compared to DnCNN-B, for complicated mixed noises, our model explicitly disentangles different noises. It benefits the conditional denoiser to differentiate mixed noises from other types.

Signal-dependent Spatially-variant Noise. We conduct experiments to examine the generalization ability of the fully convolutional model on signal-dependent noise [39, 76, 77]. Given a clean image x, the noises in the noisy observation y contain both signal-dependent components with variance $x\sigma_s^2$ and independent components with variance σ_c^2. Table 3.2 shows that for non-blind models like BM3D and FFDNet, only a scalar noise estimator [71] is applied, thus they cannot well cope with the spatially-variant cases. In this experiment, DnCNN-B is the original blind model trained on AWGN with σ ranged between 0 and 55. It shows that spatially-variant Gaussian noises can still be handled by fully convolutional models trained with spatially-invariant AWGN [5]. Compared to DnCNN-B, the proposed network explicitly estimates the pixel-wise map to make the model more flexible and possible for real noise adaptation.

3.5.2 Evaluation with Real RGB Noise

In this section, we introduce the evaluation results on real RGB noiay images. We present the qualitative and quantitative comparisons with state-of-the-arts.

Qualitative Comparisons. Some qualitative denoising results on DND are shown in Figure 3.6. The compared results of DND are all directly obtained online from the original submissions of the authors. The methods we include for the comparison cover blind real denoisers (CBDNet, NI [78] and NC [79]), blind Gaussian denoisers (CDnCNN-B) and non-blind Gaussian denoisers (CBM3D, WNNM [75], and FFDNet). From these example denoised results, we can observe that some of them are either noisy (as in DnCNN and WNNM), or spatially-invariantly over-smoothed (as in FFDNet). CBDNet performs better than others but it still suffers from blur edges and uncleaned background. Our proposed method (PD) achieves a better spatially-variant denoising performance by smoothing the background while preserving the textural details in a full blind setting.

Quantitative Results on DND Benchmark. The images in the DND benchmark are captured by digital camera and demosaiced from raw sensor data, so we simply set the stride number $s = 2$. We follow the submission guideline of the DND dataset to evaluate our algorithm. Recently, many learning-based methods like Path-Restore [80], RIDNet [81], WDnCNN [82], and CBDNet, achieved promising performance on DND, but they are all finetuned on real noisy images, or use prior knowledge in the meta-data of DND [40]. For fair comparison, we select some representative conventional methods(MCWNNM, EPLL, TWSC, CBM3D), and learning-based methods trained only with synthetic noises. The results are shown in Table 3.3. Models trained on AWGN (DnCNN, TNRD, MLP) perform poorly on real RGB noises mainly due to the large gap between AWGN and real noise. CBDNet improves the results significantly by training the deep networks with artificial realistic noise model. Our AWGN-RVIN-trained model with PD refinement achieves much better results (+0.83dB) than CBDNet trained only with synthetic noises, and also boosts the performance of other AWGN-based methods (+PD). Compared to the base model, the proposed adaptation methods improve the performance on real noises by 5.8 dB. Note that our

Table 3.3: Comparison of PSNR and SSIM on DND Benchmark. PD: Pixel-suffle Down-sampling Strategy. Among all models trained only with synthetic data.

Method	Category	Type	PSNR	SSIM
WNNM[75]	Low Rank	Non-blind	34.67	0.8646
MCWNNM[35]	Low Rank	Non-blind	37.38	0.929
KSVD[37]	Sparse Coding	Non-blind	36.49	0.8978
TWSC[68]	Sparse Coding	Non-blind	37.93	0.940
NCSR[83]	Sparse Coding	Non-blind	34.05	0.8351
MLP[84]	Deep Learning	Non-blind	34.23	0.833
TNRD[19]	Deep Learning	Non-blind	33.65	0.830
CBDNet(Syn)[39]	Deep Learning	Blind	37.57	0.936
CBM3D[34]	Filter	Non-blind	34.51	0.850
CBM3D(+PD)	Filter	Non-blind	*35.02*	*0.873*
CDnCNN-B[1]	Deep Learning	Blind	32.43	0.790
CDnCNN-B(+PD)	Deep Learning	Blind	*35.44*	*0.876*
FFDNet[5]	Deep Learning	Non-blind	34.40	0.847
FFDNet(+PD)	Deep Learning	Non-blind	*37.56*	*0.931*
Our Base Model	Deep Learning	Blind	32.60	0.788
Ours(+PD)	Deep Learning	Blind	**38.40**	**0.945**

model is only trained on synthetic noises, and does not utilize any prior data of DND.

3.5.3 Ablation Study on Real RGB Noise

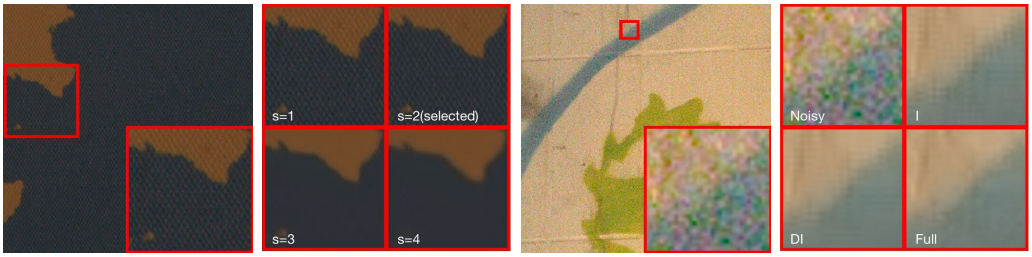
In this section, we show the extensive ablation study to evaluate the components of the proposed pipeline.

Adding RVIN. Training models with mixed AWGN and RVIN noises will benefit the removal of dead or over-exposure pixels in real images. For comparison, we train another model only with AWGN, and test it on real noisy night photos. An example utilizing the full pipeline is shown in Figure 3.7, in which it demonstrates the superiority of the existence of RVIN in the training data. Even though model trained with AWGN can also achieve promising denoising performance, it is not effective on dead pixels.

Stride Selection. I apply different stride numbers while refining the denoised results, and compare the visual quality in Figure 3.8 (a)(b). For

(a) Noisy image (b) AWGN only (c) AWGN-RVIN

Figure 3.7: Denoised performance of models trained with AWGN in (b) and mixed AWGN-RVIN in (c). During testing, $k = 0$ and $s = 2$.

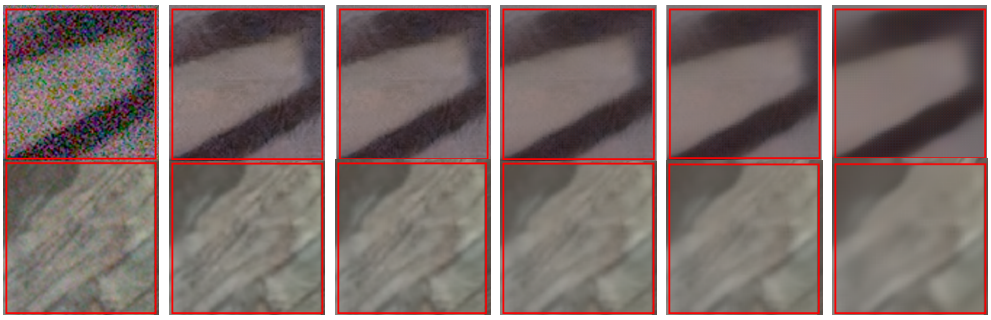


(a) Noisy image (b) Denoised. (c) Noisy Image (d) Denoised.

Figure 3.8: (a)(b): Denoised performance of different stride s when $k = 0$, and (c)(d): Ablation study on refinement. $s = 2$ and $k = 0$.

arbitrary given RGB images, the stride number can be computed using our adaptation algorithm with the assistance of noise estimator. In our experiments, the selected stride is the smallest s that $r_s < \tau$. Small stride number will treat large noise patterns as textures to preserve, as shown in Figure 3.8 (b). While using large stride number tends to break the textural structures and details. Interestingly, as shown in Figure 3.8 (b), the texture of the fabric is invisible while applying $s > 2$.

Image Refinement Process. The ablation on the refinement steps is shown in Figure 3.8 (c)(d) and Table 3.4, in which we compare the denoised results of I (i.e. directly pixel-shuffling upsampling after step (2)), DI (i.e, denoising I using $\mathcal{G}$), and Full (i.e, the current whole pipeline). It shows that both I and DI will form additional visible artifacts, while the whole pipeline smooths out those artifacts and has the best visual quality.



(a) Noisy (b) 0 (c) 0.3 (d) 0.5 (e) 0.8 (f) 1

Figure 3.9: Ablation study on merging factor k, and $s = 2$.

Table 3.4: Ablation study on refinement steps.

Model	(s=1)	(s=3, Full)	(s=2,I)	(s=2,DI)	(s=2,Full)
PSNR	32.60	37.90	37.00	37.20	**38.40**
SSIM	0.7882	0.9349	0.9339	0.9361	**0.9452**

Blending Factor k. Due to the ambiguous nature of fine texture and mid-frequency noises, interactions between human perception and denoising effectiveness are inevitable. k is this parameter introduced as a "linear" adjustment of denoising level for a more flexible and interactive user operation. Using blending factor k is more stable and safe to preserve the spatially-variant details than directly adjusting the estimated noise level like CBDNet. In Figure 3.9, as k increases, the denoised results tend to be over-smoothed. This is suitable for images with more background patterns. However, smaller k will preserve more fine details which are applicable for images with more foreground objects. In most cases, users can simply set k to 0 to obtain the most detailed textures recovery and visually plausible results.

3.6 Conclusions

In this chapter, we revisit the real image blind denoising from a new viewpoint. Due to the demosaicing process, we assumed the realistic noises are spatially/channel -variant and correlated, and addressed adaptation from AWGN-RVIN noises to real RGB noises. Specifically, we proposed an image blind and non-blind denoising network trained on AWGN-RVIN noise model. The network consists of an explicit multi-type multi-channel noise estimator

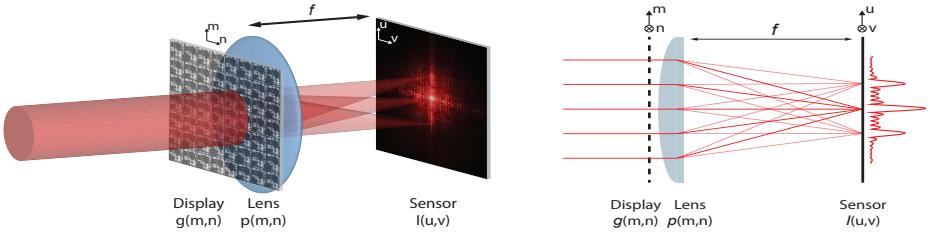
and an adaptive conditional denoiser. To generalize the network to real RGB noises, we investigated Pixel-shuffle Down-sampling (PD) refinement strategy. The PD adaptor was applied to the testing RGB images, and we showed qualitatively that PD behaves better in both spatially-variant denoising and details preservation. Results on DND benchmark and other realistic noisy images demonstrated the newly proposed model with the strategy are efficient in processing spatial/channel variance and correlation of real noises without explicit modeling.

CHAPTER 4

PHYSICS-BASED DATA SYNTHESIS ON REAL COMBINED RESTORATION

In this chapter, we study a physics-based training data synthesis for a more complicated degradation problem called Under-Display Camera imaging. We study the performance of deep networks trained on the data synthesized by the newly-proposed degradation model.

4.1 Introduction



(a) 3D schematics (b) 2D schematics

Figure 4.1: Under-Display Camera (UDC), a new imaging system that mounts display screen on top of a traditional digital camera lens.

Under-display Camera (UDC) is a new imaging system that mounts display screen on top of a traditional digital camera lens, as shown in Fig. 4.1. Such a system has mainly two advantages. First, it follows a new product trend of full-screen devices [85] with larger screen-to-body ratio, which can provide better user perception and intelligent experience [86]. Without seeing the bezel and extra buttons, users can easily access more functions by directly touching the screen. Second, it provides better human computer interaction. By putting the camera in the center of the display, it enhances teleconferencing experiences with perfect gaze tracking, and it is increasingly relevant for larger display devices such as laptops and TVs.

Unlike pressure or fingerprint sensors which can be more easily integrated

into a display, it is relatively hard to maintain the function of an imaging sensor after being mounted behind a display. The imaging quality of a camera will be severely degraded due to lower light transmission rate and diffraction effects. As a result, images captured will be noisy and blurry. Therefore, while bringing better user experience and interaction, UDC may sacrifice the quality of photography, face processing and other downstream vision tasks.

As discussed in the previous chapters, enhancing the degraded images can be better addressed by learning-based image restoration approaches. However, since they are only trained on synthesis data with a single degradation type, existing state-of-the-art models can be hardly utilized to enhance real-world low-quality images with complicated and combined degradation types. To address complicated real degradation using learning-based methods, collecting real paired data or synthesizing near-realistic data by fully understanding the degradation model is necessary.

In this chapter, we define and present a novel Under-Display Camera image restoration problem. UDC restoration task is a combination of tasks such as low-light enhancement, de-blurring, and de-noising. Without loss of generality, the analysis focuses on **two types of displays**, a 4K Transparent Organic Light-Emitting Diode (T-OLED) and a phone Pentile OLED (P-OLED), and **a single camera type,** a 2K FLIR RGB Point Grey research camera. We acquire the training data by either collecting real data with a newly proposed data acquisition system, or synthesizing near-realistic data with a model-based pipeline.

4.2 Formulation

In this section, we introduce the formulation of the UDC imaging process. We first analyze the optical system, and derive the physics-based forward model for data synthesis. Given the display pattern and related measurements, our method synthesizes degraded data from degradation-free images.

4.2.1 Optical System Analysis

In this work, we focus on OLED displays as it has superior optical properties compared to traditional LCDs (Liquid Crystal Display). Due to confidentiality

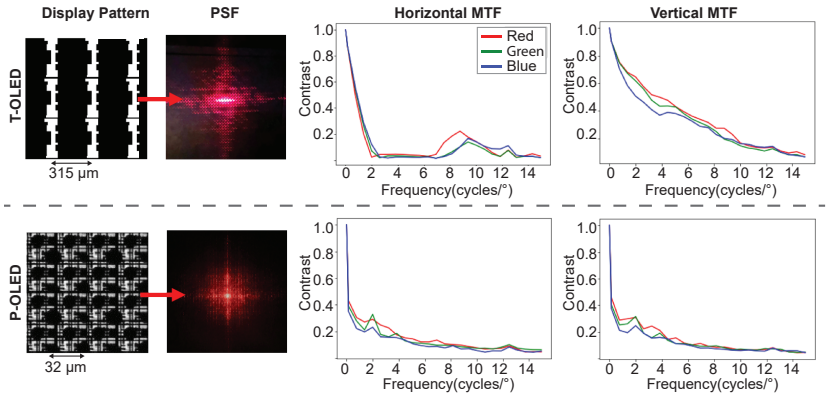


Figure 4.2: Optics characteristics of UDC. From left to right: Micrography of display patterns, PSFs (red light only) and MTFs (all red, green, and blue lights).

Table 4.1: Comparison of two displays

Metrics	T-OLED	P-OLED
Pixel Layout Type	Stripe	Pentile
Open Area	21%	23%
Transmission Rate	20%	2.9%
Major Degradation	Blur, Noise	Low-light, Color Shift, Noise

reasons it is often difficult to obtain the sample materials used for demos from commercial companies. In this case, we select the displays with different transparencies to improve the generalization. Note that all the displays are **nonactive** in our paper, since in real scenario, the display can be turned off locally when the camera is in operation to (1) reduce unnecessary difficulty from display contents while not affecting user experience and (2) provide users with the status of the device and thus ensure privacy.

Owing to transparent materials being used in OLED display panels, visible light can be better transmitted through the OLEDs than LCDs. In the meantime, pixels are also arranged such that open area is maximized. In particular, we focus on 4k Transparent organic light-emitting diode (T-OLED) and a phone Pentile OLED (P-OLED). Fig. 4.2 is a micrograph illustration of the pixel layout in the two types of OLED displays. The structure of the 4K T-OLED has a grating-like pixel layout. P-OLED differs from T-OLED in sub-pixel design. It follows the basic structure of RGBG matrix. In this section, we analyze the two types of displays according to their light transmission rate, point spread function (PSF), and modulation transfer function (MTF).

Light Transmission Rate We measure the transmission efficiency of the OLEDs by using a spectrophotometer and white light source. Table 4.1 compares the light transmission rate of the two displays. For T-OLED, the open area occupies about 21%, and the light transmission rate is around 20%. For P-OLED, although the open area can be as large as 23%, the light transmission rate is only 2.9%. P-OLED is a flexible/bendable display, which has a poly-amide substrate on which the OLED is formed. Such a substrate may appear yellow in transmission. Thus, images captured through a polyamide-containing display panel by a UDC may also appear yellow. As a result, imaging through a P-OLED results in lower signal-to-noise ratio (SNR) comparing to using a T-OLED, and has a color shift issue.

Diffraction Pattern and Point Spread Function (PSF) Light diffracts as it propagates through obstacles with sizes that are similar to its wavelength. Unfortunately, the size of the openings in the pixel layout is on the order of wavelength of visible light, and images formed will be degraded due to diffraction. Here we characterize the system by measuring the point spread function (PSF). We do so by pointing a collimated red laser beam ($\lambda = 650$ nm) at the display panel and recording the image formed on the sensor, as demonstrated in Fig. 4.1. Fig. 4.2 shows the PSFs. An ideal PSF shall resemble a delta function, which then forms a perfect image of the scene. However, light greatly spreads out in UDC. For T-OLED, light spreads mostly across the horizontal direction due to its nearly one dimensional structure in the pixel layout, while for P-OLED, light is more equally distributed as the pixel layout is complex. Therefore, images captured by UDC are either blurry (T-OLED) or hazy (P-OLED).

Modulation Transfer Function (MTF) Modulation Transfer Function (MTF) is another important metric for an imaging system, as it considers the effect of finite lens aperture, lens performance, finite pixel size, noise, non-linearities, quantization (spatial and bit depth), and diffraction in our systems. We characterize the MTF of our systems by recording sinusoidal patterns with increasing frequency in both lateral dimensions, and we report them in Fig. 4.2. For T-OLED, contrasts along the horizontal direction are mostly lost in the mid-band frequency due to diffraction. This phenomenon is due to the one-dimensional pixel layout of the T-OLED. Fig. 4.4 shows severe

smearing horizontally when putting T-OLED in front of the camera. While for P-OLED, the MTF is almost identical to that of display-free camera, except with severe contrast loss. Fortunately, however, nulls have not been observed in any particular frequencies.

4.2.2 Image Formation Model

In this section, we derive the image formation process of UDC. In other words, given a calibrated pixel layout and measurements using a specific camera, degraded images can be simulated from a scene. From the forward model, we can synthesize datasets from ground truth images.

Given an object in the scene $\mathbf{x}$, the degraded observation $\mathbf{y}$ can be modeled by a convolution process,

$$\mathbf{y} = (\gamma \mathbf{x}) \otimes \mathbf{k} + \mathbf{n}, \tag{4.1}$$

where γ is the intensity scaling factor under the current gain setting and display type, $\mathbf{k}$ is the PSF, and $\mathbf{n}$ is the zero-mean signal-dependent noise. Notice that this is a simple noise model that approximately resembles the combination of shot noise and readout noise of the camera sensor, and it will be discussed in a later section.

Intensity Scaling Factor (γ) The intensity scaling factor measures the changing ratio of the average pixel values after covering the camera with a display. It simultaneously relates to the physical light transmission rate of the display, as well as the digital gain setting of the camera. γ can be computed from the ratio of δ-gain amplified average intensity values $I_d(\delta, s)$ at position s captured by UDC, to the 0-gain average intensity values $I_{nd}(0, s)$ by naked camera within an enclosed region S. It is represented by,

$$\gamma = \frac{\int_S I_d(\delta, s) ds}{\int_S I_{nd}(0, s) ds} \tag{4.2}$$

Diffraction Model We approximate the blur kernel $\mathbf{k}$, which is the Point Spread Function (PSF) of the UDC. As shown in Fig 4.1, in our model, we assume the display panel is at the principle plane of the lens. We also

assume the input light is a monochromatic plane wave with wavelength λ (i.e. perfectly coherent), or equivalently light from a distance object with unit amplitude. Let the display pattern represented by transparency with complex amplitude transmittance be $g(m,n)$ at the Cartesian co-ordinate (m,n), and let the camera aperture/pupil function $p(m,n)$ be 1 if (m,n) lies inside the lens aperture region and 0 otherwise, then the display pattern inside the aperture range $g_p(m,n)$ becomes,

$$g_p(m,n) = g(m,n)p(m,n). \tag{4.3}$$

At the focal plane of the lens (i.e, 1 focal length away from the principle plane), the image measured is the intensity distribution of the complex field, which is proportional to the Fourier transform of the electric field at the principle plane [87]:

$$I(u,v) \propto \left| \iint_{-\infty}^{\infty} g_p(m,n) \exp\left[-j\frac{2\pi}{\lambda f}(mu+nv)\right] \mathrm{d}m\mathrm{d}n \right|^2. \tag{4.4}$$

Suppose $G_p(v_m, v_n) = F(g_p(m,n))$, where $F(\cdot)$ is the Fourier transform operator, then

$$I(u,v) \propto |G_p(v_m,v_n)|^2 = \left| G_p(\frac{u}{\lambda f}, \frac{v}{\lambda f}) \right|^2, \tag{4.5}$$

which performs proper scaling on the Fourier transform of the display pattern on the focal plane.

Therefore, to compute the PSF $\mathbf{k}$ for image $\mathbf{x}$, we start from computing the Discrete Fourier Transform (DFT) with squared magnitude $M(a,b) = |\hat{G}_p(a,b)|^2$ of the $N \times N$ microscope transmission images $\hat{g_p}$ of the display pattern and re-scaling it. Then, the spatial down-sampling factor r becomes,

$$r = \frac{1}{\lambda f} \cdot \delta_N N \cdot \rho, \tag{4.6}$$

where δ_N is the pixel size of the $\hat{g_p}$ images, and ρ is the pixel size of the sensor. Finally, $\mathbf{k}$ can be represented as

$$k(i,j) = \frac{M_{\downarrow r}(i,j)}{\sum_{(\hat{i},\hat{j})} M_{\downarrow r}(\hat{i},\hat{j})}. \tag{4.7}$$

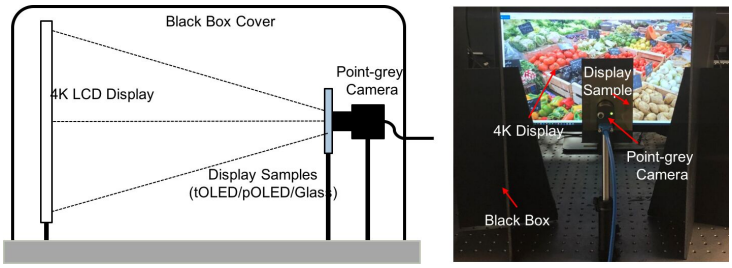


Figure 4.3: Monitor-Camera Imaging System (MCIS).

k is a normalized form since we want to guarantee that it represents the density distribution of the intensity with diffraction effect. Note that only PSF for a single wavelength is computed for simplicity. However, scenes in the real-world are by no means monochromatic. Therefore, in order to calculate an accurate color image from such UDC systems, PSF for multiple wavelengths shall be computed.

Adding Noises We follow the commonly used shot-read noise model [40, 88, 89] to represent the real noises on the imaging sensor. Given the dark and blur signal $w = (\gamma \mathbf{x}) \otimes \mathbf{k}$, the shot and readout noises can be modeled by a heteroscedastic Gaussian,

$$\mathbf{n} \sim \mathcal{N}(\mu = 0, \sigma^2 = \lambda_{read} + \lambda_{shot} w), \tag{4.8}$$

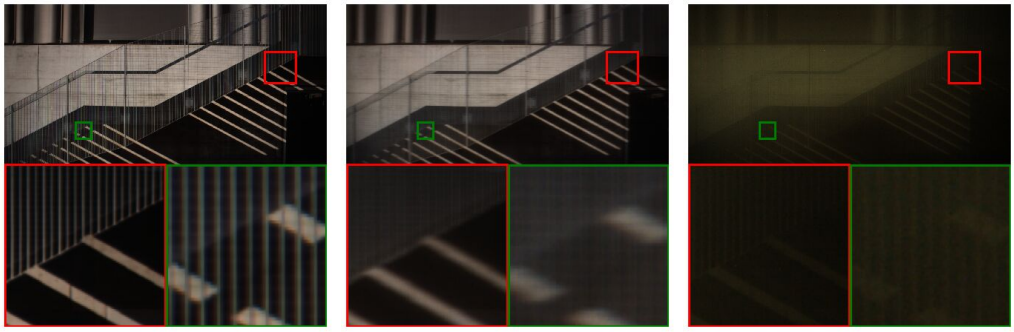
where the variance σ is signal-dependent, and λ_{read} , λ_{shot} are determined by camera sensor and gain values.

4.3 Data Acquisition and Synthesis

In this section, we describe the proposed imaging system for data collection and the practical process of synthesizing paired data.

4.3.1 Monitor-Camera Imaging System (MCIS)

To collect real paired data for training, we propose a novel image acquisition system called Monitor-Camera Imaging System (MCIS). In particular, we display natural images with rich textures on high-resolution monitor and



(a) Display-free (b) TOLED (c) POLED

Figure 4.4: Real samples collected by the proposed MCIS.

capture them with a static camera. The method is more controllable, efficient, and automatic to capture a variety of scene contents than using mobile set-ups to capture limited static objects or real scenes.

The system architecture is shown in Fig. 4.3. MCIS consists of a 4K LCD monitor, the 2K FLIR RGB Point-Grey research camera, and a panel that is either T-OLED, P-OLED or Glass (i.e, no display). The camera is mounted on the center line of the 4K monitor, and adjusted to cover the full monitor range. We calibrate the camera gain by measuring a 256×256 white square shown on the monitor and matching the RGB histogram. For fair comparison and simplicity, we adjust the focus and fix the aperture to f/1.8. It guarantees a reasonable pixel intensity range avoiding saturation while collecting data with zero gain. Suppose we develop a real-time video system, the frame rate has to be higher than 8 fps. So the lowest shutter speed is 125 ms for the better image quality and the higher Signal-to-Noise Ratio (SNR).

We split 300 images from DIV2K dataset [90], and take turns displaying them on a 4K LCD in full screen mode. We either rotate or resize the images to maintain the Aspect Ratio. For training purposes, we capture two sets of images, which are the degraded images $\{y_i\}$, and the degradation-free set $\{x_i\}$.

To capture $\{x_i\}$, we first cover the camera with a thin glass panel which has the same thickness as a display panel. This process allows us to avoid the pixel misalignment issues caused by light refraction inside the panel. To eliminate the image noises in $\{x_i\}$, we average the 16 repeated captured frames. Then we replace the glass with a display panel (T-OLED or P-OLED), calibrate the specific gain value avoiding saturation, and capture

Table 4.2: Camera Settings for different set of collected data

Parameteres	No-Display	T-OLED	P-OLED
Aperture	f/1.8		
FPS/Shutter	8/125ms		
Brightness	0		
Gamma	1		
Gain	0	6	25(Full)
White-balance	Yes	None	None

Table 4.3: Measured parameters for data synthesis

Parameteres	T-OLED			P-OLED		
	R	G	B	R	G	B
γ	0.97	0.97	0.97	0.34	0.34	0.20
λ (nm)	640	520	450	640	520	450
r	2.41	2.98	3.44	2.41	2.98	3.44

$\{y_i\}$. For each set, we record both the 16-bit 1-channel linear RAW CMOS sensor data as well as the 8-bit 3-channel linear RGB data after in-camera pipeline that includes demosaicing. The collected pairs are naturally well spatially-aligned in pixel-level. They can be directly used for training without further transformations.

Due to the yellow substrate inside the P-OLED, certain light colors, especially blue, are filtered out , and this filtering changes the white balance significantly. We therefore did not further alter the white balance. The light transmission ratio of P-OLED is extremely low, so we set up the gain value to be the maximum (25) for higher signal values. All the detailed camera settings for the two display types are shown in Table 4.2. One real data sample is shown in Fig. 4.4

4.3.2 Realistic Data Synthesis Pipeline

We follow the image formation pipeline to simulate the degradation on the collected $\{x_i\}$. A model-based data synthesis method will benefit concept understanding and further generalization. Note that all the camera settings are the same as the one while collecting real data. We first transform the 16-bit raw sensor data $\{x_i\}$ into four bayer channels x_r, x_{gr}, x_{gl}, and x_b. Then, we multiply the measured intensity scaling factor γ, compute the normalized

and scaled PSF k, and add noises to the synthesize degraded data.

Measuring γ : To measure γ for each channel using the MCIS, we select the region of interest S to be a square region of size 256×256, and display the intensity value input from 0 to 255 with stride 10 on the monitor. We then record the average intensity both with and without the display for each discrete intensity value, and plot the relationship between display-covered values and no-display-covered ones. Using linear regression, we obtain the ratios of lines for different RGGB channel. For T-OLED, the measured γ is 0.97, same for all the channels. For P-OLED, $\gamma = 0.20$ for the blue channel, and $\gamma = 0.34$ for the other three channels.

Computing PSF : Following equation 4.3, we acquire the transmission microscope images of the display pattern and crop them with the approximated circular aperture shape with diameter $3333\mu m$, the size of the camera aperture. In equation 4.6, the $\delta_N N$ is $3333\mu m$. ρ equals to $1.55\mu m/pixel$ in Sony sensor. However, after re-arranging the raw image into four RGGB channels, ρ becomes 3.1 for each channel. The focal length for the lens is $f = 6000\mu m$. $\lambda = (640, 520, 450)$ for R, G, B channel, which are the approximated center peaks of the R, G, B filters respectively on the sensor. It yields the down-sampling ratio $r = (2.41, 2.98, 3.44)$ for the R, G and B channels.

Adding Noises : Finally, we measure λ_{read} and λ_{shot} to estimate the noise statistics. We display random patterns within the 256×256 window on the monitor, collect the paired noisy and noise-free RAW sensor data, and compute their differences. For each of the RGGB channel, we linearly regress the function of noise variance to the intensity value, and obtain the ratio as the shot noise variance, and the y-intersection as the readout noise variance. We then repeat the process for 100 times and collected pairs of data points. Finally, we estimate the distribution and randomly sample λ_{read} and λ_{shot} from it. All the measurements are listed in Table 4.3.

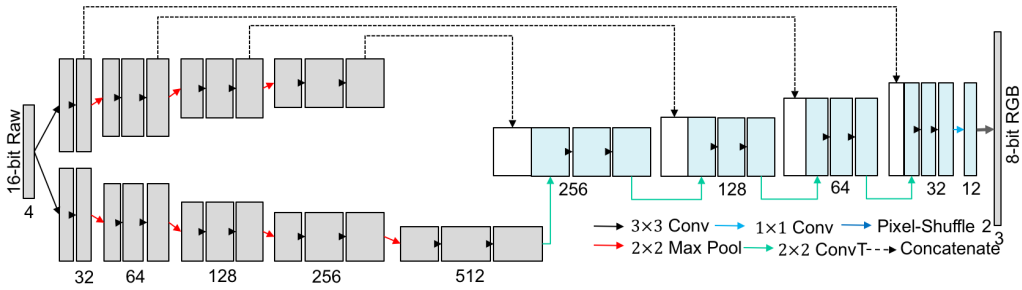


Figure 4.5: Network structure. It takes a 4-channel RAW sensor data observation y, and outputs the restored 3-channel RGB image x.

4.4 Image Restoration Baselines

We use the collected real paired data, synthetic paired data, simulated PSF, and all the necessary measurements to perform image restoration. We split the 300 pairs of images in the UDC dataset into 200 for training, 40 for validation and 60 images in the testing partition. All the images have a resolution of 1024×2048.

4.4.1 Deconvolution Pipeline (DeP)

The DeP is a general-purpose conventional pipeline concatenating denoising and deconvolution (Wiener Filter), which is an inverse process of the analyzed image formation. To better utilize the unsupervised Wiener Filter (WF) [91], we first apply the BM3D denoiser to each RAW channel separately, afterwards we linearly divide the measured γ with the outputs for intensity scaling. After that, WF is applied to each channel given the pre-computed PSF $\mathbf{k}$. Finally, RAW images with bayer pattern are demosaiced by linear interpolation. The restored results are evaluated on the testing partition of the UDC dataset.

4.4.2 Learning-based Methods

UNet. We propose a learning-based restoration network baseline as shown in Figure 4.5. The proposed model takes a 4-channel RAW sensor data observation y, and outputs the restored 3-channel RGB image x. The model conducts denoising, debluring, white-balancing, intensity scaling, and demosaicing in a single network, whose structure is basically a UNet. We split the

encoder into two sub-encoders, one of which is for computing residual details to add, and the other one learns content encoding from degraded images. By splitting the encoder, compared with doubling the width of each layer, we will have feIr parameters, and make the inference and learning more efficient. To train the model from paired images, we apply the L_1 loss, which will at large guarantee the temporal stability compared with adversarial loss [92]. Besides, we also apply $SSIM$ and Perception Loss (VGG Loss) for ablation study.

We crop patches of 256×256, and augment the training data using the raw image augmentation [4] while preserving the RGGB bayer pattern. We train the model for 400 epochs using Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$) with learning rate 10^{-4} and decay factor 0.5 after 200 epoches. We also train the same structure using the synthetic data (denoted as **UNet(Syn)**) generated by the pipeline proposed in section 4.2.2.

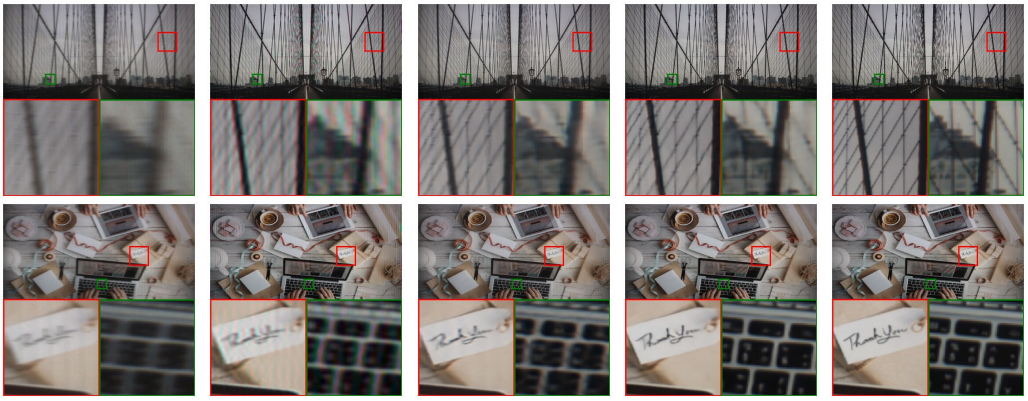
ResNet. Additionally, a data-driven ResNet trained with the same data is utilized for evaluation. UNet and ResNet-based structures are two widely-used deep models for image restoration. The ResNet used 16 residual blocks with 64 feature width from EDSR [13]. The model also takes 4-channel RAW data, and outputs 3-channel RGB images. The above mentioned baselines represent a conventional image processing pipeline and a ResNet-based deep model. Other model variants can be further explored in future work.

4.5 Experimental Results

In this section, the details of the our experiments and the results are shown. We qualitatively and quantitatively compare the proposed two pipelines with other state-of-the-arts. We also conduct ablations study to show the functions of different components.

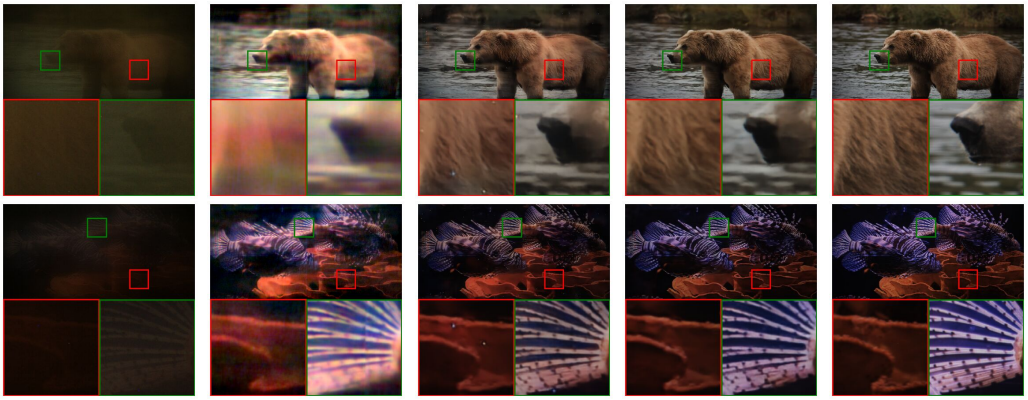
4.5.1 Qualitative and Quantitative Comparisons

The qualitative restoration results are shown in Figure 4.6 and 4.7. As shown, image Deconvolution Pipeline (DeP) successfully recovers image details but still introduces some artifacts, and suffers from the inaccuracy of the computed ideal PSF. The UNet-based model achieves better visual quality and denoising



(a) T-OLED (b) DeP[91] (c) UNet(Syn) (d) UNet(Real) (e) GT

Figure 4.6: Restoration Results Comparison for T-OLED. GT: Ground Truth.



(a) P-OLED (b) DeP[91] (c) UNet(Syn) (d) UNet(Real) (e) GT

Figure 4.7: Restoration Results Comparison for P-OLED. GT: Ground Truth.

performance. The results of UNet trained with the synthetic data are visually better than the ones of DeP.

The quantitative results are listed in Table 4.4. We report the performance in PSNR, SSIM, a perceptual metric LPIPS [93], inference time T (ms/MPixels) and GFLOPs. The inference time is tested with one single Titan X, and the GFLOPs is computed by input size of $512 \times 1024 \times 4$. ResNet achieves a comparable performance to UNet, but it requires more computation operations and longer inference time. The proposed UNet-based structure is efficient and effective, which can therefore be deployed for real-time inference for high-resolution inputs with a single GPU. In Table 4.4, we demonstrate that synthetic data still has gaps with the real data, though it has already

Table 4.4: Pipeline Comparison

.

				4K T-OLED		P-OLED	
Pipeline Structure	#P ↓	GFLOPs ↓	T ↓	PSNR/SSIM ↑	LPIPS ↓	PSNR/SSIM ↑	LPIPS ↓
DeP	-	-	-	28.50/0.9117	0.4219	16.97/0.7084	0.6306
ResNet	**1.37M**	721.76	92.92	36.26/0.9703	0.1214	27.42/0.9176	0.2500
UNet(Syn)	8.93M	**124.36**	**21.37**	32.42/0.9343	0.1739	25.88/0.9006	0.3089
UNet	8.93M	**124.36**	**21.37**	**36.71/0.9713**	**0.1209**	**30.45/0.9427**	**0.2219**

Table 4.5: Ablation Study on UNet alternatives.

Alternatives				4K T-OLED		P-OLED	
	#P ↓	GFLOPs ↓	T ↓	PSNR/SSIM ↑	LPIPS ↓	PSNR/SSIM ↑	LPIPS ↓
UNet Basseline	8.93M	124.36	21.37	36.71/0.9713	0.1209	30.45/0.9427	0.2219
Double Width	31.03M	386.37	40.42	37.00/0.9730	0.1171	30.37/0.9425	0.2044
Single Encoder	7.76M	97.09	15.85	36.47/0.9704	0.1288	30.26/0.9387	0.2318
$L_1 \rightarrow L_1 + SSIM$	8.93M	124.36	21.37	36.69/0.9714	0.1246	30.37/0.9403	0.2131
$L_1 \rightarrow L_1 + VGG$	8.93M	124.36	21.37	36.31/0.9711	0.1130	30.37/0.9403	0.2130

greatly out-performed the DeP for the two display types. The domain gap mainly comes from the following aspects. First, due to the existing distances between display and lens, in real data there appears visible patterns of the display on the image plane. We recall in the assumption of the diffraction model, the display panel is exactly at the principle plane of the lens system. The cause of the visible bands are illustrated in the supplementary material. Second, the approximated light transmission rate may not be accurate, the measured values may be influenced by other environment light sources. Third, impulse noises caused by dead pixels or over-exposure in the camera sensors widely exist in the real dataset. Those factors provide more improvement space for the proposed data synthesis model.

4.5.2 Ablation Study

For the best-performed UNet structure, we compare different UNet alternatives in Table 4.5. We increase the parameter size by splitting the original encoders into two sub-encoders, so the performance is also increased. The increment parameter size and inference time is far less than doubling the width of each layer of UNet, but the performance improvement is comparable (T-OLED), even better (P-OLED). We claim that the proposed UNet structure will both maintain a small number of parameters and operations, and achieves a real-time high-quality inference. For more training loss, we add $SSIM$ or VGG loss in additional to L_1 loss with 1:1 ratio. However, the performance gains on either $SSIM$ or perceptual metric LPIPS are not significant enough, and are not visually distinctive. Adversarial loss is not

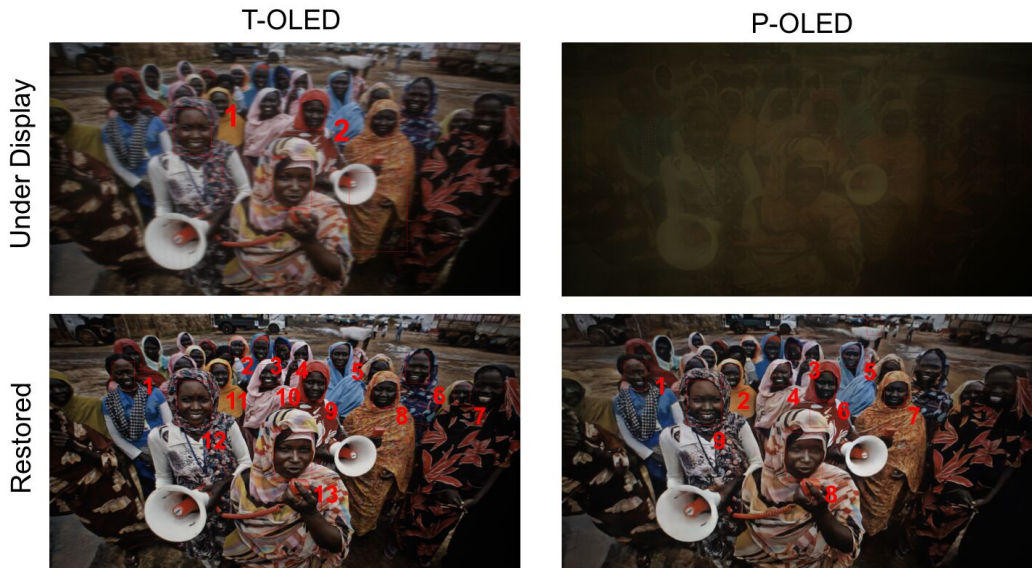


Figure 4.8: Face detection performance before and after applying restoration. Without display, the original face recall rate is 60%. Covering the camera with T-OLED or P-OLED will decrease the recall rate to 8% and 0%. After image restoration, the recall rates recovered back to 56% and 39%.

implemented due to its temporal instability compared to GAN-based training. For complicated problems like UDC, training the model solely with L_1 loss is effective enough for good quantitative and qualitative performance.

4.5.3 Downstream Applications

The proposed image restoration also enhances the performance of downstream applications including face detection. Figure 4.8 shows an example of detecting faces using MTCNN [94]. Without display, the original face recall rate is 60%. Covering the camera with T-OLED or P-OLED will decrease the recall rate to 8% and 0%. After image restoration, the recall rates are recovered to 56% and 39%.

4.6 Ethics Statement

Ethics issues aroused from device cameras are mainly the privacy concerns caused by the recording function and the notification of recording. First, there is no current law that requires companies to tell a user when the user

is being recorded. Because of the lack of any supervisory law, Apple iOS14 introduced a "camera-on" indicator, to tell users when their camera is on; Android does not have any standard camera-on indicator, but users can download third-party apps that will act as camera-on indicators. For camera on the laptop, users can purchase a privacy screen to ensure there is not hidden recording.

The proposed UDC may also have similar concerns, but the hidden-camera design may cause some unique problems. Users may not know where the camera is, and people who do not own the device might not even know that the camera exists. The under-display location makes it more difficult to hide the camera using privacy protection screens, because the same display surface may contain information that the user needs to see. Since there is not a specific law, users should be educated to behave properly and morally with the device. Besides, while designing the device, we should design auxiliary software, indication LED or cell phone cases that are mutually compatible with privacy screens. More efforts should be paid on resolving ethical concerns if the UDC technology can be successfully embedded into devices.

4.7 Conclusion

In this chapter, we defined and presented a novel imaging system named Under-Display-Camera (UDC). Deploying UDC to full-screen devices improves the user interaction as well as teleconferencing experience, but does harm to imaging quality and other downstream vision applications. We systematically analyzed the optical systems and modelled the image formation pipeline of UDC, and both collected real data using a novel acquisition system and synthesized realistic data and the PSF of the system using optical model. We then proposed to address the image restoration of UDC using a Deconvolution-based Pipeline (DeP) and data-driven learning-based methods. The experiments showed that the former achieved basic restoration and the latter demonstrated an efficient real-time high-quality restoration. The model trained with synthetic data also achieved a remarkable performance indicating the potential generalization ability.

UDC problem has its promising research values in complicated degradation analysis. Future work can be exploring UDC-specific restoration models

and working with aperture and display researchers to analyze the influential factors of image degradation. It will make the restoration model generalized for mass production, as an ultimate goal.

CHAPTER 5

COLOR AND SPATIAL TRANSFORMATION FOR REFERENCE-BASED IMAGE INPAINTING

In this chapter, we will extend the topic to real-world image inpainting, as a more challenging task for image restoration due to the more extreme ill-posed issues. We will discuss a reference-based image inpaiting pipeline which aligns and harmonizes two given images, and utilizes the contents from one of them to fill the missing regions of the other. We will demonstrate a robust color-spatial transformation module and pixel-wise merging modules which are adaptive to real-world inputs with different resolution and color-spatial misalignment.

5.1 Introduction

Image inpainting is an image restoration task where the goal is to fill in specific regions of the image while making the entire image visually realistic. The regions to be filled are called hole regions, and could contain undesired foreground objects or small distracting elements, or corrupted regions of the image. Much research has been devoted to improving image inpainting either by image self-similarity (e.g. [95]) or deep generative models (e.g. [96, 97, 98]). Such methods synthesize realistic semantics and textures by reusing similar patches from non-hole regions or learning from large collections of images, respectively. However, those methods still struggle in cases when holes are large, or the expected contents inside hole regions have complicated semantic layout, texture, or depth.

These problems can be addressed if there happens to be a second reference image of the same scene that exposes some desired image content that can be copied to the hole. This task is referred to as *reference-guided* image inpainting in the literature [99], but this topic is less explored. In our paper, we call the image with the hole indicated for removal the *target image*. In

Figure 5.1: Results of our reference-guided inpainting for user-provided images. We show multiple practical applications like replacing and removing foreground people and objects. Each triad shows the target image with the hole, the source image used as a reference, and the inpainting result. Our method has strong performance and addresses challenging real-world issues such as parallax, 90 degree image rotations, and lighting inconsistency between the source and target images.

general, there could be multiple other *source images* used as references. These could be taken by the photographer for the same scene after objects have moved or the photographer moved the camera to a different viewpoint to expose the background. Alternatively, a source image could be collected from the Internet [100]. If one such source image contains new desired appearance for the target hole region, then we can copy the pixels from the source to fill in the target hole regions. In this paper we assume that the user has identified a particular source image with the new desired appearance, so we refer to this as *the source* image. We imagine that dedicated apps might be created for aiding the photographer in this process, or for automatically retrieving suitable such source images from the Internet.

Although the reference source image makes the inpainting task easier, reference-guided inpainting is still quite challenging for several reasons. First, the hole regions could be very large, which makes the task of guessing the pixel colors in the hole region less well-posed. Second, we wish for our task to be as general as possible, so we allow an uncalibrated camera to freely translate to different 3D positions for the source and target image, because this can allow the photographer to intentionally reveal regions behind a foreground object to be removed. Such translations, however, can induce large parallax, which cannot be modeled in image space by a simple 2D warp such as a global homography. Unlike video inpainting or multi-view Structure-from-Motion (SfM), we assume the system will not have access to more than two photos. Thus, it is harder in our setting to reliably estimate 3D structures, depth, and

point correspondences. Third, depending on the camera and photography setup, the photographs may have substantially different exposure, white balance, or lighting environment, and if one photograph comes from the Internet, then it will have different camera response curves. Existing methods based purely on warping cannot resolve the resulting complex issues of color matching. Finally, there may exist regions in the source image that do not exist after warping due to pixels being out of the image or occluded.

To address these challenges, we propose a multi-homography fusion pipeline combined with deep warping, color harmonization, and single image inpainting. We address the issue of parallax by assuming that there may be multiple depth planes inside the hole. Loosely inspired by recent work on multiplane images [101, 102, 103, 104], we propose multiple homographic registrations of the source image to the target, each corresponding to an assumption that the scene geometry lies on a different 3D plane. Given a target and a source image, we first estimate the matched feature points between the two images, cluster the inliers according to their estimated depths in the target image, and for each cluster estimate a single homography to perform an initial image registration. We call each of these candidate alignment images a *proposal*. For each proposal, we then tackle the challenge of color matching by using a deep bilateral color transformation, and we address parallax issues by refining the warp using a learned per-pixel spatial transformation. We then merge all the transformed source image proposals by learning a set of fusion masks. Finally, we address the last challenge regarding regions which do not exist in the source image by using a state-of-the-art single image inpainting method to complete missing regions, and learn to merge it as well.

In summary, the main contributions of our method are: (1) We propose TransFill, a multi-homography estimation pipeline to obtain multiple transformations of the source image, where each aligns a specific region to the target image; (2) We propose to learn a color and spatial transformer to simultaneously perform a color matching and make a per-pixel spatial transformation to address any residual differences after the initial alignment; (3) We learn weights suitable for combining all final proposals with a single image inpainting result.

5.2 Related Work

Image inpainting. Inpainting research can be divided into two categories: traditional methods that work by propagating colors or matching patches, and deep methods that learn semantics and texture from large image datasets.

Some traditional methods propagate pixel colors by anisotropic diffusion [105] or solving PDEs [106]. Such methods work well for thin hole regions but as the hole regions grow larger they tend to result in over-blurring. Patch-based image inpainting methods work by finding similar matches elsewhere in the image and copying the resulting texture [107, 95]. Those methods tend to result in high-quality texture but may give implausible structure and semantics.

Our work is more closely related to deep models for inpainting that use a single image. Context encoders analyze the surroundings of the hole [108], local and global discriminators [109] can improve local texture and overall image layout, and partial [110] and gated convolutions [96] can reduce artifacts from filter responses at the hole boundary.

More recently, some deep methods have focused on inferring other information first: these can be roughly categorized into using edges [111], segmentation masks [112], low-frequency structures [113, 114], and other possible maps like depth. The ill-posed nature [115] of single-image inpainting makes it challenging to complete larger holes and higher-resolution images. Recent works demonstrate neural networks can generate high-resolution images [116, 117, 118], but for large holes these methods can still generate results that appear semantically implausible or have artifacts in the fine-scale texture. Since our method has a source reference image, we can better establish consistency with the ground truth image by learning appropriate spatial and color transformations for a source image patch.

Video inpainting. A few classical works in this area are Wexler [107] and Granados [119], which globally optimize patch-based energies, and Newson [120], which estimates multiple homographies using a piecewise planar assumption for the scene. Xu [121] estimates the optical flow to learn the pixel warping field. Recently, the Onion-Peel Network (OPN) [99] leverages non-local designs inside the network, making it feasible to apply multi-source inpainting for a larger temporal range. Lee [122] proposed a Copy-and-Paste Network to learn the alignment of consecutive frames for

video inpainting. Zhao [123] reuse contents from an unrelated image for a reference-based inpainting. Their method is based on only a single affine transform, which we show is not enough in our experiments, and exhibits residual color and geometric incompatibilities that are problematic in our multi-view scenario. Xue [124] is a specialized method designed to remove reflective or occluding elements near the camera such as fences.

Image harmonization. Image harmonization refers to matching the color distribution and appearance when compositing a foreground from one image on a background from another image. Traditional methods transfer color statistics locally and globally [125, 126, 127] and use gradient-domain based blending [128, 129, 130]. Digital photomontage [131] also demonstrated copy-and-paste workflows that can change the appearance of a foreground subject. Unlike our method, photomontage required user input and assumes the photographs have been aligned. Recently, CNN-based harmonization models [132, 133] have emerged, including methods involving segmentation masks [134] for region selection, and discriminators for domain verification [135]. Deep bilateral filtering has also been used to better preserve edges and details while transforming image color space [136, 137]. Our work is the first to integrate harmonization with a neural network for reference-guided inpainting. We apply a deep bilateral color transformation to address color inconsistencies while preserving edges.

Image alignment. Image alignment or registration involves placing multiple images in the same coordinate system. It is widely used for video stabilization [138], image stitching [139, 140], and serves as an important pre-processing step for many video and image applications like face analysis. Homography warping is a widely used global parametric method. Sparse local features like SIFT [141] can be matched either using nearest neighbour, or deep models like OANet [142] and SuperGlue [143], and the resulting correspondences can be used to estimate warping models. Recently, deep models have been explored to directly learn homography parameters [144, 145, 146], demonstrating their advantages on low-light and low-texture images.

Issues of parallax due to content at different depths can be better addressed by mesh-based warping [138, 147, 148] or pixel-wise dense optical flow [149, 150, 151, 152, 153, 154]. Liu proposed the Content Preserving Warp (CPW) [148] to maintain the rigidity of motions. Recently, Ye proposed deep meshflow [155] to make mesh estimation more robust on different scenes. Due

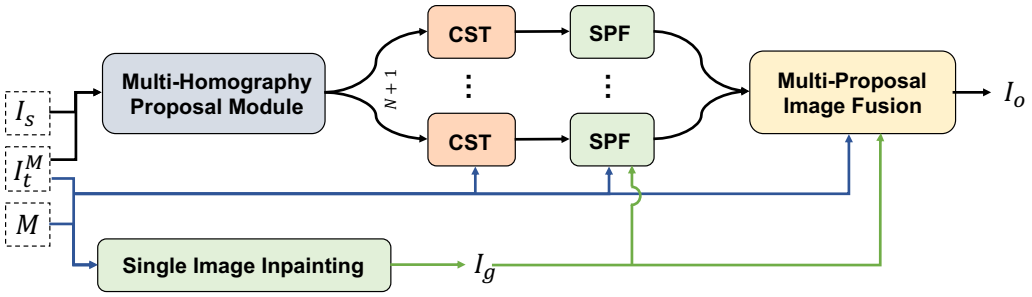


Figure 5.2: System pipeline. Given the target image I_t^M masked by an associated binary hole image M, and a single source image I_s, we first propose multiple global homographies using the multi-homography proposal module, and locally adjust color and spatial misalignments in each proposal using our Color-Spatial Transformer (CST). Then we merge each proposal with the output I_g from a single-image inpainting model using Single-Proposal Fusion (SPF), and finally selectively blend all the proposals.

to the sparsity of the mesh, image contents can be better retained while warping. However, optical-flow based methods can provide greater flexibility in permitted motions. Our pipeline uses multiple global homographies followed by per-pixel warping fields to combine the advantages of various alignment methods.

5.3 Method

We will first give an overview of our pipeline. Suppose we are given a target image $I_t \in \mathbb{R}^{W \times H \times 3}$, an associated mask $M \in \mathbb{R}^{W \times H \times 1}$, and a single source image $I_s \in \mathbb{R}^{W_s \times H_s \times 3}$. Note that M indicates the hole regions with value one, and elsewhere with zero. The masked target image is then denoted by $I_t^M = (1 - M) \odot I_t$. We assume there is sufficient overlap in content between the two images especially nearby (but not necessarily within) the masked regions. Our task is to generate contents inside the masked regions of I_t by effectively reusing contents of I_s. More specifically, we wish to geometrically align I_s with I_t in the vicinity of the hole region globally and locally, and adjust any color inconsistency. We fill any regions that are occluded or outside the image using a state-of-the-art single image inpainting method.

Our pipeline follows four steps as shown in Figure 5.2. It includes an initial registration using multiple homography proposals, per-pixel color and spatial transformations for each proposal, single-proposal and multi-proposal fusion.

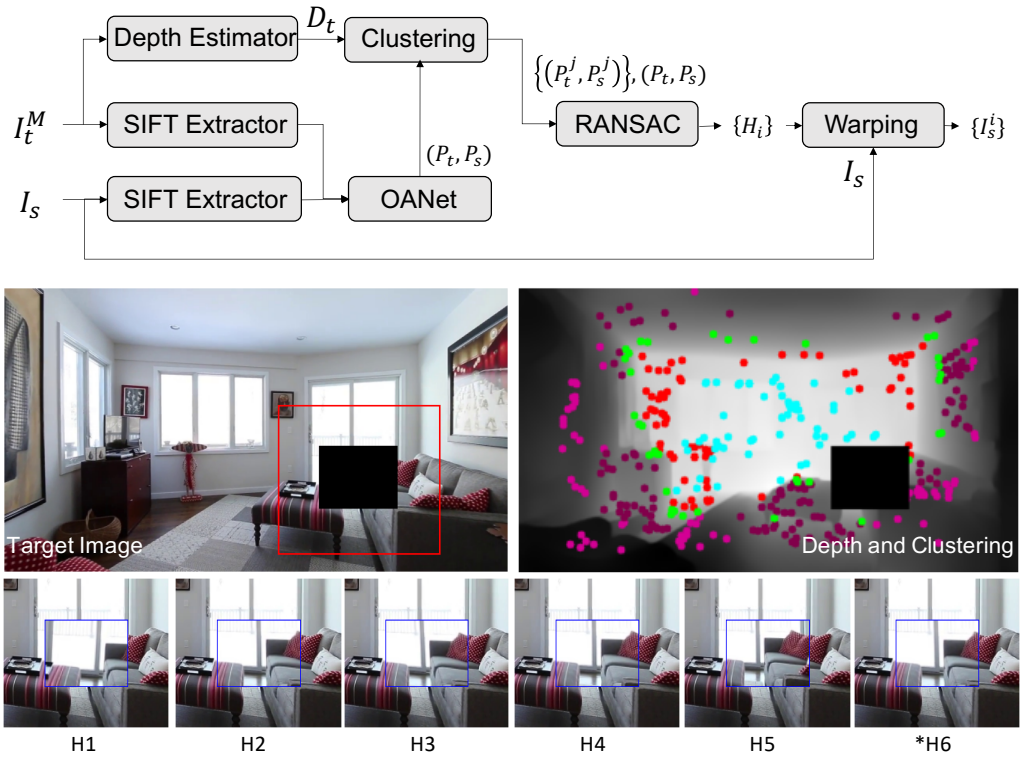


Figure 5.3: Multi-homography Proposal Module. We compute the monocular depth D_t of the non-hole region I_t^M, and cluster the feature matching points into N sub-groups using the depth values. Each estimated homography H_i will align different regions within the hole. $^*H_6$ indicates a homography estimated using all the points.

5.3.1 Multi-homography Proposals

In this stage, we first globally warp the source image I_s to align it with the masked target image I_t^M. Provided the contents inside the hole region occur at multiple depth planes, or the camera motion is not a simple rotation, a single homography is not sufficient to perfectly align the source and target image [156]. Therefore, we propose to estimate multiple homography matrices to transform I_s. Ideally, each homography-transformed I_s can align with I_t within a specific image depth level range or local spatial area, as shown in Figure 5.3.

To obtain different transformation matrices, we first extract SIFT [157] features from I_t^M and I_s, and feed all the extracted feature points and their descriptors into a pre-trained OANet [142] for outlier rejection. The lightweight OANet efficiently establishes the correspondences between I_t^M

and I_s by considering the order of the points in the global and local context. OANet outputs the inliers forming a point set P_t in I_t^M, and its corresponding matched point set P_s in I_s. To consider different possible depth planes within and nearby the hole region, we are inspired by the Multi-Plane Image (MPI) [102] idea for scene synthesis. We estimate the depth map D_t from I_t^M using a deep learning based monocular depth estimator [158] , and record the depth value for each point in P_t. We then cluster those points into a partition of N subsets $\{P_t^j\}, j \in [1, N]$ by their depth values using an agglomerative clustering method [159], where $P_t = \cup_{j=1}^{N} P_t^j$. The corresponding matched points in P_s are used to form the subsets $P_s = \cup_{j=1}^{N} P_s^j$ accordingly.

For each subset's pairs of points (P_t^j, P_s^j), we estimate a single homography using RANSAC [160]. By further including the homography estimated from the full set of points (P_t, P_s), we obtain $N + 1$ homography matrices overall. We denote them by $H_i, i \in [1, N + 1]$. Finally, we transform the source image I_s using the estimated H_i, and obtain a set of warped source images $\{I_s^i\}$, where $I_s^i \in \mathbb{R}^{W \times H \times 3}, i \in [1, N + 1]$. We set $N = 5$ in our experiments.

5.3.2 Color-Spatial Transformation (CST) Module

The global homography-warped source image sets $\{I_s^i\}$ are regarded as the initialization of the warping of I_s. However, as shown in Figure 5.3 and 5.4, while directly compositing I_s^i and I_t^M using $I_s^i \odot M + I_t^M$, due to the possibly inaccurate homography estimation or challenges of large parallax, there may be small misalignments inside and near the hole region, especially along the hole boundary. Additionally, the composite image may suffer from color and exposure differences. Therefore, we propose another refinement step that we call a Color-Spatial Transformer (CST). This simultaneously adjusts the color and alignment for each I_s^i. The structure of CST is illustrated in Figure 5.4. I_s^i will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $\hat{I}_s^i$.

In our design of the color and spatial transformers, we would like to retain the texture details and the rigidity of the source image contents. Additionally, we prefer the color transformation and warping operations to be decoupled and not have to use auxiliary losses for each component. Inspired by deep bilateral filtering [136] and Spatial-Transformer Network (STN) [161], we

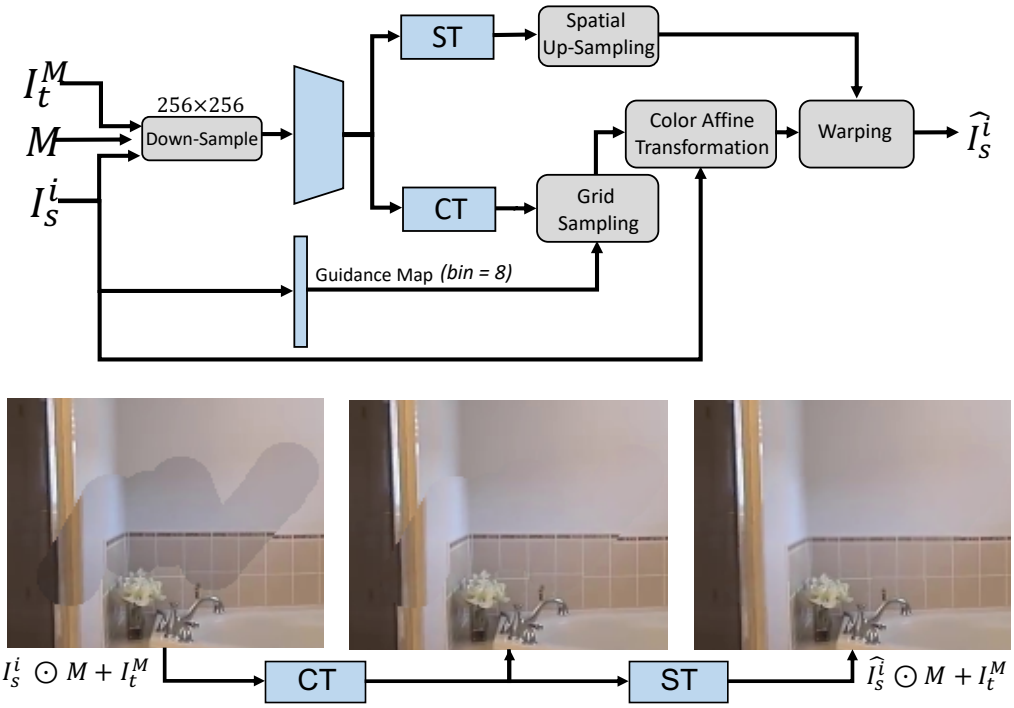


Figure 5.4: Structure of the Color-Spatial Transformer Module. I_s^i will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $\hat{I}_s^i$. The bottom row shows examples of the refinement stages. Blocks with blue color indicate there are learned parameters, otherwise they are parameter-free.

propose to learn the transformations in a lower resolution, and obtain the full-resolution coefficients using up-sampling. Specifically, given I_s^i, I_t^M and M, we down-sample them to 256×256 to obtain $I_s^i \downarrow$, $I_t^M \downarrow$ and $M \downarrow$. Then we compute the high-level features $u_s^i = B(I_s^i \downarrow, I_t^M \downarrow, M \downarrow)$ using a shared network B. After that, the color and spatial transformation coefficients will be learned by the CT and ST sub-networks.

Color Transformation (CT). To transform the color in RGB space of I_s^i to I_{sc}^i, we learn an affine transformation with parameters $A_c^i = [K_c^i \quad b_c^i] \in \mathbb{R}^{W \times H \times 3 \times 4}$. Formally, for each pixel at location p, $I_{sc}^i(p) = K_c^i(p) I_s^i(p) + b_c^i(p)$, where $K_c^i(p) \in \mathbb{R}^{3 \times 3}$ and $b_c(p) \in \mathbb{R}^{1 \times 3}$. To better preserve the edges and textual details, we adopt deep bilateral filtering [136]. Specifically, we learn a bilateral grid $\bar{A}_c^i = B_c(u_s^i) \in \mathbb{R}^{s \times s \times d \times 3 \times 4}$ in a lower resolution, and a single-channel guidance map $g_c^i = G_c(I_s^i) \in \mathbb{R}^{W \times H \times 1}$ in full-resolution. We fix $s = 8$ and $d = 8$ in our experiments. B_c and G_c are the trainable networks for estimating the grid and guidance map. Finally, A_c^i is tri-linearly sampled

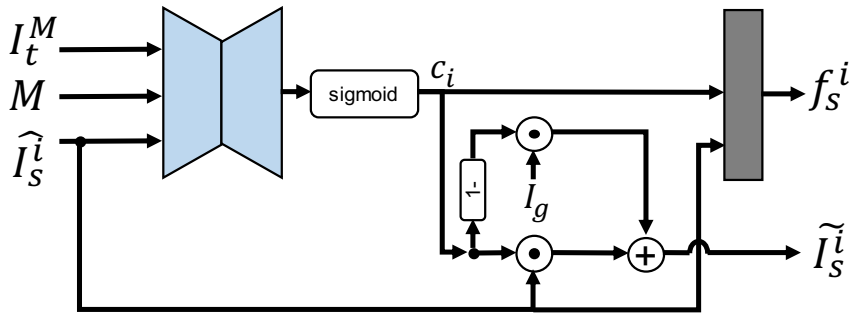


Figure 5.5: Single-Proposal Fusion (SPF) module. This takes I_t^M, M, a single $\hat{I}_s^i$ and I_g as inputs, where I_g is the result of a single image inpainting method. SPF outputs a confidence map c_i, the merged $\tilde{I}_s^i$, and the packed features f_s^i.

from $\bar{A}_c^i$ using the normalized triplet $(x, y, g_c^i(p))$.

Spatial Transformation (ST). We learn the spatial warping offset $A_s^i = [A_{sx}^i \quad A_{sy}^i] \in \mathbb{R}^{W \times H \times 2}$ along the horizontal and vertical axes. To better preserve the rigidity of the image contents inside hole region, we propose to learn the warping field $\bar{A}_s^i = B_s(u_s^i) \in \mathbb{R}^{s \times s \times 2}$ in a lower resolution, and up-sample it to A_s^i using bi-linear interpolation. Finally, $\hat{I}_s^i = \text{Warp}(I_{sc}^i; A_s^i)$. The objective loss to learn the CST module is defined by,

$$\mathcal{L}_{CS}^i = ||M_v \odot M \odot (I_t - \hat{I}_s^i)||_1, \tag{5.1}$$

where $M_v = 1(I_s^i > 0)$ is the valid mask indicating the pixel regions after initial homography warping.

5.3.3 Single-Proposal Fusion (SPF) Module

The Single-Proposal Fusion (SPF) module learns to estimate a confidence map and other features for the refined results $\hat{I}_s^i$ from the CST module by merging it with the outputs of a well-trained single image inpainting model called ProFill [117]. The inpainting results from ProFill often generate good structures, so the intuition for the SPF module is that we independently do an image comparison of each proposal against this ProFill reference, to better constrain the overall learning task and learn confidence and difference features that can help the harder downstream multi-proposal fusion task. As shown in Figure 5.5, the module takes I_t^M, M, a single $\hat{I}_s^i$ and I_g as inputs, where I_g is the output from a single image inpainting method. In this paper,

we use a pre-trained ProFill [117] model and freeze its weights while training the whole pipeline. The module outputs a confidence map of $\hat{I}_s^i$ denoted by c_i of the same spatial size as I_t^M. The output $\tilde{I}_s^i$ of merging is

$$\tilde{I}_s^i = c_i \odot \hat{I}_s^i + (1 - c_i) \odot I_g, \tag{5.2}$$

The values in the confidence map range from zero to one, and higher-valued regions should contain more informative and realistic pixels. The composited result of merging $I_t^M + M \odot \tilde{I}_s^i$ can also be displayed to the user as an intermediate result demonstrating the performance of a single proposal. In our experiments, we will show that compared to learning to merge multiple $\hat{I}_s^i$ directly, it is better to condition on the outputs from a well-learned SPF module.

Additionally, we utilize a shallow convolutional module to concatenate the learned confidence map c_i and the output of the CST module $\hat{I}_s^i$, and output a three-channel feature map f_s^i to be fed into the final multi-proposal fusion module in section 5.3.4. Similarly, when we input I_g to the SPF, we obtain the feature f_g. The objective function for learning the SPF is defined as,

$$\mathcal{L}_E^i = ||M \odot (I_t - \tilde{I}_s^i)||_1, \tag{5.3}$$

and an additional Total Variance loss is imposed on c_i to enforce the smoothness of the map.

$$\mathcal{L}_c^i = \mathcal{L}_{\text{TV}}(c_i), \mathcal{L}_{\text{TV}}(u) = \left\| \frac{\partial u}{\partial x} \right\|_1 + \left\| \frac{\partial u}{\partial y} \right\|_1 \tag{5.4}$$

5.3.4 Multi-Proposal Fusion (MPF) Module

The Multi-Proposal Fusion (MPF) module merges the $N + 1$ proposals of the refined source images $\hat{I}_s^i$ and the single-image inpainting results I_g together. The module is fed with the packed features f_s^i and f_g from the SPF module. Pixel-wise merging weights $\overline{c_i}, i \in [1, N + 1]$ and c_g are learned through a UNet [30] with softmax ($c_g + \sum_{i=1}^{N+1} \overline{c_i} = 1$) by merging different portions of

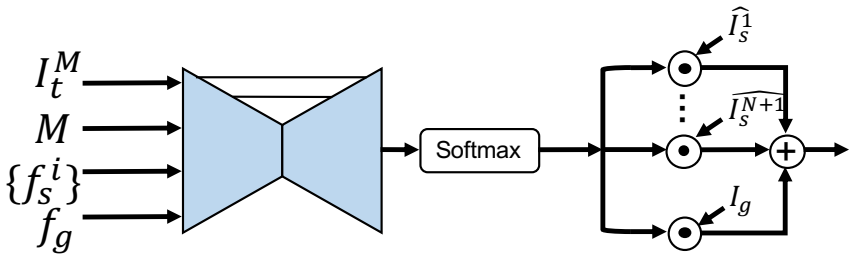


Figure 5.6: Structure of the Multi-Proposal Fusion (MPF) Module. We feed the UNet with packed features f_s^i and f_g from the SPF module, and learn a spatially-varying merging mask for all the proposals.

proposals as I_m,

$$I_m = c_g \odot I_g + \sum_{i=1}^{N+1} \overline{c_i} \odot \hat{I}_s^i, \tag{5.5}$$

Then the final result $I_o = I_t^M + M \odot I_m$ is learned by the objective functions,

$$\mathcal{L}_o = ||M \odot (I_t - I_o)||_1 + VGG(M \odot I_t, M \odot I_o), \tag{5.6}$$

where the VGG loss matches features of the pool5 layer of a pre-trained VGG19 [162]. Similarly, total variance losses are imposed to the weighting maps $\overline{c_i}$ and c_g, so we have the losses $\mathcal{L}_{\overline{c}}^i = \mathcal{L}_{\text{TV}}(\overline{c})$ and $\mathcal{L}_c^g = \mathcal{L}_{\text{TV}}(c_g)$. Therefore, the overall loss function with $\lambda_1 = 1, \lambda_2 = 1$ becomes

$$\mathcal{L}_{all} = \mathcal{L}_o + \lambda_1 \mathcal{L}_c^g + \textstyle\sum_{i=1}^{N+1} (\mathcal{L}_{CS}^i + \mathcal{L}_E^i + \lambda_2 (\mathcal{L}_c^i + \mathcal{L}_{\overline{c}}^i)). \tag{5.7}$$

5.4 Experimental Results

In this section, we present our dataset, implementation details, and quantitative and qualitative results.

5.4.1 Datasets and Implementation

Datasets. We trained the model on the RealEstate10K dataset [102]. This was collected from YouTube videos labelled as real estate footage. In total it consists of more than 8000 video clips with length from 1 to 10 seconds. For each clip, we randomly sampled pairs of images with a displacement of

10, 20, and 30 frames apart. We call this "Frame Displacement" (FD). This resulted in 188184 frame pairs for training, and 20290 pairs for testing. We generated random free-form brush-and-stroke holes like in DeepFillv2 [96]. We also collected 3K more pairs of real user-provided image pairs to serve as practical user cases for testing.

For training the Color-Spatial Transformer (CST), although RealEstate10K contains sufficient samples with real multi-view data and different exposures across image pairs, it lacks image pairs with large color inconsistency. Therefore, we synthesized misaligned color-different images from the MIT-Adobe5K dataset [163], and uniformly mixed these data with RealEstate10K for training. Adobe5K contains 5000 images, and for each image it provides five additional expert-retouched images to form 5000 sets in total. We regard the original samples as target images and synthesized the misaligned source images using the method in [144]. We make two binary variables for whether there is a color difference (C) and whether there is spatial misalignment (S), and synthesized pairs with CS, $C\bar{S}$, $\bar{C}S$ and $\bar{C}\bar{S}$ with equal probability from 4000 sets to form a balanced training set, leaving 1000 sets for validation.

Implementations. We obtained a pre-trained OANet [1] model for image feature matching and outlier rejection. We applied the pretrained model[2] of Hu [158] to estimate the depth map from a single target image. We also obtained a pre-trained ProFill [117] from the authors. All the above-mentioned model weights were frozen during training. Additionally, we pre-trained the CST module using the mixed dataset in advance for 400 epochs, and froze its weights afterwards. Finally, the whole pipeline was trained end-to-end for 400 more epochs. We used a patch size of 256×256 for training and arbitrary size for inference, and a learning rate of 10^{-4} with decay rate 0.5 after 200 epochs. We used the Adam optimizer [164] with betas (0.9, 0.999). The code is implemented in PyTorch [165].

5.4.2 Baseline Models

We chose baselines that are similar to, but may not exactly the same as our task, including approaches addressing image stitching [166], optical flow-guided video inpainting [121], non-local patch matching for multiple photo

[1]OANet: https://github.com/zjhthu/OANet
[2]Hu et al.: https://github.com/JunjH/Revisiting_Single_Depth_Estimation

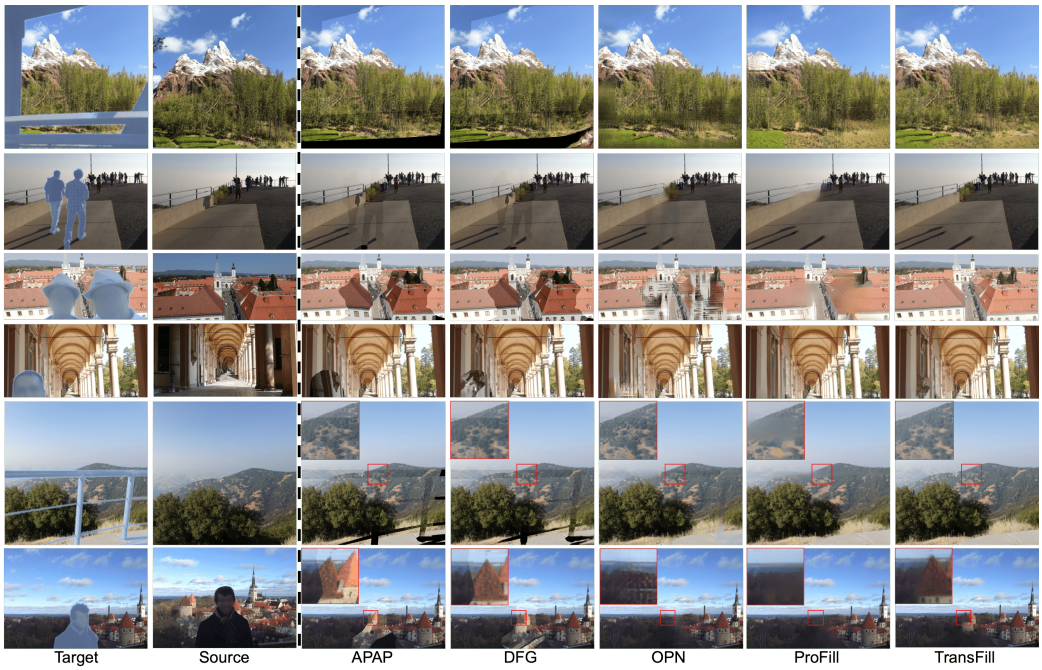


Figure 5.7: Comparison with baselines on challenging user-provided image pairs. For better visualization, we only crop the regions of interest from the whole target and source images. Please zoom in to see the details.

inpainting [99], and a state-of-the-art single image inpainting method [117] with the reference image concatenated so the method has access to the same inputs as the rest.

APAP [166]: As-Projective-As-Possible is a baseline image stitching algorithm that resolves depth parallax. We used the official Matlab [3] implementation for testing.

DFG [121]: Deep Flow-Guided Video Inpainting treats video inpainting as pixel propagation. It fills the holes by completing the optical flow field estimated by FlowNet2.0 [154]. We used their official[4] pre-trained model for testing.

OPN [99]: Onion-Peel Network is a recent work addressing video and group photo inpainting using non-local attention blocks. We used their official PyTorch code[5].

ProFill [117]: ProFill is a state-of-the-art single-image inpainting method that also contains a contextual attention module [98]. We used the official

[3]APAP: https://cs.adelaide.edu.au/~tjchin/apap/
[4]DFG: https://github.com/nbei/Deep-Flow-Guided-Video-Inpainting
[5]OPN: https://github.com/seoungwugoh/opn-demo

pre-trained model [6] from the authors. When testing, we fed in the target with the homography-warped source image. Before testing on RealEstate10K, we also fine-tuned OPN and ProFill on RealEstate10K training frames for fairness.

5.4.3 Qualitative Comparison

Visual Results on User-Provided Images. In Figure 5.7, we show visual results of testing on real user-provided images. We indicate the hole region on the target image, and crop only the region of interest due to the space limits. More results can be found in the appendix. APAP and DFG well-preserve the source image contents due to the global homography warping, but they still suffer from color inconsistencies and alignment issues. We also experimented with combining Poisson blending with APAP but found it gives color bleeding artifacts: see the appendix for details. OPN usually works well when there are multiple reference frames which have similar scales and color distributions within the same video clips. However, if only one source reference image exists, the non-local attention module struggles to search for similar local patches and fails. ProFill with the contextual attention module usually does well in searching for textures, but the estimated intermediate coarse results cannot be matched with specific image contents. Thus the reference-based ProFill can only achieve texture or object removal but not background contents recovery. Compared to them, ours better reuses the background patterns and achieves a content-aware alignment and composition. The generated results are more faithful to and compatible with the target image. The multi-homography proposal approach provides more options for warping initialization when the matched features are too complex for a single homography. It helps to resolve challenging cases when the hole regions do not belong to the dominant plane in the image as shown in row four. Additional higher-resolution results can be found at the following link: Additional Results.

[6] ProFill: https://zengxianyu.github.io/iic/

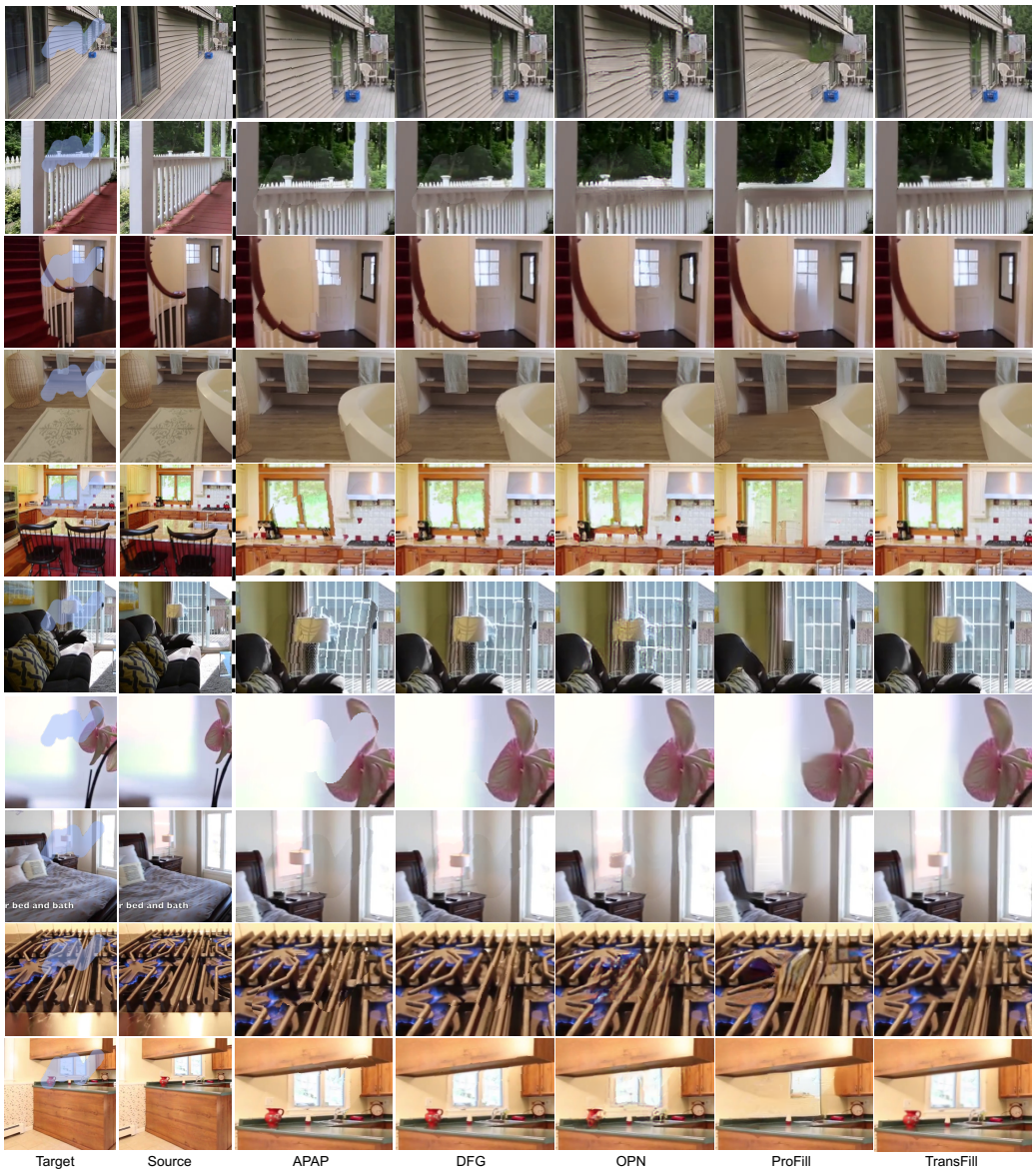


Figure 5.8: Visual results comparison on the RealEstate10K dataset with FD=10. These have been cropped. **Please zoom in so that there are about 3-4 images across the width of the screen to reveal the significant differences in fine details.** Compared with the baselines, our proposed TransFill achieves better spatial alignment and faithfulness to the source image content.

Visual Results on RealEstate10K We present more visual results in Figure 5.8 on the RealEstate10K dataset. Compared with the baselines, our proposed TransFill achieves better spatial alignment and content faithfulness.

Visual Results on Synthetic Adobe-5K In Figure 5.9, we show more results on the synthetic Adobe-5K dataset to evaluate the performance of our color transformation. As stated in the paper, we synthesize misaligned and color inconsistent images from Adobe-5K dataset. The spatial transformation is a simple homography-based warping, so the CST module works well to align the images and match the color. More challenging cases can be visualized in user-provided images.

Unfolding the Model: Intermediate Results In Figure 5.10 and 5.11, we unfold the whole pipeline of TransFill to visualize the intermediate results of each proposed module. We demonstrate the process of image completion in a more intuitive way. After proposing different homography-warped images, the CST effectively adjusts the misalignment and color mismatching. Then the proposed TransFill fills in the holes by selectively merging the well-aligned and color-consistent regions from different proposals. Imperfect regions are finally filled with the output from ProFill.

Figure 5.9: Visual results on the synthetic Adobe-5K dataset. For each group of photos, the left one is the composition of I_t^M and I_s. We transform the color and warp I_s to make it consistent with I_t^M and composite them as the right image. Our CST module resolves the color mismatches and spatial misalignment problems.

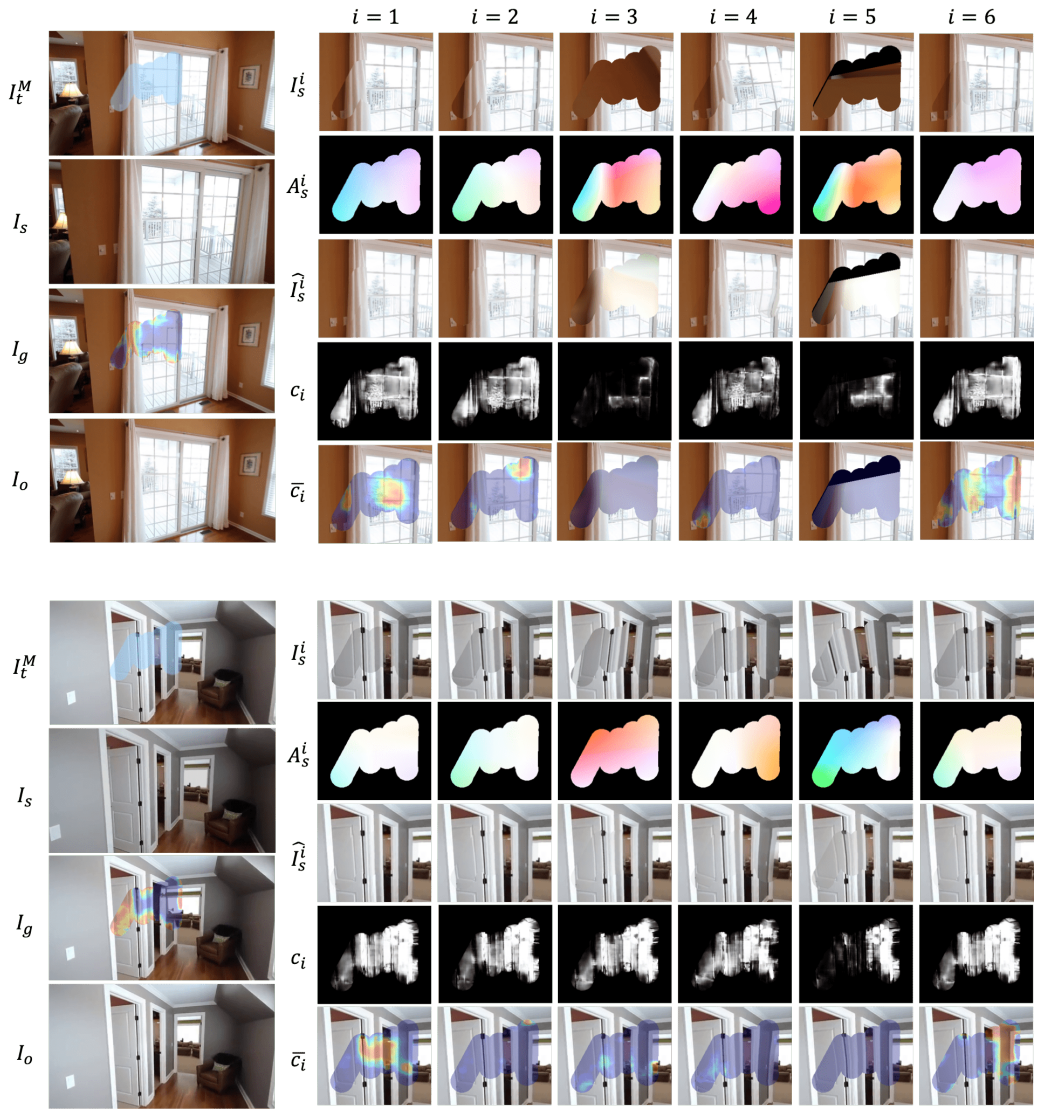


Figure 5.10: Unfolding the whole pipeline to visualize the intermediate results of each module.

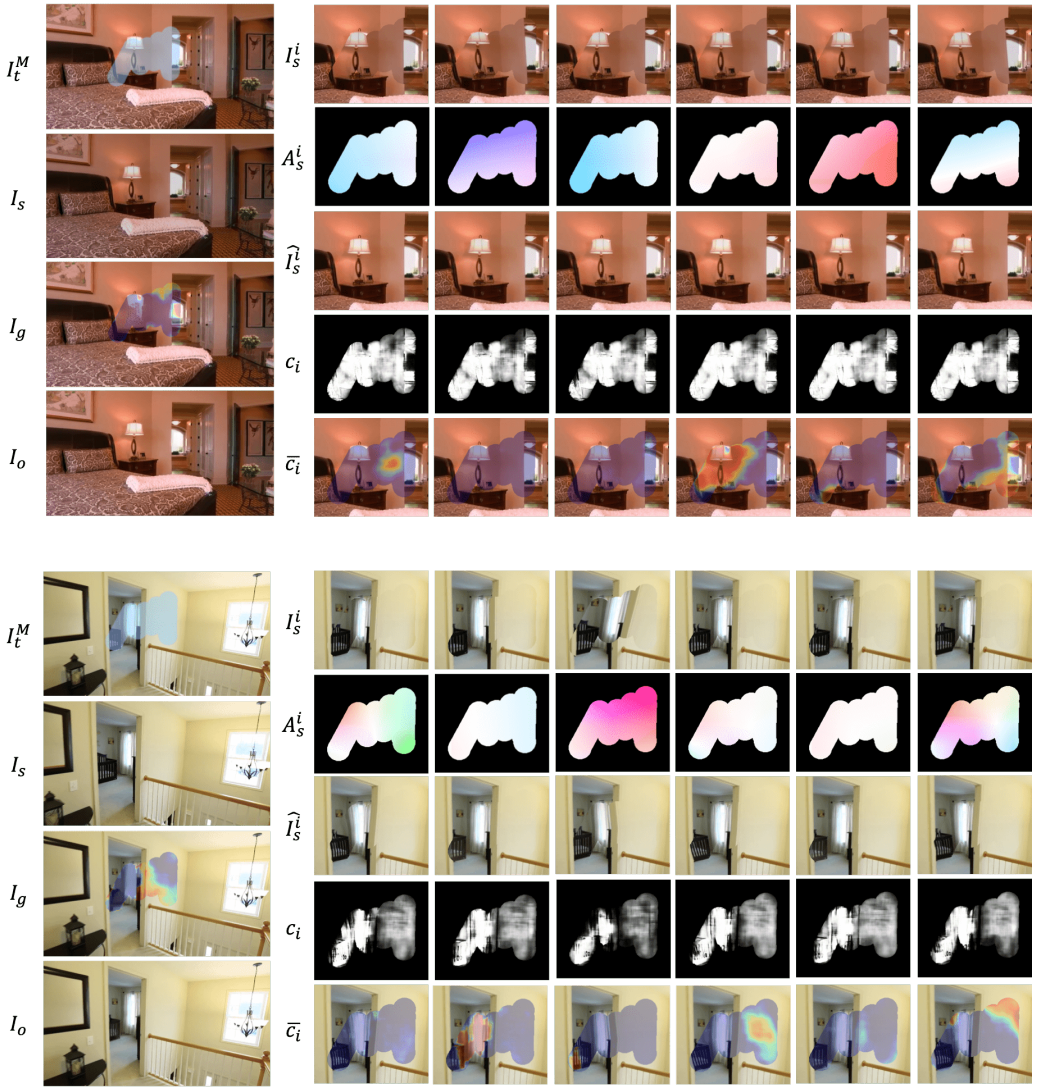


Figure 5.11: Unfolding the whole pipeline to visualize the intermediate results of each module. For some challenging cases when the line alignment is hard, our model can also leverage the outstanding performance of line generation of ProFill to synthesize the door frame.

5.4.4 Quantitative Comparison

Results on RealEstate10K. The quantitative comparison on RealEstate10K is shown in Table 5.1. OPN and ProFill are more suitable for large batch testing. We tested them on the entire testing set. Results on cropped image pairs with Frame Displacement (FD) 10, 20 and 30 are reported in terms of PSNR, SSIM and LPIPS scores [167] based on AlexNet [168]. APAP and DFG are not suitable for large batch testing and their performance may be influenced by non-existing regions, so we sampled a 300-image subset from FD=10 as *Small Set* to test. Results showed that contextual-attention based ProFill failed to faithfully reconstruct the source contents. Optical-flow based DFG achieved better results by smoothly completing the flow field. OPN with atomic patch matching was not better than our warping-based approach. The TransFill thus demonstrated its superiority in faithful reconstruction.

User Study on User-Provided Images. To better evaluate the performance on our user-provided images, we conducted a user study via Amazon Mechanical Turk (AMT). We compared our method with each baseline separately and presented users with binary choice questions. We requested the users to choose one fill result which looks more realistic and faithful. To guarantee the reliability of the users' feedback, we require the users to take a qualification test before they evaluate. The test presents users with the 10 trivial pairs I_t and I_t^M and users who answer correctly more than 8 questions are approved to take the official test. We also mix 10 random sanity check questions with the real questions. No users had to be disqualified due to failing the initial test, and only very few users (4 users) got check questions later in the study wrong (5.7% of total opinions), so we conclude that the user responses are reliable.

For each method pair, we randomly sampled 80 examples, and each example was evaluated by seven independent users. For each sample, one method was regarded as "preferred" if at least five users selected it. Samples voted by three or four users are considered confusing samples and filtered out. We reported TransFill's Preference Rate (PR) in Table 5.1. The high preference rate demonstrates the effectiveness of TransFill. We also conducted a one-sample permutation t-test with 10^6 samples by assuming a null hypothesis that on average 3.5 users prefer one method. The p-values are all sufficiently small so the preference for our method was statistically significant.

Table 5.1: Quantitative Comparisons **FD**: Frame Displacement. The three rows of each cell show the numbers of PSNR, SSIM and LPIPS score.

	RealEstate10K: PSNR ↑/ SSIM ↑ / LPIPS ↓				
Model	FD=10	FD=20	FD=30	All	Small Set
APAP [166]	-	-	-	-	31.94
	-	-	-	-	0.9738
	-	-	-	-	0.0251
DFG [121]	-	-	-	-	36.17
	-	-	-	-	0.9873
	-	-	-	-	0.0155
OPN [99]	33.45	32.47	31.32	32.43	33.40
	0.9765	0.9734	0.9699	0.9734	0.9771
	0.0201	0.0258	0.0320	0.0261	0.0207
ProFill [117]	31.18	31.14	30.83	31.05	30.95
	0.9689	0.9687	0.9683	0.9687	0.9690
	0.0423	0.0425	0.0440	0.0429	0.0419
TransFill (Ours)	**39.59**	**37.39**	**35.62**	**37.58**	**38.83**
	0.9919	**0.9877**	**0.9839**	**0.9879**	**0.9914**
	0.0116	**0.0162**	**0.0213**	**0.0164**	**0.0126**

5.4.5 Ablation Study

Type and Number of Multi-Homography Proposals. This ablation study was conducted on the testing set of the RealEstate10K. For each alternative, we re-trained the model. We compared the proposed depth-based points clustering methods with other alternatives including random and spatial clustering in Table 5.3. When we proposed five homography matrices, depth-based clustering works best. The results were fairly close when we set N to either 3 or 5, but $N = 5$ was slightly better in PSNR. However, these are much better than using just one global homography.

Color-Spatial Transformation Module.

Table 5.4 shows that the order of the Color-Spatial Transformer did not make too much difference. However, according to our experiments, adjusting the color first made the training converge faster since the guidance map was computed from a fixed I_s^i. Table 4.4 also demonstrated that both the Color and Spatial Transformer were necessary.

Recall that while introducing the Color-Spatial Transformer, we intend to preserve the texture details and the rigidity of the source image contents.

Table 5.2: User Study. **PR**: Preference Rate

	User-provided Images: User Study	
Model	PR	p-value
APAP [166]	90.76%	$p < 10^{-6}$
DFG [121]	87.50%	$p < 10^{-6}$
OPN [99]	95.65%	$p < 10^{-6}$
ProFill [117]	81.67%	$p < 10^{-6}$
TransFill (Ours)	-	-

Table 5.3: Ablation Study on Multi-Homography Proposals.

Clustering	N	Outlier Rejection	PSNR↑	SSIM↑	LPIPS↓
Depth	N=5	OANet	**37.576**	**0.9879**	0.0164
Depth	N=5	Ratio Test [157]	37.444	0.9876	0.0168
Random	N=5	OANet	37.499	0.9873	0.0166
Spatial	N=5	OANet	37.384	0.9876	0.0169
Depth	N=3	OANet	37.537	0.9878	**0.0162**
None	N=1	OANet	37.092	0.9868	0.0172

Therefore, given $A_c^i = [K_c^i \quad b_c^i] \in \mathbb{R}^{W \times H \times 3 \times 4}$, and $\bar{A_s^i} = B_s(u_s^i) \in \mathbb{R}^{s \times s \times 2}$, we fix $s = 8$ and $d = 8$ in our experiments. We find d does not influence the performance a lot, and the guidance map is automatically learned to uniformly span the necessary bins like in the HDRNet[136]. Figure 5.12 shows the comparison when we set different s values. It suggests that increasing s gives more degrees of freedom to the learned warping field A_s^i. However, while encountering larger holes like in Figure 5.12, better flexibility does not better align the contents as expected, but distorts the contents inside the hole. The transformed color field also becomes less smooth as s increases. In an extreme case, suppose we replace the deep bilateral grid and directly learn a full-resolution pixel-wise color-warping field with total variance constraints as in the last column, the model struggles to infer a reasonable color-warping operation within a large hole.

We conclude that CST with smaller s value like $s = 8$ generalizes better to inference images with varying spatial resolutions. It is mainly due to the ill-posedness of image completion. Unlike conventional image registration tasks where all the pixels of the matched regions are available, hole regions are missing in the inpainting task. Less freedom in the hole area preserves better content integrity and semantics.

Table 5.4: Color-Spatial Transformation. **C**: Color, **S**: Spatial

Order	PSNR↑	SSIM↑	LPIPS↓
$C \rightarrow S$	**37.576**	**0.9879**	0.0164
$S \rightarrow C$	37.566	**0.9879**	**0.0163**
Only S	36.717	0.9866	0.0182
Only C	36.228	0.9849	0.0179

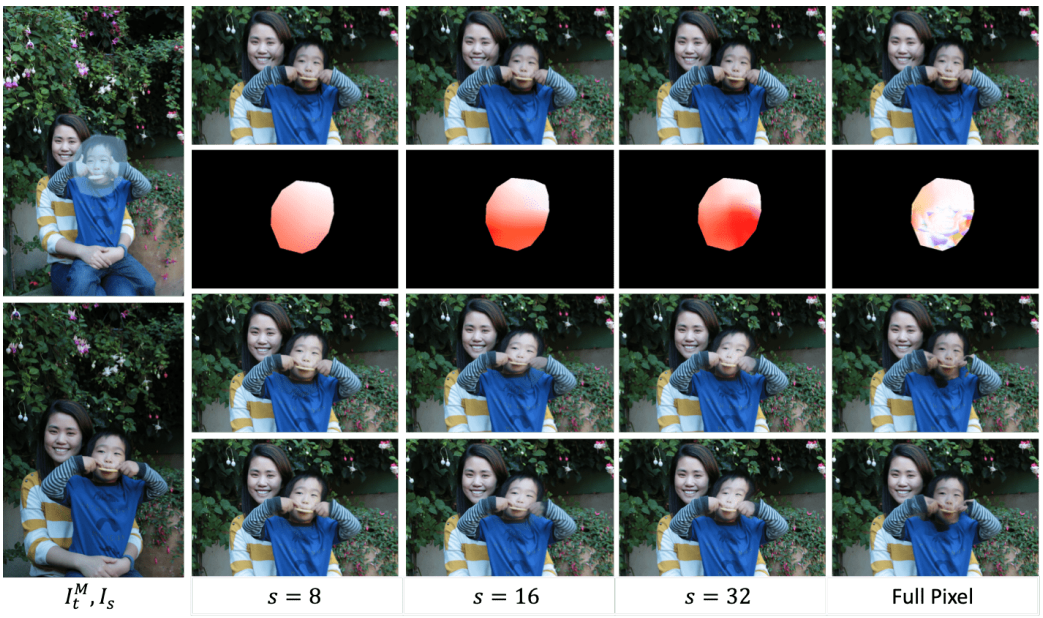


Figure 5.12: Ablation study on the resolution setting in the Color-Spatial Transformer. First row, direct composition of target image I_t^M and one of the single-homography transformed source images I_s^i. Second row, the learned pixel-wise warping field A_s^i visualized using color wheel in [169]. Third row, the color-spatial transformed image $\hat{I}_s^i$. Last row, the final merging result I_o.

Pipeline Components. Table 5.5 indicates that refining the source image with CST outperforms directly merging the initialized homography-warped images. SPF and its output confidence c_i effectively guided the learning of MPF. The proposed full pipeline achieved the best performance.

Importance of Single-Proposal Fusion (SPF) Our experiments exhibit that the proposed Single-Proposal Fusion (SPF) module before the Multi-Proposal Fusion (MPF) is necessary for effectively learning the final merging weights of all the proposals. We find directly learning the weights to fuse all the proposals is very challenging. The learned weights have a hard time becoming sparse even though the same total variance loss is imposed. A comparison of the merging mask $\bar{c}_i$ between the model with and without SPF

Table 5.5: Ablation Study on Pipeline Components. **CST**: Color-Spatial Transformer, **SPF**: Single-Proposal Fusion.

CST	SPF	PSNR↑	SSIM↑	LPIPS↓
✓	✓	**37.576**	**0.9879**	**0.0164**
	✓	35.579	0.9838	0.0183
✓		36.710	0.9861	0.0188
		33.484	0.9782	0.0249

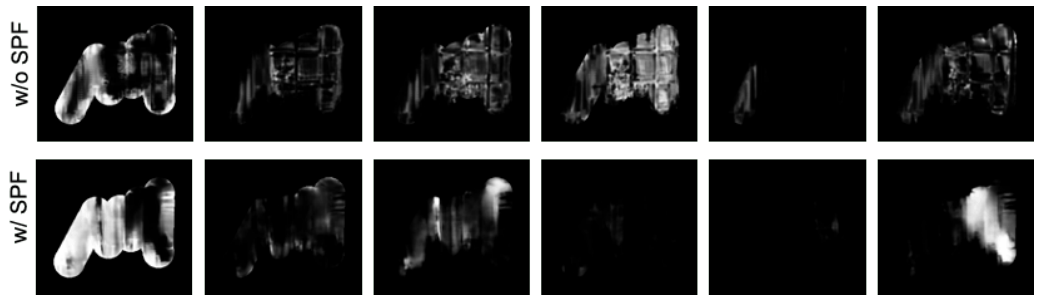


Figure 5.13: Final fusion masks $\bar{c}_i$ learned by the model with or without the Single-Proposal Fusion (SPF) module. By using SPF outputs as guidance to learn the MPF, the final weights learned tend to be more sparse.

is shown in Figure 5.13. Using SPF outputs c_i as a structure guidance for learning the fusion of multiple proposals works better in practice.

Correlation between c_i and $\bar{c}_i$ In our experiments, the learned single-proposal fusion mask c_i and multi-proposal fusion mask $\bar{c}_i$ demonstrate strong correlation. Specifically, by zeroing out one of the c_i, the values in $\bar{c}_i$ will also vanish. This shows the MPF constructs the correspondence to make $\bar{c}_i$ be conditioned on c_i. This provides more flexibility for our model to incorporate user interactions. Suppose users want to eliminate the elements in some proposals, one can simply zero them out and the final results will only be merged from other selected proposals. Such a process is demonstrated in Figure 5.14.

APAP with Poisson Blending We experimented with using Poisson blending [128] combined with APAP. The testing result on the *Small Set* of images with only few non-existing regions is increased from 31.94dB / 0.9738 to 32.56dB / 0.9754 in terms of PSNR / SSIM. However, we did not incorporate Poisson blending in the baseline because we found in some cases there could be significant color bleeding artifacts due to strong color mismatches and non-existing regions especially along the boundary of the hole. Some visual comparisons are shown in Figure 5.15.

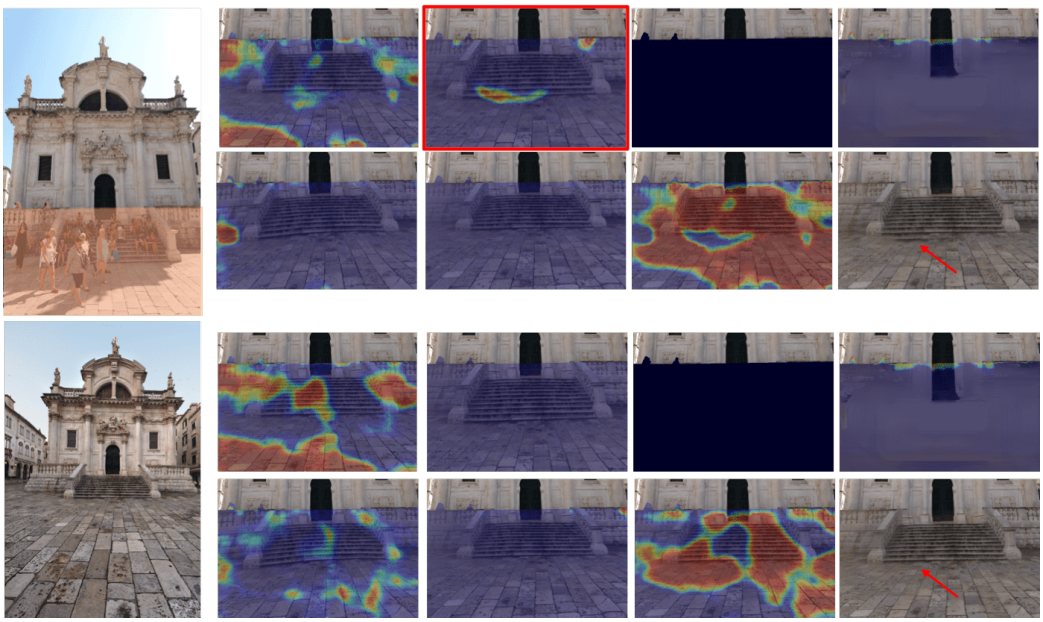


Figure 5.14: One example of user-involved interactive editing. For the given target and source images on the left, we generate seven proposals including one I_g from ProFill. For the upper group of images, we visualize the regions selected by our model to synthesize the final results. The image with red bounding box yields an unexpected artifacts of stairs. By zeroing out its corresponding c_2, we can correspondingly obtain zero-valued $\bar{c}_2$ as shown in the lower image group. Other maps are also correspondingly redistributed. The final result on the lower-right position is then generated by merging the other selected proposals with nonzero weighted masks. As we can see, the artifact disappears.

Using ProFill with Partial Masks As we stated that single image techniques don't work well for larger holes, while in our work, the single-image inpainting is computed over the full mask area. We also thought about using ProFill or other single-image inpainting method with partial mask, but could not find a principled and an end-to-end way to do this. However, we analyzed an approximation of this approach where we used the confidence map c_g estimated by our method, and binarized it to do a post-hoc fill (with ProFill) of each hole region of the target that corresponds to single image inpainting (where the content is not visible in the source image or not well reused). Comparisons are shown in Figure 5.16. This reveals that since the mask was learned for merging purposes, a post-hoc filling using the mask may introduce other artifacts like broken door frames. The average testing results on RealEstate10K decreased from 37.58 dB / 0.9879 / 0.0164 to 37.13

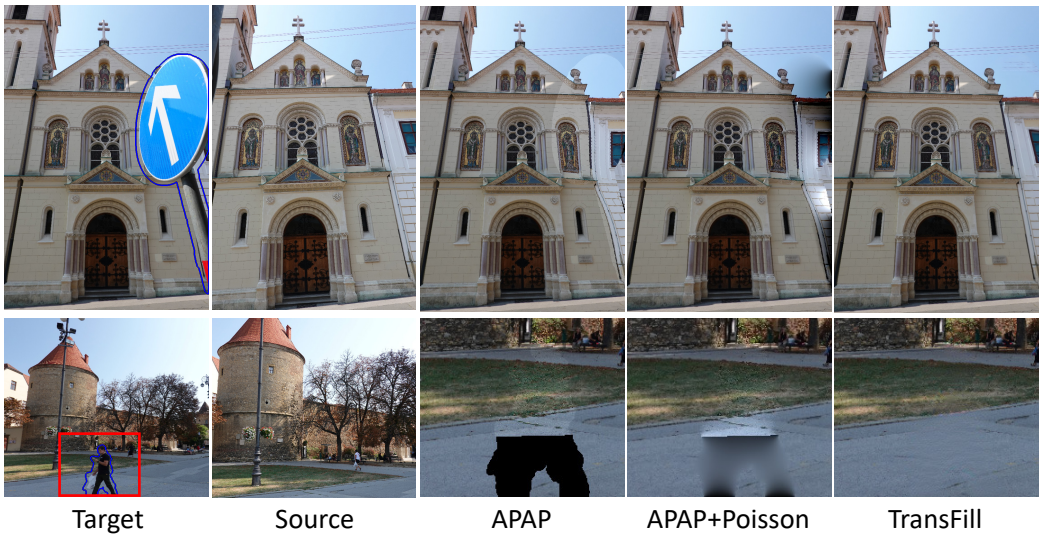


Figure 5.15: Ablation study on APAP with Poisson blending post-processing. The color bleeding artifacts are significant in some cases when there are strong color mismatches or non-existing regions. APAP does not include image inpainting, so regions that are outside the source image appear as black.

dB / 0.9871 / 0.0173 in terms of PSNR / SSIM / LPIPS. However, using partial masks to fill only non-existing regions might work better for images with larger non-existing regions, and become more robust if another approach of learning is taken.

5.5 Failure Cases

Figure 5.17 shows some examples of failure cases when the viewing angle changes are large. The color matching module may struggle if there are extreme lighting differences. We may also encounter outpainting artifact issues caused by ProFill.

5.6 Ethics Statement

The goal of image inpainting research is to generate realistic image contents while removing or changing some original contents of the photos. Our model achieves a high preference rate in the user study due to its high performance

Figure 5.16: Post-hoc refilling results using ProFill. The TransFill columns show a zoom of the original output, the post-hoc filling result (TransFill + Post), and the region to be refilled from the confidence c_g (Binary Mask). The re-filling with a partial mask may introduce additional artifacts like broken door frames.

of reusing existing contents from another image. However, we believe that there are scenarios where our model can be used in a wrong way. First, users may remove some objects from the image to make fake contents, and our generator based on ProFill may also generate biased contents.

countermeasures against the falsification or bias of images may include both technical measures and legal measures. **Technical measures** may include: first, associated with our generator, we also train an adversarial discriminative detector to differentiate fake generated images with real photos. We can also add watermarks in original photographs that could be decoded to determine whether or not the image has been modified. Second, though we did not find explicit bias issues in the model results, in a future development, we will consider building up a more balanced training dataset to better avoid similar issues. **Legal measures** include the draft U.S. bill H.R. 3230 [170], which would require any visual display that modifies an image of a person, if the user knows that the image will be distributed over the internet, to prominently display a notification stating that the image has been digitally modified. Given the importance of those ethical problems, we encourage more explorations towards this direction in the future.

Figure 5.17: Failure cases. These demonstrate limitations with large changes in viewing angle, outpainting artifacts from the off-the-shelf single image inpainting module ProFill, and challenges in handling dramatically different lighting environments.

5.7 Limitations, Discussion and Conclusions

The proposed method has limitations in certain situations. First, the pipeline may not work well on extreme low-light or texture inputs containing very few SIFT feature points. Second, our homography-based transformation is not suited for image pairs with extreme viewpoint changes. Third, the current model may struggle to transfer color if the lighting environment is very different, such as day to night. This is because we use an effective bilateral grid color matching, but do not incorporate any specialized models that reason further about lighting (e.g, [171]). Additionally, we utilize the pre-trained ProFill to fill the missing pixels so the final generation quality highly depends on the performance of the single image inpainting module ProFill. That module could be replaced by other state-of-the-art models, and could potentially be optimized with the multi-fusion pipeline together. We leave that for future work.

In conclusion, we contribute a multi-source image inpainting model based on multiple homography, deep warping and color harmonization. The results outperform state-of-the-art single image and multi-source inpainting methods, especially when the hole contains complicated depth.

CHAPTER 6

DISCUSSION

In this chapter, we discuss the philosophy about practical image restoration and related tasks, and build up the connections among different chapters. We mainly present the goals of making deep restoration models practical and the ways of deploying those models for large-scale applications, and discuss the current exploration in this paper towards the goals. The limitations will be analyzed and future work for more innovative methods and applications are anticipated.

6.1 Practical Deep Image Restoration

In practical applications, image restoration is supposed to have two main characteristics: First, models should be responsive and effective enough when the testing data contains complicated degradation factors, even unknown degradation types and levels. Second, models should have high efficiency with short inference time. Compared with conventional optimization-based methods or inverse filtering algorithms, deep learning-based restoration models, as a new trend, learn a forward pass for inference, bringing in faster estimation speed and much better performance within specific training data domains whose images can be distorted by arbitrary degradation factors.

However, two drawbacks are also very obvious. First, due to the data-driven nature of deep learning-based methods, a convolution neural network model is domain specific. The performance of the model is constrained by the quality and size of the training dataset, and the domain difference between the traniing and the testing data. Given any unseen testing data whose domain is slightly different from the one of the training data, the model will experience a severe performance drop. It brings a problem of limited model scalability and adaptability. Suppose the industrial applications require a

de-blurring model which can handle testing images with arbitrary and diverse blur kernels, or a de-noising model which can successfully enhance the quality of noisy user photos shot from unknown cameras, it will be infeasible to customize a deep model for a specific testing input. Second, for deep models, larger parameter size and deeper network structures usually yield better performance, but it is actually not necessary. Performance and efficiency of using smaller models can be still improved by model compression, neural architecture search, network design, convolutional kernel design or any other advanced methodology. Balancing the performance and efficiency of deep models are constantly an open problem and emerging research area. The related knowledge from other vision tasks can be shared with image restoration models.

In this dissertation, towards more practical deep image restoration models, we claim the key to resolving the drawbacks of the models are the data itself along with task-specific knowledge, and domain-adaptive deep model design. In our experiments, the training and testing data may have domain difference like different Bayer patterns in Chapter 2 or distinctive spatial and channel noise distribution in Chapter 3. Besides, the restoration task can be too novel and no training data is available and it requires us to synthesize realistic data in Chapter 4. Furthermore, testing and training data may have different spatial size while naïve CNN fails to estimate resolution-sensitive information like warping offset, regional color transformation or degradation kernel etc.

We found that simply training a naïve CNN using the prepared training data from a specific domain results in degraded model performance during testing. However, inputting the common knowledge from respective tasks, which is mostly ignored by deep learning researchers and engineers, could greatly boost the model performance. Besides, simple model design can improve the adaptability by narrowing down the gap between training and testing. The exploration among the above mentioned methodology finally forms this dissertation.

6.2 Exploration

In this dissertation, towards more practical deep image restoration models, we mainly explored and discussed the methodology of real data acquisition

(Chapter 4), realistic data synthesis (Chapter2 and 4), testing data domain adaptation (Chapter 2 and 3), and deep model design (Chapter 3 and 5).

Specifically, for RAW image processing task in Chapter 2, unifying the Bayer pattern of the RAW images was found to be a highly useful trick while previous researchers just directly feed data with mixed patterns into the deep models. Our proposed training strategy is proven to be effective while coping with RAW data and can easily improve the robustness of deep models.

Similarly, in Chapter 3, we found the major domain difference between Gaussian-corrupted images and most real RGB noisy images captured by users sources from the spatial and channel correlation of the noise patterns, which are generated by interpolation process of demosaicing algorithms with in camera ISP. However, before that, very few research works of deep image denoising studied real RGB noises or tried to discuss the domain difference by unfolding the camera pipeline. They simply use toy examples like Gaussian or Impulse noise model and study network designs. Though they obtain fairly high quantitative performance on synthetic data, the models can not be deployed for practical applications because most user images are not Gaussian-corrupted. Our proposed method greatly boost other previous deep learning models which are trained on Gaussian noises by a large margin, and the simple adaptation module makes previous models much more practical.

When encountering complicated degradation type like diffraction from lens occlusion in Chapter 4, we either collect real paired data or synthesize the realistic data by analyzing the image formation optical pipeline to resolve the training and benchmark data insufficiency problem of such challenging restoration tasks. For a specific Under-Display Camera imaging task, our physics-based data synthesis model can convert a occlusion display panel and related optical measurements into a set of parameters of image degradation including point spread function (blur kernel) and image intensity scaling factors etc. The model is potentially practical for the analysis of similar tasks, like designing and optimizing a better display panel for UDC [172], or resolving lens flare problems [173].

Finally, we revisited deep model design in Chapter 5 to specifically address the resolution adaptability of image registration and harmonization modules for reference-based image inpainting task [174]. We found that a deep edge-preserving bilateral filtering could make the model predict spatial and color transformation parameters using a fixed-size input in the low-resolution branch

during training and testing, while using a learned high-resolution guidance map for a flexible sampling during inference. Therefore, we can maintain the domain of the deep models while extending it to multiple input sizes. The entire inpainting pipeline thus become more practical while being tested on real user photo pairs of arbitrary exposure difference, spatial misalignment and resolution difference.

The exploration in this dissertation is focusing on utilizing the information and knowledge which is always ignored and under-explored in previous research works to boost the performance of deep image restoration or enhancement models. The methodology introduced in difference sections is proven to be effective for a variety of image restoration tasks by extensive experiments, and can be potentially generalized to other similar tasks.

6.3 Limitations

In this dissertation, we mostly utilized heuristic methods to resolve training-testing domain misalignment. Those methods mainly have two disadvantages. First, the design of the domain adaptation methods can not be easily extended and generalized to universal image restoration tasks. For example, the Pixel-shuffle Downsampling method introduced in Chapter 3 can only be applied to adapting testing domain from AWGN to spatially-correlated RGB noises. Similarly, the realistic data synthesis methods introduced in Chapter 4 are specifically designed for Under-Display Camera imaging or diffraction-related restoration tasks. The success of those methods highly depends on the understanding of the imaging pipeline of different tasks, due to the diversity and distinction among image restoration tasks, it is infeasible to apply the same rules to all of them. Second, those methods can still result in some remaining domain misalignment while failing to consider some factors in the practical pipeline. For example, in Chapter 3, the assumption of the real RGB images is based on the fact that there is a demosaicing process in the camera ISP, and the demosaicing is interpolating between neighbour pixels. Some more advanced pipeline designs like tone mapping or other non-linear transformation inside the ISP may also greatly influence the imaging results. We may never be able to consider all the influential factors due to their diversity. In Chapter 4, while synthesizing realistic noises for training, we did

not consider the factors like camera vignette, lens distortion, the non-zero distance between the display and the camera lens, and noises sourcing from the dead pixels etc. Those factors result in the gap between the model trained with real data and the one trained our synthetic data both quantitatively and qualitatively.

Besides, we did not discuss the efficiency and complexity in a deeper level. The models are mostly UNet-based structures in this dissertation. Limited network structures do not consider the deployment efficiency on different devices and inference speed in a live stream with high frame rates. The balancing between the inference time and network complexity is partially discussed in Chapter 4, but not sufficient enough for more general cases.

6.4 Future Work

Given the above mentioned limitations of this dissertation, towards a more practical deep image restoration models, efforts should be put more on the following aspects.

First, unsupervised image restoration models or degradation data synthesis using generative models can be further explored. With advanced generative model like Generative Adversarial Networks (GANs) [175], Variational AutoEncoder (VAE) [176], Normalizing Flow [177], or diffusion-based methods [178], we can study how to learn the data domain distribution from unpaired data. Besides, few-shot learning can be studied given the consumption of data collection for training. However, the performance of those model may not be easily comparable with fully-supervised models, so improving its performance will be worth exploring. Generative models are also used to improve the perceptual quality, which can be continuously explored in the future.

Second, given the task-specific knowledge like optical system formulation or some real measurements of imaging hardware, we should study the methodology to include them into the deep network design. Domain knowledge inputs will be sure to improve the restoration performance if being properly learned by the network. Some good ideas include simulating the camera ISP [40, 179] within the network.

Third, while designing deep models, utilizing and embedding more traditional signal processing methods into the networks will greatly help the

model learn the utilize image information. Some good examples include the novel Fourier Convolution (FC) [180] and our super-resolution work named cross-scale self-attention [181] etc.

Last but not the least, improving the model inference speed and reducing the capacity while preserving the restoration performance is important to explore. Our work, hasing-based self-attention [182], is another good example for this direction.

CHAPTER 7

CONCLUSION

In this dissertation, we aim at addressing training data insufficiency and training-testing domain shift issues of deep image restoration models. With the proposed learning, testing and data synthesis strategies, deep models trained on realistic synthetic data can still be well-deployed for testing on real collected data from a different testing domain. To demonstrate each strategy, we present four specific image restoration tasks, including real RAW image denoising, real RGB denoising, combined diffraction restoration of Under-Display Camera, and hole-filling in reference-based image inpainting.

We first presented the Bayer pattern manipulation methods for real RAW image denoising task. We demonstrated that unifying the Bayer (BayerUnif) patterns of the training paired images benefits the generalization ability of the image restoration network while testing on unseen noisy inputs. The proposed Bayer-preserving data augmentation (BayerAug) for RAW images also improves the robustness of the network. The two strategies can be regarded as plug-and-play adaptation modules for networks trained with RAW sensor images.

By considering the Bayer pattern and demosaicing algorithm in the in-camera pipeline, we then presented a Pixel-Shuffle Downsampling (PSD) adaptation strategy to apply AWGN-trained denoiser to real RGB image denoising. Due to the complicated signal transformation within the in-camera pipeline, the noise distribution of RGB images cannot be simply represented in an analytical form. However, from a novel view point, we showed that the proposed PSD can break down the spatial correlation of RGB noises into approximated spatial-variant Gaussian noises, which can be better processed by deep denoising models trained with AWGN. Experiments demonstrated that PSD adaptation can boost the performance of existing AWGN-trained models on real RGB noisy image benchmark.

We then discussed a physics-based image formation pipeline to synthesize

corrupted images with combined degradation factors. Specifically, we studied the diffraction effect caused by lens occlusion in Under-Display Camera (UDC). We showed that the model trained with synthetic image pairs achieved visually similar performance to the model trained with real pairs. Extensive experiments demonstrated the superior performance of our restoration model on UDC problems.

Finally, we extended our discussion to reference-based image inpainting. To complete the missing regions in the target image, we registered the source image using multiple homography, and trained deep models to further refine the color and spatial difference. Our scale-robust and content-preserving Color-Spatial Transformer (CST) works well on adjusting real image difference, though the model is trained on synthetic data and the testing image scale can be diverse. Along with the image blending modules, the proposed pipeline demonstrated state-of-the-art performance on the challenging multi-image inpainting task.

This dissertation showed that realistic training data synthesis, data domain adaptation and scale-robust model design efficiently improve the performance of deep restoration model while being tested on real inputs, if we could understand and formulate the image formation process. In the future, more efforts can be put on learning-based domain adaptation and data augmentation methods using adversarial training or flow-based generative models etc..

REFERENCES

[1] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. 1, 13, 14, 21, 25

[2] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, and T. Huang, "When AWGN-based denoiser meets real noises," *arXiv preprint arXiv:1904.03485*, 2019. 1, 4

[3] Y. Zhou, J. Jiao, H. Huang, J. Wang, and T. Huang, "Adaptation strategies for applying AWGN-based denoiser to realistic noise," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 10 085–10 086. 1, 13

[4] J. Liu, C.-H. Wu, Y. Wang, Q. Xu, Y. Zhou, H. Huang, C. Wang, S. Cai, Y. Ding, H. Fan et al., "Learning raw image denoising with bayer pattern unification and bayer preserving augmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0. 1, 40

[5] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018. 1, 13, 17, 22, 23, 25

[6] A. Abdelhamed, R. Timofte, and M. S. Brown, "Ntire 2019 challenge on real image denoising: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0. 1, 13, 15

[7] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8183–8192. 1

[8] S. Nah, R. Timofte, S. Baik, S. Hong, G. Moon, S. Son, and K. Mu Lee, "Ntire 2019 challenge on video deblurring: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0. 1, 15

[9] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, 2019. 1

[10] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 695–704. 1

[11] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, p. 72, 2008. 1

[12] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 154–169. 1

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144. 1, 15, 40

[14] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0. 1

[15] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3291–3300. 1, 4, 5, 6, 9, 10, 15

[16] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 60–65. 4

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering. image processing, ieee transactions on 16 (8), pp. 2080-2095," 2007. 4

[18] Y. Tai, J. Yang, X. Liu, and C. Xu, "MEMNET: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4539–4547. 4

[19] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017. 4, 25

[20] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in neural information processing systems*, 2009, pp. 769–776. 4

[21] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341–349. 4

[22] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810. 4

[23] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454. 4

[24] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4, 9

[25] J. Anaya and A. Barbu, "RENOIR–a dataset for real low-light image noise reduction," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 144–154, 2018. 4, 15

[26] K. Hirakawa and T. W. Parks, "Joint demosaicing and denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2146–2157, 2006. 4

[27] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 191, 2016. 4

[28] B. E. Bayer, "Color imaging array," July 20 1976, uS Patent 3,971,065. 4

[29] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 192, 2016. 4

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 6, 10, 56

[31] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017. 10

[32] Y. Zhou, D. Liu, and T. Huang, "Survey of face detection on low-quality images," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 769–773. 12

[33] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5232–5239. 12

[34] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3d transform-domain collaborative filtering," in *Image Processing: Algorithms and Systems VI*, vol. 6812. International Society for Optics and Photonics, 2008, p. 681207. 13, 25

[35] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *ICCV*, 2017. 13, 25

[36] N. Yair and T. Michaeli, "Multi-scale weighted nuclear norm image restoration," in *CVPR*, 2018. 13

[37] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006. 13, 25

[38] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *arXiv preprint arXiv:1711.10925*, 2017. 13

[39] K. Z. W. Z. L. Z. Shi Guo, Zifei Yan, "Toward convolutional blind denoising of real photographs," in *arXiv preprint arXiv:1807.04686*, 2018. 13, 23, 25

[40] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 036–11 045. 13, 16, 24, 35, 79

[41] M. Lebrun, M. Colom, and J.-M. Morel, "Multiscale image blind denoising," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3149–3161, 2015. 13

[42] D. Liu, B. Wen, X. Liu, Z. Wang, and T. S. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," *arXiv preprint arXiv:1706.04284*, 2017. 13

[43] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *CVPR*, 2018. 13

[44] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, 2008. 13, 16, 21

[45] Stanford, "Demosaicking and denoising," https://web.stanford.edu/group/vista/cgi-bin/wiki/index.php/Demosaicking_and_Denoising, 2015. 13

[46] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008. 14

[47] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *ICCV*, 2015. 14

[48] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *CVPR*, 2017. 15

[49] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *CVPR*, 2017. 15

[50] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *CVPR*, 2018. 15

[51] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673. 15

[52] Y. Fan, J. Yu, D. Liu, and T. S. Huang, "Scale-wise convolution for image restoration," *arXiv preprint arXiv:1912.09028*, 2019. 15

[53] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, 2018. 15

[54] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *NeurIPS*, 2018. 15

[55] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *ICLR*, 2019. 15

[56] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074. 15

[57] X. Zhang, Q. Chen, R. Ng, and V. Koltun, "Zoom to learn, learn to zoom," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3762–3770. 15

[58] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1586–1595. 15, 22

[59] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1692–1700. 15, 19

[60] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, "Camera lens super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1652–1660. 15

[61] R. Wan, B. Shi, L.-Y. Duan, A.-H. Tan, and A. C. Kot, "Benchmarking single-image reflection removal algorithms," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3922–3930. 15

[62] A. Punnappurath and M. S. Brown, "Reflection removal using a dual-pixel sensor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1556–1565. 15

[63] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1712–1722. 16

[64] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: Noise modeling with conditional normalizing flows," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3165–3173. 16

[65] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6840–6848. 16

[66] C. O. Ancuti, C. Ancuti, M. Sbert, and R. Timofte, "Dense haze: A benchmark for image dehazing with dense-haze and haze-free images," *arXiv preprint arXiv:1904.02904*, 2019. 16

[67] J.-S. Lee, "Refined filtering of image noise using local statistics," NAVAL RESEARCH LAB WASHINGTON DC, Tech. Rep., 1980. 17

[68] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," *arXiv preprint arXiv:1807.04364*, 2018. 17, 25

[69] J. Xu, D. Ren, L. Zhang, and D. Zhang, "Patch group based bayesian learning for blind image denoising," in *ACCV*, 2016. 17, 22, 23

[70] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016. 18

[71] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE transactions on image processing*, vol. 22, no. 12, pp. 5226–5237, 2013. 19, 20, 23

[72] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, p. 205, 2009. 20, 21, 22

[73] Online, "[online] available:," https://ni.neatvideo.com/home, 2015. 22

[74] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064. International Society for Optics and Photonics, 2006, p. 606414. 23

[75] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2862–2869. 23, 24, 25

[76] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008. 23

[77] X. Liu, M. Tanaka, and M. Okutomi, "Practical signal-dependent noise parameter estimation from a single noisy image," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4361–4371, 2014. 23

[78] Online, "[online] available:," https://ni.neatvideo.com/, 2015. 24

[79] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: a blind image denoising algorithm," *Image Processing On Line*, vol. 5, pp. 1–54, 2015. 24

[80] K. Yu, X. Wang, C. Dong, X. Tang, and C. C. Loy, "Path-restore: Learning network path selection for image restoration," *arXiv preprint arXiv:1904.10343*, 2019. 24

[81] S. Anwar and N. Barnes, "Real image denoising with feature attention," *arXiv preprint arXiv:1904.07396*, 2019. 24

[82] R. Zhao, K.-M. Lam, and D. P. Lun, "Enhancement of a cnn-based denoiser based on spatial and spectral analysis," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1124–1128. 24

[83] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013. 25

[84] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2392–2399. 25

[85] D.-M. Chen, B. Xiong, and Z.-Y. Guo, "Full-screen smartphone," Sep. 3 2019, uS Patent App. 29/650,323. 29

[86] V. D. J. Evans, X. Jiang, A. E. Rubin, M. Hershenson, and X. Miao, "Optical sensors disposed beneath the display of an electronic device," Oct. 17 2019, uS Patent App. 16/450,727. 29

[87] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company Publishers, 2005. 34

[88] S. W. Hasinoff, "Photon, poisson noise," *Computer Vision: A Reference Guide*, pp. 608–610, 2014. 35

[89] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 299–314, 2007. 35

[90] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135. 36

[91] F. Orieux, J.-F. Giovannelli, and T. Rodet, "Bayesian estimation of regularization and point spread function parameters for wiener–hunt deconvolution," *JOSA A*, vol. 27, no. 7, pp. 1593–1607, 2010. 39, 41

[92] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 40

[93] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595. 41

[94] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. 43

[95] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009. 46, 49

[96] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4471–4480. 46, 49, 58

[97] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, "Foreground-aware image inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 5840–5848. 46

[98] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514. 46, 59

[99] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, "Onion-peel networks for deep video completion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4403–4412. 46, 49, 59, 67, 68

[100] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! internet-based inpainting," in *Proceedings of the 20th British Machine Vision Conference, London*, 2009. 47

[101] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5515–5524. 48

[102] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *arXiv preprint arXiv:1805.09817*, 2018. 48, 53, 57

[103] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019. 48

[104] R. Tucker and N. Snavely, "Single-view view synthesis with multiplane images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 551–560. 48

[105] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424. 49

[106] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE transactions on image processing*, vol. 10, no. 8, pp. 1200–1211, 2001. 49

[107] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 463–476, 2007. 49

[108] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544. 49

[109] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017. 49

[110] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100. 49

[111] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "EdgeConnect: Generative image inpainting with adversarial edge learning," 2019. 49

[112] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, "SPG-Net: Segmentation prediction and guidance network for image inpainting," *arXiv preprint arXiv:1805.03356*, 2018. 49

[113] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: Image inpainting via structure-aware appearance flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. 49

[114] L. Liao, J. Xiao, Z. Wang, C.-w. Lin, and S. Satoh, "Guidance and evaluation: Semantic-aware image inpainting for mixed scenes," *arXiv preprint arXiv:2003.06877*, 2020. 49

[115] C. Zheng, T.-J. Cham, and J. Cai, "Pluralistic image completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1438–1447. 49

[116] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6721–6729. 49

[117] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," *arXiv preprint arXiv:2005.11742*, 2020. 49, 55, 56, 58, 59, 67, 68

[118] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7508–7517. 49

[119] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *European Conference on Computer Vision*. Springer, 2012, pp. 682–695. 49

[120] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *Siam journal on imaging sciences*, vol. 7, no. 4, pp. 1993–2019, 2014. 49

[121] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3723–3732. 49, 58, 59, 67, 68

[122] S. Lee, S. W. Oh, D. Won, and S. J. Kim, "Copy-and-paste networks for deep video inpainting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4413–4421. 49

[123] Y. Zhao, B. Price, S. Cohen, and D. Gurari, "Guided image inpainting: Replacing an image region by pulling content from another image," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1514–1523. 50

[124] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman, "A computational approach for obstruction-free photography," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–11, 2015. 50

[125] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to color transfer," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1434–1439. 50

[126] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001. 50

[127] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-scale image harmonization," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–10, 2010. 50

[128] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*, 2003, pp. 313–318. 50, 70

[129] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum, "Drag-and-drop pasting," *ACM Transactions on graphics (TOG)*, vol. 25, no. 3, pp. 631–637, 2006. 50

[130] M. W. Tao, M. K. Johnson, and S. Paris, "Error-tolerant image compositing," in *European Conference on Computer Vision*. Springer, 2010, pp. 31–44. 50

[131] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 294–302. 50

[132] J.-Y. Zhu, P. Krahenbuhl, E. Shechtman, and A. A. Efros, "Learning a discriminative model for the perception of realism in composite images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3943–3951. 50

[133] C. Xiaodong and P. Chi-Man, "Improving the harmony of the composite image by spatial-separated attention module," *arXiv preprint arXiv:1907.06406*, 2019. 50

[134] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3789–3797. 50

[135] W. Cong, J. Zhang, L. Niu, L. Liu, Z. Ling, W. Li, and L. Zhang, "DoveNet: Deep image harmonization via domain verification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8394–8403. 50

[136] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–12, 2017. 50, 53, 54, 68

[137] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6849–6857. 50

[138] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "Meshflow: Minimum latency online video stabilization," in *European Conference on Computer Vision*. Springer, 2016, pp. 800–815. 50

[139] K.-Y. Lee and J.-Y. Sim, "Warping residual based image stitching for large parallax," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8198–8206. 50

[140] C. Herrmann, C. Wang, R. S. Bowen, E. Keyder, M. Krainin, C. Liu, and R. Zabih, "Robust image stitching with multiple registrations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 53–67. 50

[141] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157. 50

[142] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Learning two-view correspondences and geometry using order-aware network," *International Conference on Computer Vision (ICCV)*, 2019. 50, 52

[143] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947. 50

[144] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *arXiv preprint arXiv:1606.03798*, 2016. 50, 58

[145] J. Zhang, C. Wang, S. Liu, L. Jia, N. Ye, J. Wang, J. Zhou, and J. Sun, "Content-aware unsupervised deep homography estimation," *arXiv preprint arXiv:1909.05983*, 2019. 50

[146] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2346–2353, 2018. 50

[147] S. Li, L. Yuan, J. Sun, and L. Quan, "Dual-feature warping-based motion model estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4283–4291. 50

[148] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–9, 2009. 50

[149] B. K. Horn and B. G. Schunck, "" determining optical flow": A retrospective," 1993. 50

[150] J. Wulff and M. J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 120–130. 50

[151] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2432–2439. 50

[152] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943. 50

[153] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1385–1392. 50

[154] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470. 50, 59

[155] N. Ye, C. Wang, S. Liu, L. Jia, J. Wang, and Y. Cui, "DeepMeshFlow: Content adaptive mesh deformation for robust image registration," *arXiv preprint arXiv:1912.05131*, 2019. 50

[156] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in *CVPR 2011*. IEEE, 2011, pp. 49–56. 52

[157] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 52, 68

[158] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," 2019. 53, 58

[159] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967. 53

[160] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 53

[161] M. Jaderberg, K. Simonyan, A. Zisserman et al., "Spatial transformer networks," in *NeurIPS*, 2015. 53

[162] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 57

[163] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input / output image pairs," in *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 58

[164] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014. 58

[165] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037. 58

[166] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2339–2346. 58, 59, 67, 68

[167] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. 66

[168] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. 66

[169] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International journal of computer vision*, vol. 92, no. 1, pp. 1–31, 2011. 69

[170] Online, "[online] available:," https://www.congress.gov/bill/116th-congress/house-bill/3230/text., 2015. 73

[171] M. Garon, K. Sunkavalli, S. Hadap, N. Carr, and J.-F. Lalonde, "Fast spatially-varying indoor lighting estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6908–6917. 74

[172] A. Yang and A. Sankaranarayanan, "Designing display pixel layouts for under-panel cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 77

[173] R. Feng, C. Li, H. Chen, S. Li, C. C. Loy, and J. Gu, "Removing diffraction image artifacts in under-display camera via dynamic skip connection network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 662–671. 77

[174] Y. Zhou, C. Barnes, E. Shechtman, and S. Amirghodsi, "Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2266–2276. 77

[175] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016. 79

[176] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 79

[177] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *arXiv preprint arXiv:1807.03039*, 2018. 79

[178] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arXiv:2006.11239*, 2020. 79

[179] A. Ignatov, L. Van Gool, and R. Timofte, "Replacing mobile camera isp with a single deep learning model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 536–537. 79

[180] L. Chi, B. Jiang, and Y. Mu, "Fast fourier convolution," *Advances in Neural Information Processing Systems*, vol. 33, 2020. 80

[181] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 80

[182] Y. Mei, Y. Fan, and Y. Zhou, "Image super-resolution with non-local sparse attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3517–3526. 80