Blind Non-Uniform Motion Deblurring using Atrous Spatial Pyramid Deformable Convolution and Deblurring-Reblurring Consistency

Dong Huo, Abbas Masoumzadeh, Yee-Hong Yang
Department of Computing Science
University of Alberta, Edmonton, Canada
{dhuo, a.masoumzadeh, herberty}@ualberta.ca

Abstract

*Many deep learning based methods are designed to remove non-uniform (spatially variant) motion blur caused by object motion and camera shake without knowing the blur kernel. Some methods directly output the latent sharp image in one stage, while others utilize a multi-stage strategy (*e.g. *multi-scale, multi-patch, or multi-temporal) to gradually restore the sharp image. However, these methods have the following two main issues: 1) The computational cost of multi-stage is high; 2) The same convolution kernel is applied in different regions, which is not an ideal choice for non-uniform blur. Hence, non-uniform motion deblurring is still a challenging and open problem. In this paper, we propose a new architecture which consists of multiple Atrous Spatial Pyramid Deformable Convolution (ASPDC) modules to deblur an image end-to-end with more flexibility. Multiple ASPDC modules implicitly learn the pixel-specific motion with different dilation rates in the same layer to handle movements of different magnitude. To improve the training, we also propose a reblurring network to map the deblurred output back to the blurred input, which constrains the solution space. Our experimental results show that the proposed method outperforms state-of-the-art methods on the benchmark datasets. The code is available at https://github.com/Dong-Huo/ASPDC.*

1. Introduction

When we are taking photos using a camera, especially the one on a mobile device, non-uniform motion blur is one of the most common types of undesirable artifacts caused by object motion and camera shake [13]. Removing such blur to recover the original sharp image plays a critical role in many high-level vision tasks, e.g. computational photography [34], image classification [43], object detection [62], and face recognition [1], because motion blur severely degrades the image quality. Both Zhang *et al*. [61] and Nah *et al*. [37] claim that motion blur can be regarded as the temporal integration of multiple sharp snapshots during the exposure time, and can be formulated as:

$$I_b = g\left(\frac{1}{T}\int_{t=0}^{T} I_{S(t)}dt\right), \tag{1}$$

in which I_b is the blurred image of the dynamic scene, T is the period of the exposure time, $I_{S(t)}$ is the sharp snapshot at timestamp t and $g()$ represents the Camera Response Function (CRF).

In addition to the above model, some works model the non-uniform motion deblurring as a linear transformation. Indeed, Bahat *et al*. [4] formulate the problem using the following equation:

$$I_b = I_s \circledast k + n, \tag{2}$$

where I_b and I_s represent the blurred and sharp image, respectively, n is the additive noise, k is the spatially variant blur kernel matrix which is different from some of the deblurring methods using a uniform blur kernel [10, 25, 44]. Each column of k represents a blur kernel for the corresponding pixel in I_s. The blur kernel matrix k is applied to the sharp image by the matrix multiplication operator $\circledast$. The objective is to find the I_s given I_b, assuming that k and n are unknown.

To handle the blind deblurring problem, many conventional methods attempt to first estimate the blur kernel, then use it to recover the sharp image with some hand-crafted priors of I_s and k [17,31,38,39,46,55,64]. However, Ren *et al*. [44] claim that hand-crafted priors are insufficient to recover the ideal sharp image, and an improper prior can even lead to an incorrect kernel. Recently, deep learning based methods [2,28,29,37,40,42,48,50,58–60] significantly improve the performance of non-uniform motion deblurring. The priors of I_s and k are implicitly learned by the network, which outputs the deblurred result directly, and bypasses the need to estimate the blur kernel. However, existing deep learning methods suffer from two main issues: 1)

Many state-of-the-art (SOTA) methods utilize a multi-stage strategy, such as multi-scale [37, 50], multi-patch [48, 59], or multi-temporal [40], all of which increase the computational cost. 2) The same convolution filters [29] or filters of the same receptive fields [42] are applied to different regions of an image. To overcome such issue, extremely deep and wide networks have been exploited to improve the generalization on different levels of blur.

To address the above issues, we propose a new *Atrous Spatial Pyramid Deformable Convolution* (ASPDC) module for region-specific convolution and for integrating features from different sizes of receptive fields, which is more suitable for non-uniform deblurring. We also propose a new reblurring network to reblur the output, which is helpful in constraining the solution space [16] of deblurring with a new deblurring-reblurring consistency loss. Note that our reblurring network is used during training only, which needs both of the blurred image and the corresponding sharp (deblurred) image. More details are shown in Section 3. Extensive experimental results demonstrate the effectiveness of the proposed method compared to other SOTA methods on the benchmark datasets.

The contributions of this paper are summarized below:

- We propose a novel non-uniform motion deblurring method with Atrous Spatial Pyramid Deformable Convolution (ASPDC) modules that realize region-specific convolution and different sizes of receptive fields, simultaneously. Our ablation study shows that both of these features significantly improve the performance using the same baseline.

- Different dilation rates in the ASDPC module extract information of different magnitudes of motion and separate the image into regions with the help of attention maps, which reduces the workload of each branch by focusing on regions with specific magnitudes of motion, instead of always considering the entire image.

- Our proposed deblurring network is end-to-end and contains only a single stage, which is more efficient and has a lower computational cost compared with other multi-stage methods.

- We propose a new end-to-end reblurring network that maps images of the deblurred domain back to the blurred domain. Then the deblurred output can be further refined using a new deblurring-reblurring consistency loss without estimating motion information.

2. Related Work

2.1. Blind Non-uniform Image Deblurring

Since the blur kernel is spatially variant and unknown, blind non-uniform image deblurring is an ill-posed problem. Conventional methods usually constrain the condition of the blur and the image. Jia [23] assumes that the image contains only one moving object near the image center, and the transparency region of the blurred image helps to calculate the upper bound of the blur kernel. Gupta *et al*. [17] calculate the integrated camera pose of selected patches with motion density function to estimate the local blur kernel and deblur the fronto-parallel scene. Hyun *et al*. [20] segment the image and handle the objects and background separately. Hyun *et al*. [21] utilize the edge-aware regularization to make the motion flow of neighboring pixels similar and to preserve edges. Anwar *et al*. [3] explore a class-specific prior to deblur a specific kind of object. Bahat *et al*. [4] find that when the blur kernel is an ideal low pass filter, applying the blur kernel to a blurred image will not change it. However, if the kernel is non-ideal, adding pink noise is helpful to estimate the blur kernel. The sharp image is recovered using the patch recurrence property [35] within and across scales with the estimated local kernel. Bai *et al*. [5] prove that the extreme downsampled case of a blurred image is an approximation of the latent sharp image, which can be used as the prior image to reconstruct the latent sharp image from coarse to fine.

In order to improve the generalization of deblurring, deep learning based methods, such as a Convolutional Neural Network (CNN), are applied to solve this problem. Some methods focus on estimating motion information to use it for recovering the latent sharp image. Sun *et al*. [49] combine the conventional methods and CNN. They assume that the blur kernel is locally invariant and the motion is linear, and use a CNN to estimate the motion field of overlapping patches. Then a Markov Random Field (MRF) model [30] is used to fuse the recovered patches. Gong *et al*. [13] estimate the motion vector of each pixel and recover the whole sharp image directly instead of individual patches, which enables taking advantage of the context information of the image. Thekke *et al*. [51] attempt to learn the weights of camera poses from a set of poses. Then the latent sharp image is reblurred with the estimated weights to calculate the reblurring error. Such a strategy can keep the cycle consistency, but it focuses on inplane motion only and the consistency is limited by a predefined camera pose set. In contrast, ours can recover more general blur without putting any specific limitations.

Some methods directly restore the sharp image in one stage. DeblurGAN [28] has an end-to-end architecture with multiple residual blocks [18] and instance normalization layers [52]. The authors use WGAN-GP [15] to stabilize the adversarial training. DeblurGAN-v2 [29] extends the previous work [28] using a much deeper architecture with an encoder-decoder network. Since the non-uniform deblurring can be formulated using the Infinite Impulse Response (IIR) model [32], Zhang *et al*. [60] exploit the spa-

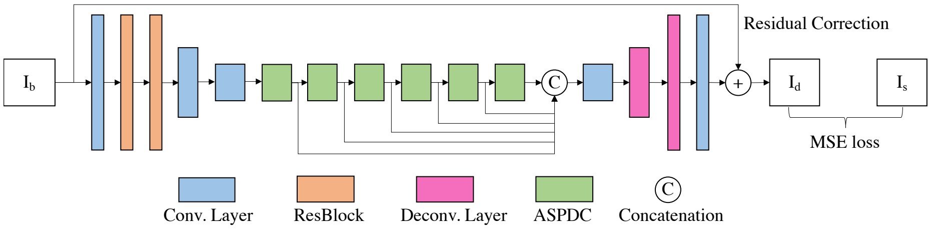
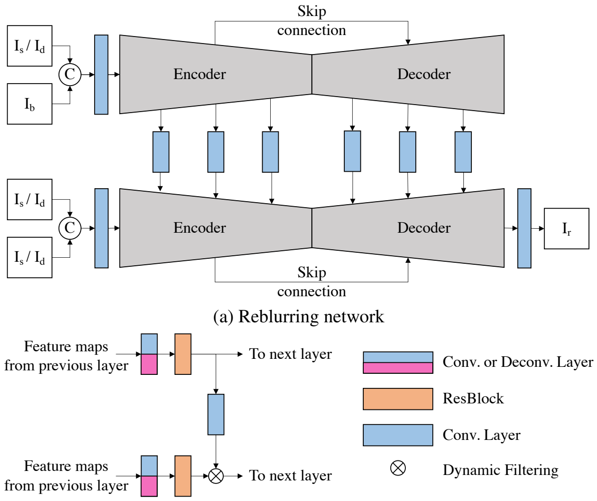


Figure 1. Overview of the deblurring network architecture.



(a) Reblurring network

(b) Feed blurred information from the upper to lower branch

Figure 2. Overview of the reblurring network architecture.

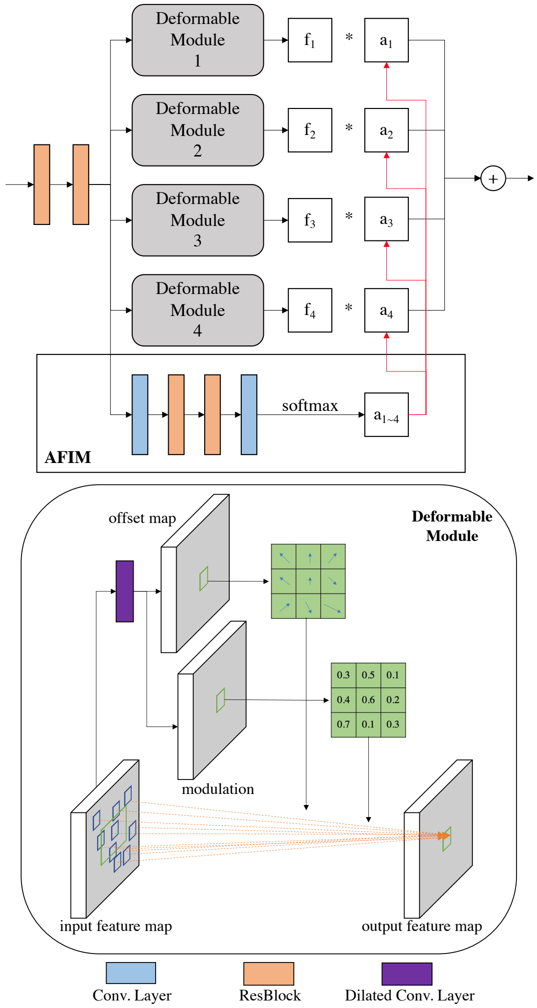


Figure 3. Schematic of the ASPDC module.

tially variant RNN to take advantage of the long-range dependency in four directions. Aljadaany *et al.* [2] implement the Douglas-Rachford splitting method [11] in a CNN which combines the advantages of both optimization and deep learning. Zhang *et al.* [61] attempt to train the model on realistic blurred images generated from a GAN. Yuan *et al.* [58] utilize the Dense Inverse Search algorithm [27] to estimate optical flow, then use it to guide the offset map in deformable convolution v2 [63], which combines motion estimation with end-to-end learning.

Some methods restore the sharp image in multiple stages. Nah *et al.* [37] propose a multi-scale architecture to deblur the dynamic scene from coarse to fine. Tao *et al.* [50] share the parameters of different scales. Zhang *et al.* [59] and Suin *et al.* [48] crop the blurred image into multiple patches of different sizes instead of downsampling to multiple scales. Thus, they preserve high frequency information. Park *et al.* [40] use a multi-temporal strategy to gradually sharpen the output.

2.2. Spatially Variant Convolution

Dai *et al.* [6] propose to learn an offset map to change the shape of the convolution filters, the result of which is more robust to geometric transformations. Huo *et al.* [19] simplify the offset map of Dai *et al.* [6] by learning an offset for each pixel instead of each region but with more maps in a single layer. Zhu *et al.* [63] extend the work of Dai *et al.* [6] by learning an extra modulation to weight the convolution filter, which can refine the effective receptive fields. Jia *et al.* [24] use an extra branch to learn the filter dynamically, but the computational cost is high for large feature maps.

Ding *et al*. [9] preserve the semantic correlation by multiple paired convolutions in local patches. The pixel-adaptive convolution of Su *et al*. [47] can be regarded as a simplified version of the paired convolution [9] by reducing the kernel size to one. Dai *et al*. [7] exploit the position information of each pixel. The local patches extracted by the data kernel are concatenated with their positions in the image, and then are passed to a convolution layer.

3. Proposed Method

3.1. Overview

Overviews of the proposed deblurring and reblurring network architecture are illustrated in Figure 1 and Figure 2, respectively. To make the training more stable, we initially train the two networks separately. The deblurring network attempts to recover the sharp image I_s from the blurred input I_b, then the deblurred output I_d is reblurred by the reblurring network. The reblurring network takes both I_s and I_b as input and outputs the reblurred image I_r. When the training of two networks are converged, we replace the input I_s of the reblurring network with the deblurred output I_d, and fine-tune the I_d with the deblurring-reblurring consistency loss. We do not use any kind of normalization (e.g. batch normalization [22] or instance normalization [52]).

Our deblurring-reblurring consistency is inspired by Guo *et al*. [16]. Deblurring and reblurring can be regarded as a pair of dual tasks. The former is the primary task and the latter is the corresponding dual task, which is similar to upsampling and downsampling in super-resolution [56]. Guo *et al*. [16] proves that the generalization bound of the dual regression (in our case, consistency) is lower than that of the primary regression (only deblurring). Therefore, it leads to more accurate deblurring results. However, simply mapping the deblurred output to the original blurred input is highly ill-posed. Lu *et al*. [33] show that the sharp image only contains the sharp content without any blur information, so it needs the blur information from the corresponding blurred image as the extra input for reblurring. Thus, we also utilize blur information.

3.2. Deblurring Network

We use two residual blocks (ResBlocks) [18] and strided convolution layers to extract high dimensional features at the beginning, and two deconvolution layers to recover the spatial dimension in the end. The last convolution layer reduces the channel size of the feature map to 3 (RGB). We find that learning the residual correction instead of directly learning the latent sharp image can make the training more stable and faster. To explain our intuition, note that the blurred image contains all of the signals from the sensor during the exposure time, as shown in Eqn 1. One of the sharp images $I_{S(t)}$, say at time t^*, is the corresponding target while many features extracted from I_b could be from times other than t^*. Hence, learning features of the other times will be easier than directly learning a specific one.

In the middle of the deblurring network, we stack six ASPDC modules (as shown in Figure 3) in which we extend the work of the deformable convolution network v2 (DCNv2) [63]. The original DCNv2 applies different convolution kernels to different regions by learning an offset map Δp and a modulation Δm. But a fixed size is used for the receptive field of each region used to generate Δp and Δm. Since Δp represents the shift of each pixel, it can be regarded as the local optical flow [58] corresponding to the motion of the object and the camera. For a non-uniform blurred image, some of the regions might have only small variations while other regions might have large movements and overlaps. In this case, the original DCNv2 uses a single convolution layer to generate Δp and Δm and treats these regions similarly, which is not an optimal choice for this problem.

To make the deformable convolution more flexible, in our ASPDC module, we build four branches with different dilation rates [57] to generate four offset maps Δp and modulations Δm with different receptive fields, and four deformable convolution outputs. As shown in Figure 3, the dilation rates of dilated convolution layers in deformable modules 2∼4 are 1, 2, and 4, respectively. The deformable module 1 also uses the dilation rate 1 but it ignores the offsets (by setting Δp as zero). Such a special module is used to recover static regions.

The outputs of the four branches in an ASPDC module are fused by an attention feature integration module (AFIM) [8]. We can write it as:

$$f_o = \sum_{i=1}^{4} a_i * f_i, \tag{3}$$

$$\sum_{i=1}^{4} a_{ij} = 1, 1 \leq j \leq h \times w, \tag{4}$$

in which f_i is the output of the i^{th} branch, a_i is a single-channel attention map generated from the AFIM, j is the index of the pixel, h and w are, respectively, the height and width of the attention map, $*$ represents the element-wise multiplication and f_o is the output of the ASPDC module. To make sure that the channel-wise sum of attention maps is 1, we utilize the softmax activation along the channel. In this case, each region that integrates information from different receptive fields significantly boosts the performance. The output feature maps of six ASPDC modules are further concatenated to stabilize training.

We use the Mean Squared Error (MSE) loss as our final deblurring loss:

$$L_{deblurring} = ||I_s - I_d||_F^2, \tag{5}$$

Method	Xu [54]	Sun [49]	Nah [37]	Kupyn [28]	Tao [50]	Zhang [60]	Kupyn [29]	Aljadaan [2]
PSNR	20.30	25.31	28.49	28.70	30.26	29.19	29.55	30.35
SSIM	0.741	0.851	0.917	0.927	0.934	0.931	0.934	0.961
Method	Zhang [59]	Suin [48]	Park [40]	Yuan [58]	Purohit [42]	Zhang [61]	Ours	Ours+
PSNR	31.20	32.02	31.15	29.81	31.76	31.10	31.97	32.09
SSIM	0.945	0.953	0.945	0.937	0.953	0.942	0.957	0.959

Table 1. Quantitative comparison on the GoPro dataset [37]. Ours/Ours+ represents our deblurring network without/with fine-tuning on the reblurring network. The best results are in red and the second best in blue.

Method	Kupyn [28]	Tao [50]	Zhang [59]	Park [40]	Suin [48]	Shen [45]	Kupyn [29]	Ours	Ours+
PSNR	24.51	28.36	29.09	29.16	29.98	28.89	26.61	29.98	30.04
SSIM	0.871	0.915	0.924	0.933	0.930	0.930	0.875	0.944	0.945

Table 2. Quantitative comparison on the HIDE dataset [45]. Ours/Ours+ represents our deblurring network without/with fine-tuning on the reblurring network. The best results are in red and the second best in blue.

where I_s and I_d are the sharp target and the deblurred output, respectively.

3.3. Reblurring Network

In order to narrow down the solution space of deblurring and to refine the deblurred output I_d, we build an end-to-end reblurring network to reblur the deblurred output and calculate the deblurring-reblurring consistency loss. Simply mapping the sharp image back to the blurred image is not impossible but difficult. Because the non-uniform blurred image domain is much larger than the sharp image domain. Hence, we need blur information from the blurred image to assist the mapping. However, directly inputting sharp and blurred image together and outputting the reblurred image is difficult to train, since the training procedure is unstable and easy to collapse to an identity mapping. The network may choose to output the blurred input directly and ignore the sharp input, which is definitely undesirable.

To handle the above problem, we utilize an architecture which is able to take full use of blur information and avoid training collapse. As shown in Figure 2a, the network contains two encoder-decoder branches, and the weights of the two branches are shared for reducing the number of parameters. The architecture of the encoder-decoder is simple, which consists of multiple conv/deconv-resblock pairs (convolution layers for the encoder and deconvolution layers for the decoder) as shown in Figure 2b. The concatenation of blurred and sharp image is used as the input of the upper branch, and two duplicate sharp images are input into the lower branch for matching the channel dimension.

The upper branch learns to compare the blurred and the sharp image and passes feature maps with blur information to the lower branch after each conv/deconv-resblock pair. In the lower branch, a convolution layer reduces channels of feature maps from the upper branch into $K \times K$ to generate a dynamic local filter [24] for each pixel. For reducing computational cost of dynamic filtering, we set K as 3 and apply the same filter on all channels of feature maps from the lower branch. The dynamic local filters are regarded as the spatial-variant blur kernels which gradually reblur the feature maps of the lower branch from beginning to end. The detailed architecture of the reblurring network is shown in supplementary material.

Similar to the deblurring network, we use the Mean Squared Error (MSE) loss here:

$$L_{reblurring} = ||I_r - I_b||_F^2. \tag{6}$$

3.4. Fine-tuning

After the training of the deblurring and reblurring network converges, we replace I_s in the reblurring network with the deblurred output I_d to refine I_d by the deblurring-reblurring consistency loss. The loss function is defined as:

$$L_{consistency} = L_{deblurring} + \lambda L_{reblurring}, \tag{7}$$

where λ is the weight of the reblurring loss, and we empirically set $\lambda = 0.1$.

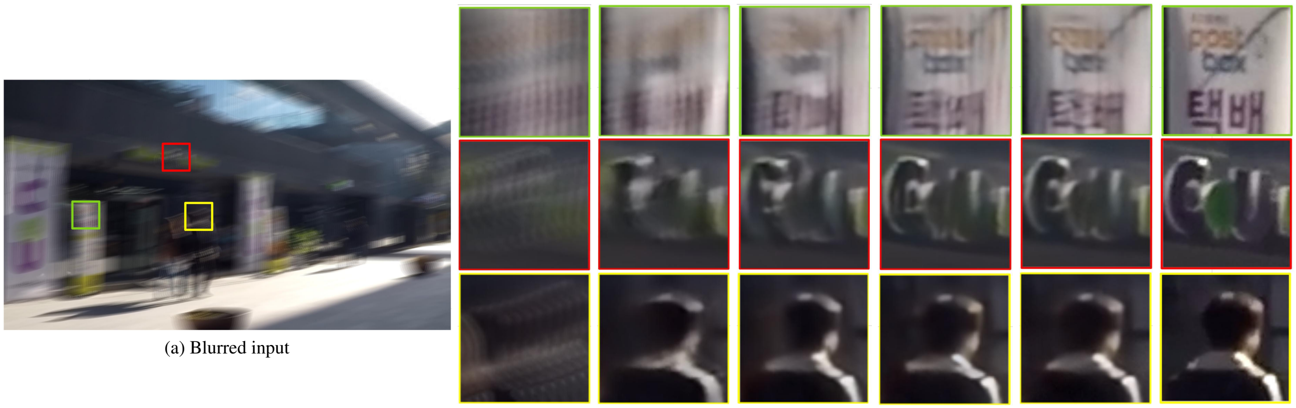
4. Experiments

4.1. Datasets

We follow the literature [28, 29, 37, 48, 59] to train our model on 2103 training images from the GoPro dataset [37]. We then use 1111 testing images from the GoPro dataset and 2025 testing images from the HIDE dataset [45] as our testing set. We also do qualitative comparison on the Real World Blurred Image (RWBI) dataset [61].

4.2. Implementation Details

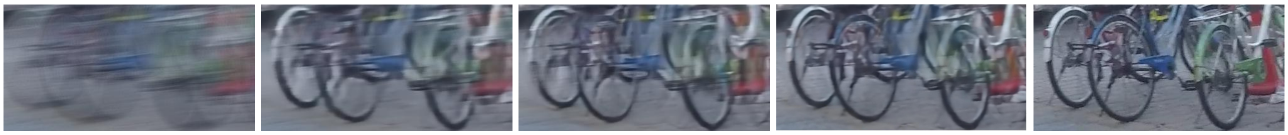
The method is implemented in PyTorch [41] and evaluated on a single NVIDIA RTX 2080 Ti GPU. During train-



(a) Blurred input

(b) Blurred (c) Park [40] (d) Zhang [59] (e) Prohit [42] (f) Ours+ (g) Sharp GT

Figure 4. Qualitative comparison on the GoPro dataset: a) Blurred input image. b-g) Magnified crops of the blurred input and deblurred outputs of compared methods, and the sharp ground truth.



(a) Blurred input (b) Park [40] (c) Zhang [59] (d) Ours+ (e) Sharp GT

Figure 5. Qualitative comparison on the HIDE dataset: a) Blurred input image. b-g) Magnified crops of the blurred input, deblurred outputs of compared methods, and the sharp ground truth.

ing, we use Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-18}$. All parameters are initialized using Xavier normalization [12]. We randomly crop the training images into 256×256 patch pairs and set the batch size as 6. The learning rate is initialized as 10^{-4} and halved every 1000 epochs. The training procedure is terminated when the learning rate reaches 10^{-6}. During fine-tuning, we set the learning rate as 10^{-5} and it is halved every 200 epochs. We stop the fine-tuning when the learning rate reaches 10^{-6}. The size of all convolution filters is 3×3. We set the initial number of channels for all convolution layers and residual blocks to 32 in the deblurring network and 16 in the reblurring network, and we double (halve) them every time we downscale (upscale) the spatial dimension.

4.3. Quantitative Comparison

We first compare our method ("ours+") with some others on the 1111 testing images from the GoPro dataset, including a conventional method (Xu *et al.* [54]), and some deep learning based methods (Sun *et al.* [49], Nah *et al.* [37], Kupyn *et al.* [28], Tao *et al.* [50], Zhang *et al.* [60], Kupyn *et al.* [29], Aljadaany *et al.* [2], Zhang *et al.* [59], Suin *et al.* [48], Park *et al.* [40], Yuan *et al.* [58], Purohit *et al.* [42] and Zhang *et al.* [61]). We also evaluate our deblurring network without fine-tuning on the reblurring network ("ours"). We use PSNR [36] and SSIM [53] as evaluation metrics. All the methods are trained on the GoPro dataset following the same strategy.

Method	Kupyn [29]	Zhang [59]	Nah [37]
Time (sec)	1.68	0.40	0.93
GPU (GB)	2.41	2.10	9.70
Method	Tao [50]	Park [40]	Ours
Time (sec)	0.78	0.05	0.28
GPU (GB)	6.09	8.49	2.25

Table 3. Average testing time and GPU usage of images of size 1280×720 on a single NVIDIA RTX 2080 Ti GPU.

The results are shown in Table 1. Our method outperforms most of the existing SOTA methods even without fine-tuning, and fine-tuning on the reblurring network can further improve the performance. In terms of PSNR, Ours+ is ranked first and is 0.07db better than the second [48]. Although our SSIM is slightly lower than the first [2], our PSNR far surpasses it which demonstrates that our method is good at both evaluation metrics. Note that we use the mean squared error loss without a sophisticated GAN [14] and still achieve good performance.

We further compare with some of the methods on the HIDE dataset in Table 2. Ours+ remains the top and Ours is ranked second. Note that unlike all other methods that follow the same strategy of training on the GoPro dataset but being tested on the HIDE dataset, the method of Shen *et*

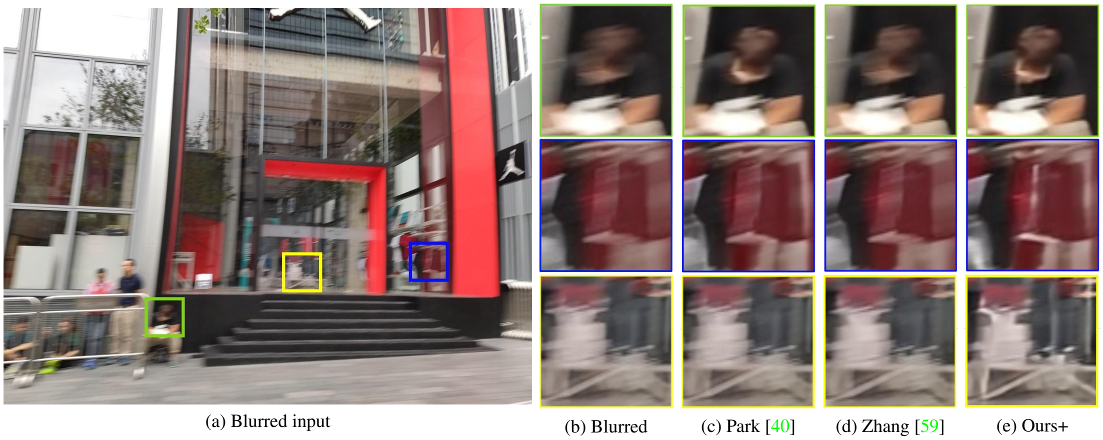


Figure 6. Qualitative comparison on the RWBI dataset: a) The blurred input image. b-e) Magnified crops of the blurred input and the deblurred outputs of compared methods. Note that no ground truth is available for RWBI.

al. [45] is trained on the HIDE dataset directly but it cannot perform better than our proposed method.

The average testing time and GPU memory usage on the GoPro dataset is reported in Table 3. Although the testing time of Park *et al.* [40] is the lowest, its GPU memory usage is almost four times of ours, and its performance (Table 1) is lower than ours. Our method is 30% faster than Zhang *et al.* [59] while only increasing the GPU memory usage by 7%. Since the architecture of Kupyn *et al.* [29] is much deeper and wider than that of other listed methods, its testing time is the highest even if it uses only a single stage. Our proposed method is a good trade-off between performance and efficiency in both memory and computation.

4.4. Qualitative Comparison

Following a similar strategy as in the quantitative comparison, we first compare our method against some others on the testing images of the GoPro dataset [37]. We compare with the two best performing and most recent methods [40, 59] with published well-trained models, along with the published results of Purohit *et al.* [42] on the same dataset. As apparent in Figure 4, the output of our method is most similar to the ground truth sharp image, comparing to others. The alphabets written on the banner of the first row are almost clearly visible for ours, while others fail to deblur them correctly. Similarly, the sign over the shop is best deblurred by ours, while the result of Purohit *et al.* [42] is also reasonably good. On the third row, our method restores the skin and bright colors better. We also use the well-trained models on the GoPro dataset to test on the HIDE dataset [45]. As visible in Figure 5, our method does a good job deblurring the bicycles in the image. We further test the same models on the RWBI dataset [61]. As seen in Figure 6, our method generally outperforms the compared methods in terms of deblurring quality. For a more complete qualitative study, please refer to the supplementary material.

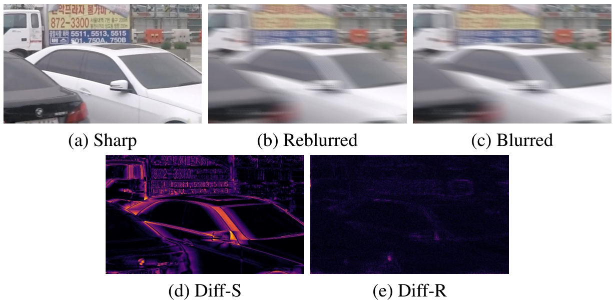


Figure 7. Comparison of a reblurred output with its corresponding blurred and sharp images. Diff-S (Diff-R) represents the difference map between the sharp image (reblurred output) and the blurred ground truth.

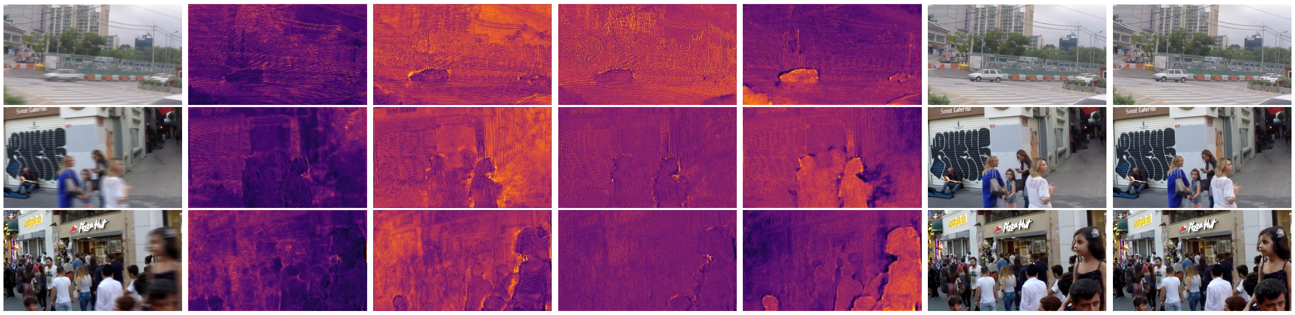
PSNR	SSIM	Mean	variance
55.71	0.9997	0.18	0.22

Table 4. Performance evaluation of reblurring network on the GoPro dataset.

4.5. Reblurring Evaluation

In addition to evaluation of the deblurring network, we evaluate the performance of the reblurring network since it is critical for the fine-tuning. As shown in Table 4, PSNR and SSIM of reblurred images are quite high, and the mean and variance of the difference map $|I_r - I_b|$ are almost 0. The experimental result illustrates that reblurred images are quite close to the original blurred images.

One of the reblurred outputs is shown in Figure 7 with the original sharp and blurred images. As we can see, the difference between the reblurred and blurred image is small enough and negligible.



(a) Blurred (b) a_1 (c) a_2 (d) a_3 (e) a_4 (f) Deblurred (g) Sharp

Figure 8. Attention maps of the last ASPDC module. Values are within [0, 1] and the brighter the higher.

Version	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
Module 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Module 2		✓			✓	✓		×3			✓	✓
Module 3			✓		✓		✓		×3		✓	✓
Module 4				✓		✓	✓			×3	✓	✓
AFIM		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
PSNR	30.24	30.65	30.94	30.85	31.82	31.73	31.53	31.83	31.93	31.72	31.16	32.12
SSIM	0.942	0.944	0.948	0.947	0.957	0.956	0.954	0.957	0.958	0.955	0.950	0.959

Table 5. Performance of different versions of the ASPDC module.

λ	0.01	0.1	1
PSNR	32.17	32.22	32.15
SSIM	0.960	0.960	0.959

Table 6. Performance of fine-tuning with different λ.

4.6. Ablation Study

To evaluate the effectiveness of each component in the ASPDC module, we compare multiple versions of it. We randomly select 200 testing images from the GoPro dataset for the validation. All versions are trained with the mean squared error loss only without fine-tuning. With regard to the performance and efficiency trade-off, we find having four ASPDC modules is optimal.

The experimental results are shown in Table 5. Version 1 has a deformable module 1 without an offset Δp, which is used as our baseline. The $\times 3$ in Version 8∼10 represents three duplicated modules (the same dilation rate but no parameter-sharing). As we can see, the deformable module 2∼4 is critical for improving performance, especially module 3. Simply duplicating modules cannot get results as good as combining different modules. It demonstrates that fusing information from different sizes of receptive fields is meaningful and justified. Version 11 demonstrates that features from different receptive fields should be fused properly. We visualize the attention maps of the last (sixth) ASPDC module in Figure 8. It shows that attention maps of small receptive fields (a_1 and a_2) focus more on static objects or objects with small movements, while attention maps of large receptive fields (especially a_4) pay more attention on objects with large movements.

For the choice of the hyperparameter λ in Eqn 7, we evaluate values in the range of 0.01 to 1 in Table 6. As we can see, the deblurring term is overwhelmed by the reblurring term when the value of λ is too large. Inversely, a small value of λ, such as 0.01, limits the effect of the reblurring term on the improvement of performance. To fully utilize the reblurring term, we set $\lambda = 0.1$ in all our experiments.

5. Conclusion

In this paper, we propose a novel end-to-end blind non-uniform motion deblurring network with new ASPDC modules, which are able to apply region-specific convolution to each pixel and integrate features from different receptive fields. Compared to SOTA methods, our method achieves better performance with high efficiency. In addition, the performance can be further improved by fine-tuning on the proposed reblurring network. In future, we plan to address the fact that none of the existing methods perform well when the magnitude of motion gets too large, resulting in issues such as color degradation or even failure in deblurring. We further plan to study deblurring-reblurring consistency of non-uniform deblurring in an unsupervised setting with no access to blurred-sharp pairs for the reblurring network.

References

[1] Insaf Adjabi, Abdeldjalil Ouahabi, Amir Benzaoui, and Abdelmalik Taleb-Ahmed. Past, present, and future of face recognition: a review. *Electronics*, 9(8):1188, 2020. 1

[2] Raied Aljadaany, Dipan K Pal, and Marios Savvides. Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution. In *CVPR*, 2019. 1, 3, 5, 6

[3] Saeed Anwar, Cong Phuoc Huynh, and Fatih Porikli. Class-specific image deblurring. In *ICCV*, 2015. 2

[4] Yuval Bahat, Netalee Efrat, and Michal Irani. Non-uniform blind deblurring by reblurring. In *ICCV*, 2017. 1, 2

[5] Yuanchao Bai, Huizhu Jia, Ming Jiang, Xianming Liu, Xiaodong Xie, and Wen Gao. Single image blind deblurring using multi-scale latent structure prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):2033–2045, 2019. 2

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 3

[7] Yongpeng Dai, Tian Jin, Yongkun Song, Shilong Sun, and Chen Wu. Convolutional neural network with spatial-variant convolution kernel. *Remote Sensing*, 12(17):2811, 2020. 4

[8] Zijun Deng, Lei Zhu, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Qing Zhang, Jing Qin, and Pheng-Ann Heng. Deep multi-model fusion for single-image dehazing. In *ICCV*, 2019. 4

[9] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, 2019. 4

[10] Jiangxin Dong, Jinshan Pan, Deqing Sun, Zhixun Su, and Ming-Hsuan Yang. Learning data terms for non-blind deblurring. In *ECCV*, 2018. 1

[11] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992. 3

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 6

[13] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 1, 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 6

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 2

[16] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *CVPR*, 2020. 2, 4

[17] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *ECCV*, 2010. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4

[19] Dong Huo and Yee-Hong Yang. Blind image super-resolution with spatial context hallucination. *arXiv preprint arXiv:2009.12461*, 2020. 3

[20] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *ICCV*, 2013. 2

[21] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *CVPR*, 2014. 2

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[23] Jiaya Jia. Single image motion deblurring using transparency. In *CVPR*, 2007. 2

[24] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 3, 5

[25] Adam Kaufman and Raanan Fattal. Deblurring using analysis-synthesis networks pair. In *CVPR*, 2020. 1

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *ECCV*, 2016. 3

[28] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*, 2018. 1, 2, 5, 6

[29] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019. 1, 2, 5, 6, 7

[30] Stan Z Li. Markov random field models in computer vision. In *ECCV*, 1994. 2

[31] Guangcan Liu, Shiyu Chang, and Yi Ma. Blind image deblurring using spectral properties of convolution operators. *IEEE TIP*, 23(12):5047–5056, 2014. 1

[32] Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016. 2

[33] Boyu Lu, Jun-Cheng Chen, and Rama Chellappa. Unsupervised domain-specific deblurring via disentangled representations. In *CVPR*, 2019. 4

[34] Rastislav Lukac. *Computational photography: methods and applications*. CRC press, 2017. 1

[35] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014. 2

[36] Manasa Nadipally. Chapter 2 - optimization of methods for image-texture segmentation using ant colony optimization. In D. Jude Hemanth, Deepak Gupta, and Valentina Emilia Balas, editors, *Intelligent Data Analysis for Biomedical Applications*, Intelligent Data-Centric Systems, pages 21 – 47. Academic Press, 2019. 6

[37] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7

[38] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. l_0-regularized intensity and gradient prior for deblurring text images and beyond. *IEEE TPAMI*, 39(2):342–355, 2016. 1

[39] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016. 1

[40] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *ECCV*, 2020. 1, 2, 3, 5, 6, 7

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[42] Kuldeep Purohit and AN Rajagopalan. Region-adaptive dense network for efficient motion deblurring. In *AAAI*, 2020. 1, 2, 5, 6, 7

[43] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. 1

[44] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *CVPR*, 2020. 1

[45] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 5, 7

[46] David Strong and Tony Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse problems*, 19(6):S165, 2003. 1

[47] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, 2019. 4

[48] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *CVPR*, 2020. 1, 2, 3, 5, 6

[49] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015. 2, 5, 6

[50] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 1, 2, 3, 5, 6

[51] Nimisha Tm, Vijay Rengarajan, and Rajagopalan Ambasamudram. Semi-supervised learning of camera motion from a blurred image. In *ICIP*, 2018. 2

[52] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2, 4

[53] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[54] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 5, 6

[55] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *CVPR*, 2017. 1

[56] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019. 4

[57] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 4

[58] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *CVPR*, 2020. 1, 3, 4, 5, 6

[59] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7

[60] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*, 2018. 1, 2, 5, 6

[61] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020. 1, 3, 5, 6, 7

[62] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. 1

[63] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 3, 4

[64] Wangmeng Zuo, Dongwei Ren, David Zhang, Shuhang Gu, and Lei Zhang. Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution. *IEEE TIP*, 25(4):1751–1764, 2016. 1