Self-Supervised Linear Motion Deblurring

Peidong Liu[1], Joel Janai[2], Marc Pollefeys[1,3], Torsten Sattler[4] and Andreas Geiger[2]

***Abstract*—Motion blurry images challenge many computer vision algorithms, e.g., feature detection, motion estimation, or object recognition. Deep convolutional neural networks are state-of-the-art for image deblurring. However, obtaining training data with corresponding sharp and blurry image pairs can be difficult. In this paper, we present a differentiable reblur model for self-supervised motion deblurring, which enables the network to learn from real-world blurry image sequences without relying on sharp images for supervision. Our key insight is that motion cues obtained from consecutive images yield sufficient information to inform the deblurring task. We therefore formulate deblurring as an inverse rendering problem, taking into account the physical image formation process: we first predict two deblurred images from which we estimate the corresponding optical flow. Using these predictions, we re-render the blurred images and minimize the difference with respect to the original blurry inputs. We use both synthetic and real dataset for experimental evaluations. Our experiments demonstrate that self-supervised single image deblurring is really feasible and leads to visually compelling results. Both the code and datasets are available at https://github.com/ethliup/SelfDeblur.**

***Index Terms*—Computer Vision for Automation, Deep Learning in Robotics and Automation**

I. Introduction

MOTION blur is one of the most common factors degrading image quality. It often arises when the image content changes quickly (e.g., due to fast camera motion) or when the environment is illuminated poorly, hence necessitating longer exposure times. Combining both situations, e.g., a self-driving car driving at dusk, further aggravates the problem. As many computer vision algorithms such as semantic segmentation, object detection, or visual odometry rely on visual input, blurry images challenge the performance of these algorithms. It is well known that many algorithms (e.g., depth prediction, feature detection, motion estimation, or object recognition) suffer from motion blur [17], [25], [26], [33]. The motion deblurring problem has thus received considerable attention in the past [7], [17], [21], [28], [32].

Existing techniques to solve this problem can be classified into two categories: the first type of approaches formulate the problem as an optimization problem [2], [4], [16], [18], [28], [37] where the latent sharp image and/or the blur kernel are optimized using gradient descent. One of the advantages of this kind of methods is that they do not require any ground truth sharp images. However, the resulting solvers usually have a large computational complexity, which limits their applicability in time-constrained settings, such as real-time robotic visual perception. Handcrafted priors on either the image or the blur kernel further limit their performances.

The second type of approaches phrase the task as a learning problem. Building upon the recent advances of deep convolutional neural networks, state-of-the-art results have been obtained for both single image deblurring [21], [32] and video deblurring [29], outperforming optimization-based techniques in terms of both quality and efficiency. However, learning-based methods typically require full supervision in the form of corresponding pairs of blurred and sharp images. Unfortunately, obtaining such pairs is not always easy due to two main reasons. One is that not every camera has the capability to capture images at enough high frame rate (1000 or more frames per second), such that we can use the recorded frames to synthesize the training data. Another reason is that it would also be difficult to obtain good quality images in real scenarios where the motion blur really occurs (e.g., at night). High frame rate limits the exposure time and would thus make the captured image extremely dark or even invisible.

Inspired by recent progress in self-supervised depth [5], [41], flow [12], [20] and representation learning [3], [24], we propose a novel approach for self-supervised image deblurring which only relies on real-world blurry image sequences for training. Self-supervised learning improves the network's generalization performance, by enabling the network to adapt to scenarios where ground truth sharp images are not available. Our network contains a deblurring network and an optical flow estimation network. However, instead of using ground truth sharp images [9], [17], [21], [32], we pose the task as an inverse rendering problem and take advantage of the physical image formation process for supervision. More specifically, given two consecutive blurry frames of a video sequence, we first predict the corresponding deblurred images using a deep neural network. A second deep neural network takes both deblurred images as input and computes the corresponding optical flow. Using this prediction, and assuming a locally linear blur kernel, our model re-renders the blurred images and compares the results to the original blurry inputs using a photometric loss function. Moreover, we constrain the optical flow network using a photo-consistency loss function. Our entire model can be trained end-to-end from pairs of consecutive blurry images captured with a consumer video camera. At test time, our network takes a single blurred image

Manuscript received: September, 10, 2019; Revised December, 06, 2019; Accepted January, 20, 2020.

This paper was recommended for publication by Editor Cesar Cadena Lerma upon evaluation of the Associate Editor and Reviewers' comments.

[1] Peidong Liu and Marc Pollefeys are with Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich, Switzerland peidong.liu@inf.ethz.ch

[2] Joel Janai and Andreas Geiger are with Autonomous Vision Group, Max Planck Institute for Intelligent Systems and Univeristy of Tübingen, Tübingen, Germany

[3] Marc Pollefeys is with Microsoft Mixed Reality and Artificial Intelligence Lab, Zürich, Switzerland

[4] Torsten Sattler is with Computer Vision and Medical Image Analysis Group, Chalmers University of Technology, Sweden

Digital Object Identifier (DOI): see top of this page.

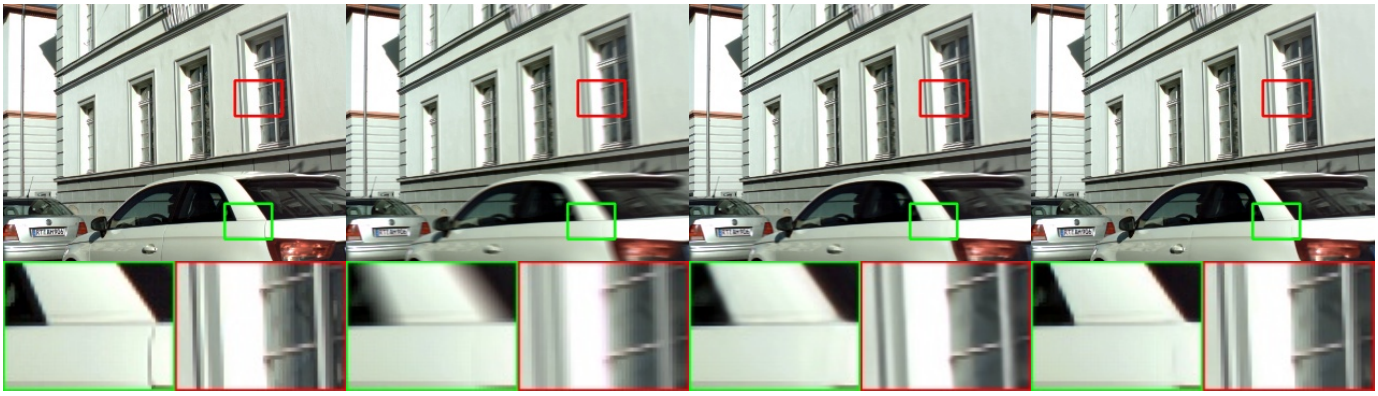

Fig. 1: **Self-supervised motion deblurring. First:** Sharp ground truth image. **Second:** Blurry input image. **Third:** Deblurring results of our self-supervised method. **Fourth:** Deblurring results of the supervised method from Tao et al. [32].

and deblurs it in real time on a single GTX 1080Ti graphic card using the learned parameters. As illustrated in Fig. 1, our approach is competitive with respect to a state-of-the-art supervised method [32] despite being fully self-supervised.

Our second contribution is a novel synthetic dataset and a real dataset. The synthetic dataset has 3606 blurry-sharp image pairs recorded with a professional high-speed camera mounted on a ground vehicle. The real dataset has 2302 blurry images and is recorded with a normal camera. We use both datasets to evaluate our algorithm against several baselines both quantitatively and qualitatively.

II. Related Work

Motion deblurring methods can be categorized into two groups: those that assume spatially uniform blur [2], [4], [16], [18], [27], [28], [37] and those considering spatially varying blur [7], [31], [35]. Uniform deblurring methods assume that the blur kernel is identical for each pixel of the input image. Spatially varying deblurring methods assume that the blur kernels for each pixel may change with respect to its spatial location. Motion deblurring methods can also be classified into non-blind deblurring [16], [27], [31] and blind deblurring [2], [4], [7], [18], [28], [35], [37] methods. Non-blind deblurring methods assume a known blur kernel to recover the latent sharp image. In contrast, blind deblurring methods need to simultaneously recover both the latent image and the blur kernel. In this paper, we solve the challenging blind single image deblurring problem with spatially varying blur.

Optimization-based methods rely on the blur formation model to recover the latent sharp image by minimizing an energy function [2], [4], [16], [18], [28], [37], [42], e.g., using Gaussian [2], [16], [37] or Poisson [27], [31] likelihood functions in the context of maximum-a-posteriori (MAP) estimation. Depending on the number of input blurry images, additional terms can be formulated by warping the other image(s) to a reference image using either dense flow or by combining relative camera poses with dense depth maps [14], [22]. Due to the nonlinear and ill-posed (in the case of a single image) nature of the problem, prior information on either the motion blur kernel or the latent sharp image must be used to constrain the solution space [2], [4], [16], [16], [28], [37], [37].

While optimization based approaches often offers better generalization performance, they are usually computational expensive, which prevents them from time constrained applications. We leverage the commonly used image formation model by optimization based approaches to construct a loss term for training our network in a self-supervised fashion. Since we optimize the parameters of our network once, the computationally complex optimization problem does not occur at test time, where we use the standard efficient feed-forward inference.

Deep Learning based methods use convolutional neural networks (CNNs) to recover the latent sharp image. Xu et al. [38] propose a CNN with two sub-networks, a two-hidden-layer deconvolution network and a two-hidden-layer outlier rejection network. The network is trained end-to-end with known ground truth sharp images. An even deeper network with 15 layers was proposed in [8] for text image deblurring. [21] further increased the number of layers to 40 in a multi-scale manner, resulting in a network with 120 layers for three scales. To further improve the network performance, an adversarial loss [6] was used in [17]. [32], [40] use recurrent neural networks for single image deblurring.

More recently, network architectures for multi-frame inputs, which are able to exploit temporal information, have been proposed [15], [29], [36]. Su et al. [29] uses a network with skip connections for video deblurring and Wieschollek et al. [36] exploit temporal information using a recurrent architecture. A spatio-temporal recurrent architecture with a small computational footprint was proposed in [15].

Deep learning-based approaches usually outperform optimization-based methods in terms of both efficiency and image quality. However, nearly all of the recent deep learning based methods are trained in a fully supervised manner with only few notable exceptions. Madam et al. [19] adapted CycleGAN [43] to the single image deblurring task. Using unpaired sharp images for training, they obtained good performance in the specific domain of images (e.g., text, faces). In contrast to our approach, they require (unpaired) sharp images for training and perform poorly on blurry images

"in the wild" as demonstrated by our experiments. Chen et al. [1] propose to include a self-consistency loss for supervision. However, they report that their self-supervised model leads to degenerated solutions and hence optimize a hybrid loss function which heavily relies on supervision in the form of sharp images. Furthermore, in contrast to our approach, their method requires triplets of blurry images for training and uses a memory and computational expensive look-up table for representing the blur kernel in a differentiable fashion. We avoid this problem using forward warping and differentiable mesh rendering.

III. Method

Fig. 1 shows the overall architecture of our model. It comprises four main parts, i.e., the DeblurNet, the FlowNet, the reblur block, and image warping. The DeblurNet is used for single image motion deblurring. It accepts a single blurry image as input and outputs the corresponding sharp image. The two deblurred images are then fed into the FlowNet to estimate a bi-directional dense optical flow field, which will be used to compute spatially varying blur kernels for each blurry image. Given the estimated blur kernels, we reblur the latent sharp image to form a self-consistency loss to supervise the training of our network. The FlowNet is trained by maximizing cross-view photometric consistency, which is estimated by image warping. While our approach uses two images for training, the DeblurNet only uses a single input image. After training, our method can thus be used for single image deblurring. The whole network is trained end-to-end without using any ground truth data in the form of sharp images or optical flow. We will now present all components of our model (i.e., the deblurring, optical flow, reblurring and image warping components as well as the loss functions) in detail.

A. Deblurring and Optical Flow

For the deblurring and optical flow modules, we take advantage of existing neural network architectures which have performed well in the past for the respective supervised learning tasks [9], [17], [21], [30], [32]. In particular, we adopt the single image deblurring network from Tao et al. [32] and the dense optical flow estimation network PWC-Net from Sun et al. [30]. We make the following modifications for the deblurring network for our particular problem: 1) We replace the deconvolution layer with bilinear upsampling followed by a 3x3 convolution to avoid upsampling artifacts. 2) We add one more Encoder-Decoder block to increase the capacity of the network. 3) We train the network at a single scale without using the LSTM layer to improve both the training and test efficiency. The resulting network is more efficient than the original network while keeping similar deblurring performance.

B. Reblurring

The reblurring module encapsulates the physical image formation process, which blurs a sharp image based on the optical flow. Digital cameras operate by collecting photons during the time of exposure and converting those into measurable charge. This process can be formalized by considering the blurred color image $\mathbf{B} \in \mathbb{R}^{W \times H \times 3}$ as the result of integrating virtual sharp images $\mathbf{I}_t \in \mathbb{R}^{W \times H \times 3}$:

$$\mathbf{B}(\mathbf{x}) = \int_0^\tau \mathbf{I}_t(\mathbf{x}) \mathrm{dt} \approx \frac{1}{2N+1} \sum_{i=-N}^{N} \mathbf{I}_i(\mathbf{x}) \tag{1}$$

Here, τ is the exposure time, $\mathbf{x} \in \mathbb{R}^2$ represents the pixel location, $\mathbf{B}(\mathbf{x})$ denotes the motion blurred image at pixel $\mathbf{x}$, and $\mathbf{I}_t(\mathbf{x})$ is the virtual sharp image at pixel $\mathbf{x}$ and time t. The continuous integration can be approximated by using a finite sample size of $2N+1$ virtual sharp frames $\mathbf{I}_i$. We denote the central reference frame, which is the latent sharp image to be estimated, as $\mathbf{I}_0$.

As the exposure time τ is typically small (<200 ms), we may assume that during the time of exposure the image content is primarily affected by image motion and not by other changes like object appearance or illumination. We thus model the virtual sharp frames $\mathbf{I}_i$ as the result of the sharp central reference frame $\mathbf{I}_0$ warped by optical flow $\mathbf{u}_{i\to 0}$:

$$\mathbf{I}_i(\mathbf{x}) = \mathbf{I}_0(\mathbf{x} + \mathbf{u}_{i\to 0}) \tag{2}$$

Here, $\mathbf{u}_{i\to 0} \in \mathbb{R}^2$ denotes the optical flow from virtual image $\mathbf{I}_i$ to reference image $\mathbf{I}_0$ at pixel $\mathbf{x}$. Thus, we can reformulate (1) as

$$\mathbf{B}(\mathbf{x}) \approx \frac{1}{2N+1} \sum_{i=-N}^{N} \mathbf{I}_0(\mathbf{x} + \mathbf{u}_{i\to 0}) \tag{3}$$

and estimate $\mathbf{I}_0$ as well as the optical flow fields $\mathbf{U}_{i\to 0}$ instead of all virtual frames $\mathbf{I}_i$ for solving the deblurring problem. However, the problem is still severely underconstrained as we would need to estimate one optical flow per frame $i \in \{-N, \dots, N\}$.

We therefore further simplify the model by assuming linear motion during the time of exposure. This is a reasonable assumption in many scenarios where the exposure time is comparably small and rapid motion changes during this time are prevented by the mass and inertia of physical objects (e.g., when the camera is mounted to a vehicle).

Let $\mathbf{u} \in \mathbb{R}^2$ denote the optical flow from frame $\mathbf{I}_0$ to frame $\mathbf{I}_1$ at pixel $\mathbf{x}$. Assuming linear motion and equidistant time steps, we obtain the optical flow from frame 0 to frame i as $\mathbf{u}_{0\to i} = i \cdot \mathbf{u}$. Note that in this model the direction of the optical flow is reversed compared to (3). We must therefore apply forward warping to obtain the virtual sharp images $\mathbf{I}_i$. This yields

$$\mathbf{B}(\mathbf{x}) \approx \frac{1}{2N+1} \sum_{i=-N}^{N} (\mathcal{W}_{0\to i} \circ \mathbf{I}_0)(\mathbf{x}) \tag{4}$$

where the operator $\mathcal{W}_{0\to i}$ warps the reference frame $\mathbf{I}_0$ into the virtual frame $\mathbf{I}_i$ based on the interpolated flow $\mathbf{u}_{0\to i}$. We next describe our implementation of the forward warping operator $\mathcal{W}_{0\to i}$. Note that this operator needs to be differentiable as both the reference frame $\mathbf{I}_0$ and the optical flow $\mathbf{U}$ are outputs of neural networks.

We first construct a regular triangular lattice from the pixel grid by connecting vertices from adjacent pixels as shown in

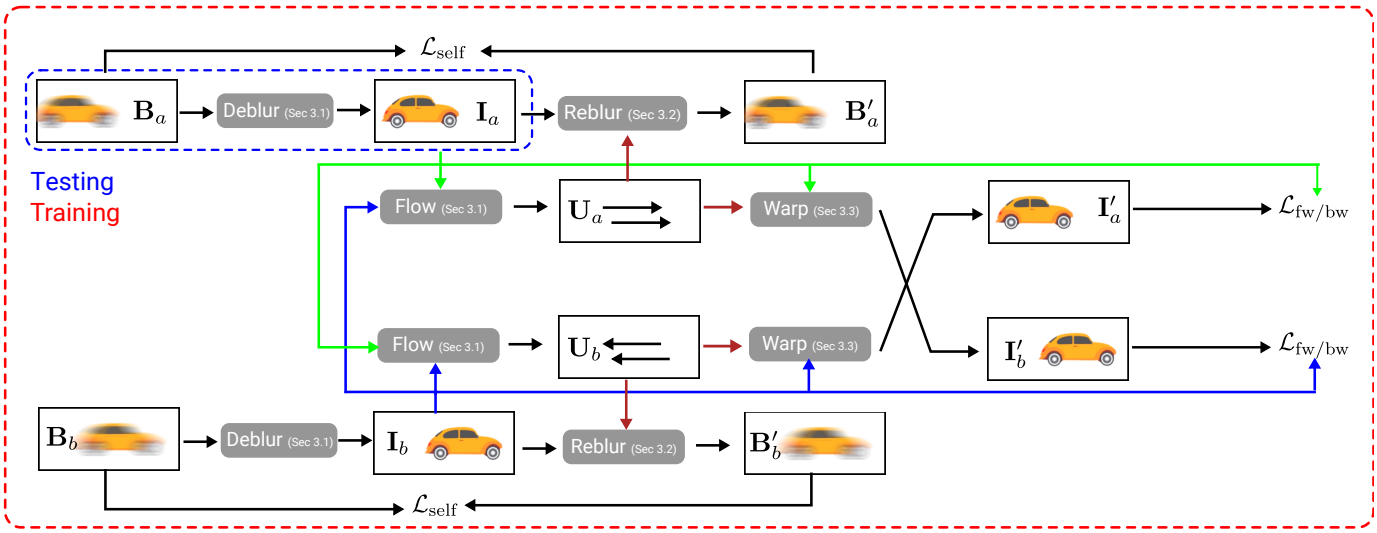

Fig. 2: **Architecture of the proposed network.** Given two consecutive blurry images, $\mathbf{B}_a$ and $\mathbf{B}_b$, as input, our network computes the corresponding deblurred images, $\mathbf{I}_a$ and $\mathbf{I}_b$, as well as the bidirectional optical flow, $\mathbf{U}_a$ and $\mathbf{U}_b$. To self-supervise the training of the network, we construct a self-consistency photometric loss $\mathcal{L}_{\text{self}}$ and a forward-backward photometric consistency losses, $\mathcal{L}_{\text{fw/bw}}$.

Fig. 3 for an example image of size 3×3 pixels. We then warp each vertex of this lattice according to the optical flow $\mathbf{u}_{0\to i}$. The intensities of $\mathbf{I}_i$ (i.e., of the blue pixels) are obtained by linear interpolation [a]. Consider the red pixel $\mathbf{x}$ in Fig. 3 as an example. Let further $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the positions of the vertices belonging to the triangle which covers the red pixel. Then, $\mathbf{I}_i$ is obtained as

$$\mathbf{I}_i(\mathbf{x}) = \omega_0 \mathbf{I}_0(\mathbf{x}_0) + \omega_1 \mathbf{I}_0(\mathbf{x}_1) + \omega_2 \mathbf{I}_0(\mathbf{x}_2) \tag{5}$$

where ω_0, ω_1 and ω_2 denote the barycentric coordinates of the point in the triangle. The synthesized motion blurred image can then be computed as the average of all the warped frames as described in (1). In the case of occlusions, i.e., when multiple triangles overlap a single pixel, we consider the triangle with the largest motion to be in front. This is a commonly used heuristics [13] which often holds true in practice (in particular when image motion is dominated by camera motion). Note that due to the linear interpolation, the warping function $\mathcal{W}_{0\to i}$ is piecewise smooth. As illustrated in Fig. 1, our model is symmetric, thus we reblur both the first and the second frame and compare the reblurred result to the original blurry images using a photoconsistency loss. We will use $\mathbf{B}'$ to denote the reblurred image and $\mathbf{B}$ to denote the blurred input image in the following.

C. Image Warping

We found that a photoconsistency loss on the reblurred images alone is insufficient to constrain the optical flow. We thus add an additional self-supervised photometric loss on the optical flow as proposed in prior work [12], [20], [34] and detailed in Section III-D. The input to this loss function is

[a]Note that bilinear interpolation cannot be applied since the warped grid might not be rectangular due to the non-uniform optical flow, as shown in Fig. 3.

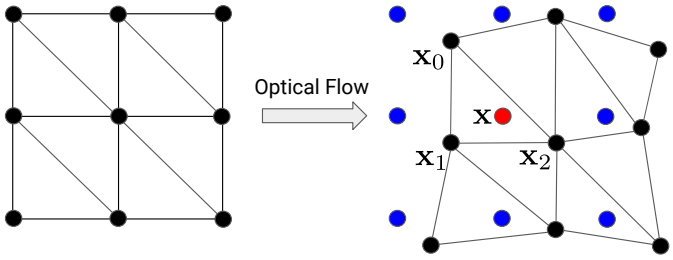

Fig. 3: **Differentiable forward warping.** We construct a regular triangular lattice from the pixel grid of the reference image $\mathbf{I}_0$ (left). We then warp each vertex of this lattice according to the optical flow $\mathbf{u}_{0\to i}$. The intensities of $\mathbf{I}_i$ (i.e., of the blue pixels) are obtained by linear interpolation.

the deblurred image and the deblurred image from the other frame warped based on the estimated optical flow. To warp the images into each other, we exploit backward warping as the optical flow in both directions is known. Let $\mathbf{I}_a \equiv \mathbf{I}_0^a$ and $\mathbf{I}_b \equiv \mathbf{I}_0^b$ denote the first and the second deblurred image, and let $\mathbf{u}_{a\to b}$ and $\mathbf{u}_{b\to a}$ denote the optical flow between them[b]. The warped deblurred images are obtained as

$$\mathbf{I}'_a(\mathbf{x}) = \mathbf{I}_b(\mathbf{x} + \mathbf{u}_{a\to b}) \tag{6}$$

$$\mathbf{I}'_b(\mathbf{x}) = \mathbf{I}_a(\mathbf{x} + \mathbf{u}_{b\to a}) \tag{7}$$

using bilinear interpolation [11]. Note that no triangular mesh needs to be constructed during backward warping.

D. Loss Functions

Our network comprises two types of losses: a self-consistency loss $\mathcal{L}_{\text{self}}$ and a forward-backward consistency loss

[b]Note that the flow $\mathbf{u}_{a\to b}$ / $\mathbf{u}_{b\to a}$ and $\mathbf{u}$ from the previous section are related by a known constant that depends on the frame rate, exposure time and N.

$\mathcal{L}_{\text{fw/bw}}$. The self-consistency loss

$$\mathcal{L}_{\text{self}} = \|\mathbf{B}'_a - \mathbf{B}_a\|_1 + \|\mathbf{B}'_b - \mathbf{B}_b\|_1 \tag{8}$$

penalizes differences between the synthesized motion blurred images $\mathbf{B}'_a$, $\mathbf{B}'_b$ and the original blurred inputs $\mathbf{B}_a$, $\mathbf{B}_b$ using a ℓ_1 loss function. Similarly, the forward-backward consistency loss

$$\mathcal{L}_{\text{fw/bw}} = \|\mathbf{I}'_a - \mathbf{I}_a\|_1 + \|\mathbf{I}'_b - \mathbf{I}_b\|_1 \tag{9}$$

penalizes differences between the warped deblurred images $\mathbf{I}'_a$, $\mathbf{I}'_b$ and the estimated deblurred images $\mathbf{I}_a$, $\mathbf{I}_b$. The final loss is a weighted combination

$$\mathcal{L} = \mathcal{L}_{\text{self}} + \lambda \mathcal{L}_{\text{fw/bw}}, \tag{10}$$

where λ is a hyper-parameter to balance both losses.

E. Occlusion Handling

As occlusions affect the training of our network, especially at image boundaries, we detect occluded image regions and mask the loss functions (8) and (9) accordingly. We follow the method used in [34] to detect occluded image regions. More specifically, we compute the non-occluded regions in $\mathbf{I}_a$ by following the optical flow $\mathbf{U}_{b\to a}$ from $\mathbf{I}_b$ to $\mathbf{I}_a$. We consider all pixels of $\mathbf{I}_a$ which can be reached from $\mathbf{I}_b$ via $\mathbf{U}_{b\to a}$ as non-occluded. Similarly, we can also compute a mask for each virtual frame by following the optical flow from the central image to the virtual frame $\mathbf{u}_{0\to i}$. Since the synthesized blurry image is computed as the average of these virtual frames, we compute the final mask for $\mathcal{L}_{self}$ as the product of all masks for the virtual frames.

F. Differences with the method proposed by [1]

The overall structure of our method is similar to the work from [1]. However, we are different in the following two key aspects: **1)** In order to achieve state-of-the-art performance, [1] uses a supervised loss (section 3.4 from [1]). [1] is thus actually a supervised method. **2)** The core component of both methods, i.e., the reblurring module, is different. In fact, blurring a sharp image using convolutions as done in [1] is physically incorrect (section 3.3 from [1]) and only holds for spatially uniform blur. We will make the differences to [1] more clear.

For simplicity, let us assume we have a one dimensional sharp image $\mathbf{I}$ with N pixels. We further assume the blur kernel corresponds to pixel $\mathbf{I}_i$ as $\{-2, 2\}$ in the form of bidirectional 1D flow. Using the definition from [1], the convolution based model results in a blurred image of $\mathbf{I}_i$ as $\mathbf{B}_i = \frac{1}{5}\sum_{j=-2}^{2} \mathbf{I}_{i+j}$. A blur kernel $\{-2, 2\}$ of $\mathbf{I}_i$ means that $\mathbf{I}_i$ will contribute to $\mathbf{B}_{i-2}, \mathbf{B}_{i-1}, \mathbf{B}_i, \mathbf{B}_{i+1}, \mathbf{B}_{i+2}$ physically, in contrast to that $\mathbf{I}_{i-2}, \mathbf{I}_{i-1}, \mathbf{I}_i, \mathbf{I}_{i+1}, \mathbf{I}_{i+2}$ will contribute to $\mathbf{B}_i$ as what the convolution based model in [1] does. Our model eliminates this problem by forward warping the sharp image $\mathbf{I}$ by a fraction of the blur kernels at each sampled timestamp. The blurred image is computed by averaging all these forward warped sharp images to simulate the real motion blurring image formation process. Experimental results shown later demonstrate that the algorithm relying on convolution based model exhibits ringing artifacts on egde boundaries, which degrade the deblurred images.

IV. Experimental Evaluation

Datasets: The dataset from [21] is commonly used to benchmark single image motion deblurring algorithms. It is collected from a hand-held camera. Stronger blur was artificially created by shaking the camera during the recordings. It results in very non-linear camera motions, which violates our motion assumption. Therefore, we collected a new large dataset using a professional Fastec TS5[c] high speed camera mounted on a car. The dataset consists of 196 sequences in total, which are collected at 1200 fps with VGA resolution in diverse environments. The motion blurred images are generated by averaging several consecutive frames (i.e., 1∼50 frames) to simulate the real physical image formation process. To reduce the redundancy per image sequence, we limit the maximum number of blurry-sharp image pairs to 20 per sequence, which results in a total of 3606 pairs. We split the dataset into 157 training sequences and 39 test sequences, which results in 2820 image pairs for training and 786 image pairs for evaluation.

We also collect a real motion blurry dataset with 2302 images. The camera is mounted on a tram and captures images at around 50 FPS with a resolution of 752×480 pixels. The dataset is collected at late afternoon and night, when the motion blur would really occur. We split 2062 images for training and 240 images for test.

Implementation details: We implemented our network in PyTorch [23]. We empirically set the hyper-parameter λ to be 2.0. To better initialize the network, we pretrain both the DeblurNet and PWC-Net on the blurry images. In particular, we pretrain the DeblurNet for 30 epochs to learn the identity mapping from blurry image to blurry image. We pretrain the PWC-Net for 200 epochs with the blurry sequences in a self-supervised manner. The learning rate used for both networks is 10^{-4}. The whole network is then trained jointly for another 500 epochs, with a learning rate of 10^{-4} for the first 260 epochs and then decayed by half every 40 epochs.

Baselines and experimental settings: We compare the single image deblurring results of our network quantitatively and qualitatively with a state-of-the-art optimization-based method [39], supervised methods [17], [21], [32] as well as the domain specific self-supervised method from [19]. We train all networks with their recommended hyperparameter settings on our synthetic dataset. For the optimization-based method from [39], we increase the blur kernel size to 10 pixels to account for the large motions present in our dataset.

Evaluation metrics: We use the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) measures commonly used in the community [21], [28], [32] to evaluate the quality of the deblurring results. Larger PSNR/SSIM values indicate better image quality. The efficiency of the methods is evaluated by their total time consumption, but excluding the image loading and saving time.

Ablation studies on the modified DeblurNet architecture: As discussed in Sec.III-A, we did several improvements to

[c]https://www.fastecimaging.com/fastec-high-speed-cameras-ts-series/

Network	PSNR↑	SSIM↑	Time ↓
SRN-Deblur [32]	34.64 dB	0.93	0.13 s
Ours (supervised)	35.04 dB	0.94	0.05 s

TABLE I: **Single image deblurring comparison on the synthetic dataset.** We compare our modified network with the original network from Tao et al. [32] by training both in a supervised way.

the original network from [32]. To evaluate the efficacy of the new network, we train both networks in a supervised manner on our synthetic dataset. Table I presents the comparisons when evaluated under the same settings. It demonstrates our DeblurNet is more efficient than the original network while has slightly better deblurring performance.

Ablation studies on the self-supervision loss for the flow network: To better understand the proposed algorithm, we perform an ablation study on the necessity to train the flow network in a self-supervised manner. We train our proposed network with and without the self-supervision loss for the flow network. The officially provided pretrained model on FlyingChair dataset [10] is used if the self-supervision loss is disabled. Experimental results demonstrate that the flow network pretrained on the FlyingChair dataset [10] can generalize to our dataset, but with limited performance. The resulting deblur network gives a PSNR metric as 31.23dB and a SSIM metric as 0.89 on our synthetic dataset, in contract to 32.24dB/0.91 if the network is trained in a fully self-supervised manner. It proves it is beneficial to train the flow network with a self-supervision loss.

Ablation studies on our proposed reblur model: As discussed in Sec.III-F, our ablation study supports our claim about the difference between the reblurring modules. For fair comparisons, we trained both the network with convolution based reblur model and the network with our physically correct reblur model under the same settings in an unsupervised fashion. The network with convolution based image formation model yields a PSNR metric of 27.22dB and a SSIM metric of 0.8 on our synthetic dataset, while ours yields PSNR and SSIM metrics as 32.24dB and 0.91 respectively.

The necessity to do self-supervised motion deblurring: In real scenarios, motion blur usually occurs in bad illumination conditions. In these cases, it impedes the acquisition with low shutter times to obtain sharp images for supervised learning. One way to address this problem is to train a network with datasets collected under good illumination conditions and transfer the model to scenarios, where motion blur would occur. However, the generalization ability is still questionable due to the large difference between the image textures for both scenarios. We thus evaluate the generalization performance quantitatively and qualitatively with both our synthetic dataset and real dataset respectively. Note that it is not easy to obtain ground truth sharp images in real scenarios. We apply the pretrained networks on our test data directly, to evaluate the generalization performance. Table II and Fig. 5 present the

	Method	PSNR↑	SSIM↑	Time ↓
Opt.-based	Xu et al. [39]	26.04 dB	0.78	377.8 s
Supervised -retrained	DeepDeblur [21]	33.55 dB	0.92	3.45 s
	DeblurGAN [17]	33.23 dB	0.91	0.06 s
	SRN-Deblur [32]	34.64 dB	0.93	0.13 s
Supervised -pretrained	DeepDeblur [21]	29.91 dB	0.87	3.45 s
	DeblurGAN [17]	28.70 dB	0.88	0.06 s
	SRN-Deblur [32]	30.71 dB	0.88	0.13 s
self-supervised	Madam et al. [19]	21.69 dB	0.75	0.25 s
	Ours	32.24 dB	0.91	0.05 s

TABLE II: **Single image deblurring on synthetic dataset.** Supervised-retrained denotes we retrained the networks with our training data. Supervised-pretrained denotes we use the official pretrained models to evaluate on our test data directly.

experimental results. It demonstrates that all the baseline networks have limited generalization ability and perform worse than our method with a large margin. It proves that self-supervision is beneficial for the network to adapt to scenarios, where the ground truth data is difficult to obtain.

Quantitative and qualitative evaluations on synthetic dataset: Table II and Fig. 4 show quantitative and qualitative comparisons on the synthetic dataset. For the qualitative results, we only compare against the best supervised method [32]. As can be seen in Table II, our method outperforms both [39] and [19] significantly in terms of PSNR and SSIM. For optimization-based single image deblurring algorithms (e.g., [39]), they usually assume the motion blur is caused by either camera rotation or in-plane translation. However, this assumption is violated in our setting for a self-driving scenario. Thus, [39] leads to poor performance on our dataset. [19] is designed for simple domain-specific blurry images, such as text and facial images. Therefore, it struggles on our dataset that exhibits complex real-world challenges which are harder to learn. In comparison to supervised methods, our method demonstrates competitive results in our quantitative and qualitative evaluation. As expected, there is still a gap between our method and the supervised methods if the ground truth sharp images are available. However, our method outperforms them with a large margin if they are pretrained on other datasets. It demonstrates that self-supervision enables the network to generalize better to real scenarios, where the ground truth data is usually difficult to obtain. It also demonstrates that our method is amongst the fastest methods and can run in real time on a single GTX1080Ti Graphic card.

Qualitative evaluations on real dataset: Since we do not have ground truth sharp images in our real dataset, we cannot refine the baseline networks on it. We thus use the official pretrained networks for the experiments. Fig. 5 demonstrates that our method can successfully deblur the blurry images, while the pretrained network from Tao et al. [32] results in images with artifacts. More experimental results can be found from our supplementary material at https://github.com/ethliup/SelfDeblur.

Fig. 4: **Qualitative comparisons on synthetic dataset. First**: Ground truth sharp image. **Second**: Input blurry image. **Third**: Deblurring results of the supervised method from Tao et al. [32]; The network is retrained on our dataset. **Fourth**: Deblurring results of the proposed self-supervised learning method.

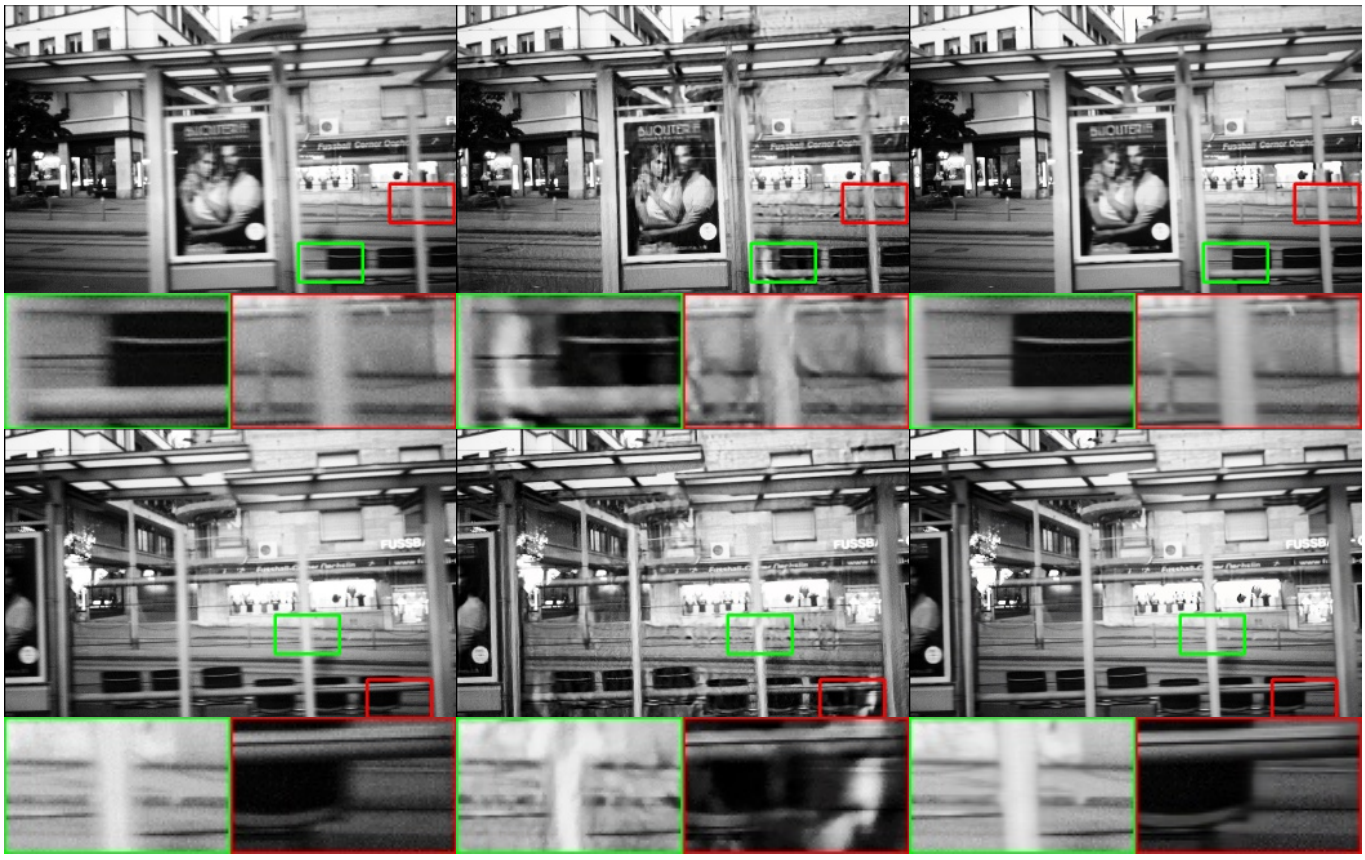

Fig. 5: **Qualitative evaluations on real dataset. First** Blurry image. **Second** Deblurred image by the official pretrained network from Tao et al. [32]. **Third** Deblurred image by our method. The images are post processed for better visualization. Best viewed in digital version.

V. Conclusion and Future Work

In this paper, we have presented a self-supervised learning algorithm for image deblurring. Instead of using ground truth sharp images, we leverage the geometric constraints between two consecutive blurry images to supervise training of our network. Both the latent sharp image and motion blur kernel are estimated by a deblur network and an optical flow estimation nework, respectively. Experimental results show that the proposed algorithm outperforms the previously self-supervised method and can produce competitive results compared to supervised methods. It further demonstrates that our method can be trained with real motion blurry data and generalizes well to real unseen data.

References

[1] H. Chen, J. Gu, O. Gallo, M. Liu, and J. Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *International Conference on Computational Photography (ICCP)*, 2018. 3, 5

[2] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics (SIGGRAPH ASIA 2009)*, 28(5), 2009. 1, 2

[3] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 1

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM Trans. on Graphics*, 2006. 1, 2

[5] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[6] I.-J. Goodfellow, J. PougetAbadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2

[7] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *European Conference on Computer Vision (ECCV)*, 2010. 1, 2

[8] M. Hradis, J. Kotera, P. Zemcik, and F. Sroubek. Convolutional neural networks for direct text deblurring. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2015. 2

[9] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[10] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 6

[11] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 4

[12] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1, 4

[13] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[14] T. H. Kim and K. M. Lee. Generalized video deblurring for dynamic scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[15] T. H. Kim, K. M. Lee, B. Scholkopt, and M. Hirsch. Online video deblurring via dynamic temporal blending network. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[16] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Neural Information Processing Systems Conference (NIPS)*, 2009. 1, 2

[17] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2018. 1, 2, 3, 5, 6

[18] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithm. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 2

[19] N. T. Madam, S. Kumar, and R. A.N. Unsupervised class-specific deblurring. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2, 5, 6

[20] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2018. 1, 4

[21] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 5, 6

[22] H. Park and K. M. Lee. Joint estimation of camera pose, depth, deblurring and super-resolution from a blurred image sequence. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017. 5

[24] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[25] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello. A visual odometry framework robust to motion blur. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2009. 1

[26] J. Qiu, X. Wang, S. J. Maybank, and D. Tao. World from blur. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[27] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the optical society of America*, 62(1), 1972. 2

[28] Q. Shan, J. Jia, and A. A. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3), 2008. 1, 2, 5

[29] S. Su, M. Delbracio, and J. Wang. Deep video deblurring for hand-held cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[30] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[31] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(8):1603–1618, Aug. 2011. 2

[32] X. Tao, H. Gao, Y. Wang, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2018. 1, 2, 3, 5, 6, 7

[33] S. Tourani, M. Sudhanshu, A. Nagariya, V. Chari, and M. Krishna. Rolling shutter and motion blur removal for depth cameras. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2016. 1

[34] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4, 5

[35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[36] P. Wieschollek, M. Hirsch, and B. Scholkopt. Learning blind motion deblurring. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[37] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision (ECCV)*, 2010. 1, 2

[38] L. Xu, J.-S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2

[39] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 5, 6

[40] J. Zhang, J. Pan, J. Ren, and Y. Song. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[41] Y. Zhang, S. Khamis, C. Rhemann, J. P. C. Valentin, A. Kowdle, V. Tankovich, M. Schoenberg, S. Izadi, T. A. Funkhouser, and S. R. Fanello. Activestereonet: End-to-end self-supervised learning for active stereo systems. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1

[42] S. Zheng, L. Xu, and J. Jia. Forward motion deblurring. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013. 2

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

Supplemental Materials: Self-Supervised Linear Motion Deblurring

I. Introduction

In this supplementary material, we present details on the relationship between $\mathbf{u}_{a\to b}/\mathbf{u}_{b\to a}$ (i.e., the optical flow between the latent sharp images) and $\mathbf{u}$ (i.e., the optical flow between the central virtual frame and the first virtual frame), described in Section III.B and III.C of the main paper, the architecture of the deblurring network, as well as additional qualitative experimental results on single image deblurring.

II. Relationship between $\mathbf{u}_{a\to b}/\mathbf{u}_{b\to a}$ and $\mathbf{u}$

In this section, we present the relationship between $\mathbf{u}_{a\to b}/\mathbf{u}_{b\to a}$ and $\mathbf{u}$. $\mathbf{u}_{a\to b}$ and $\mathbf{u}_{b\to a}$ are the bidirectional dense optical flows between the latent sharp images $\mathbf{I}_a$ and $\mathbf{I}_b$, respectively. We assume the motion between $\mathbf{I}_a$ and $\mathbf{I}_b$ to be linear. Without loss of generality, we assume $\mathbf{u}$ is used to synthesize the first blurry image $\mathbf{B}_a$. Thus, we obtain $\mathbf{u}$ as the flow from the central virtual frame $\mathbf{I}_0$ to the first virtual image $\mathbf{I}_1$ by linearly scaling $\mathbf{u}_{a\to b}$ according to

$$\mathbf{u} \approx \frac{\tau_a}{2\mathrm{N}\Delta t}\mathbf{u}_{a\to b} \ , \tag{1}$$

where τ_a is the exposure time of $\mathbf{B}_a$, $2\mathrm{N}+1$ is the number of sampled virtual sharp frames to synthesize $\mathbf{B}_a$, and Δt is the time interval between $\mathbf{I}_a$ and $\mathbf{I}_b$. Similarly, if $\mathbf{u}$ is used to synthesize the second blurry image $\mathbf{B}_b$, we get

$$\mathbf{u} \approx \frac{\tau_b}{2\mathrm{N}\Delta t}\mathbf{u}_{b\to a} \ , \tag{2}$$

where τ_b is the exposure time of $\mathbf{B}_b$.

III. Architecture of the Deblurring Network

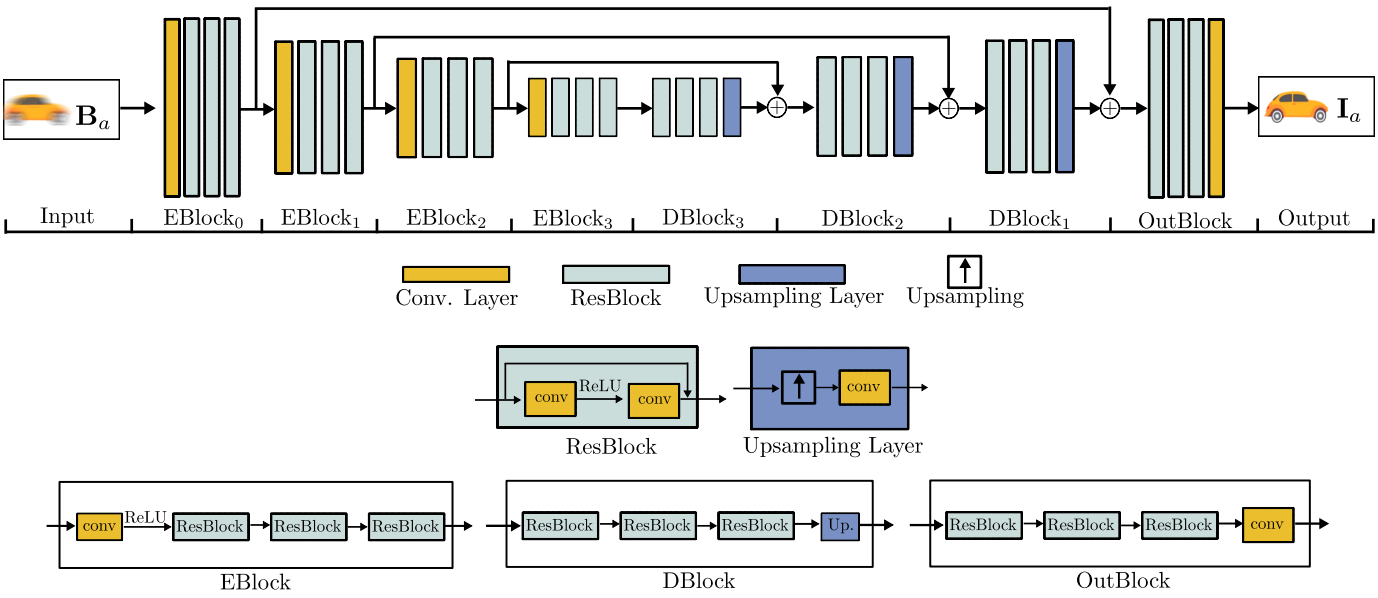

Fig. 1: **Architecture of the deblurring network.** Given the input blurry image $\mathbf{B}_a$, the deblurring network outputs the deblurred latent sharp image $\mathbf{I}_a$. Best viewed in enlarged digital version.

We adapt the single image deblurring network from Tao et al. [3] for our approach. We make the following modifications for our particular problem: 1) We replace the deconvolution layer with bilinear upsampling followed by a 3x3 convolution to avoid upsampling artifacts. 2) We train the network at a single scale without using the LSTM layer to improve both the training and test efficiency. 3) We add one more Encoder-Decoder block to increase the capacity of the network. The details are shown in Fig. 1.

IV. Additional Qualitative Experimental Results

In Fig. 2 to Fig. 8, we present additional qualitative experimental results on single image deblurring on the synthetic dataset. The results demonstrate that our method can generate visually compelling sharp images that are competitive to three state-of-the-art supervised methods [2]–[4]. For fair comparisons, we retrain all the networks on our Fastec dataset. It also significantly outperforms the state-of-the-art optimization-based method from Xu et al. [1] and the self-supervised method from [5]. The optimization based method from Xu et al. [1] fails to deblur blurry images from this dataset. To make the problem tractable, they assume that the motion blur is caused by either camera rotation or in-plane translation. However, those assumptions are violated in our Fastec dataset, where the motion blur is also caused by the 3D scene geometry.

To further demonstrate the temporal consistency of our method, we also present the experimental results for image sequences from Fig. 9 to Fig. 14. The experimental results demonstrate that our network can deblur an image sequence temporally consistent, on both the synthetic and real datasets.

References

[1] L. Xu, S. Zheng and J. Jia, Unnatural L0 sparse representation for natural image deblurring, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[2] S. Nah, T. Kim and K. Lee, Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[3] X. Tao, H. Gao, Y. Wang, X. Shen, J. Wang and J. Jia, Scale-recurrent network for deep image deblurring, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[4] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin and J. Matas, DeblurGAN: Blind motion deblurring using conditional adversarial networks, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[5] N. Madam, S. Kumar and A.N. Rajagopalan, Unsupervised Class-Specific Deblurring, In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

2

Fig. 2: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

Fig. 3: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

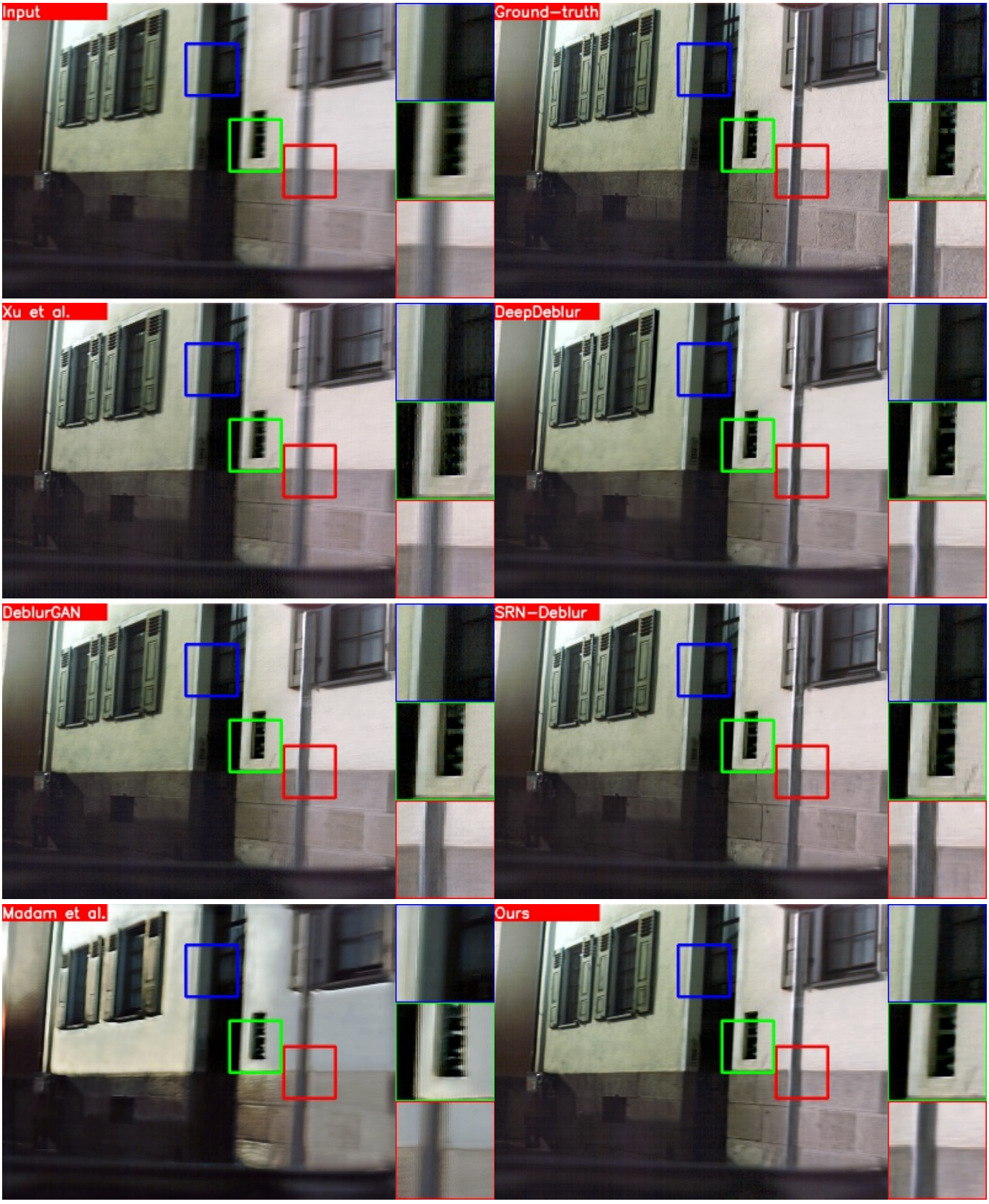

Fig. 4: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

Fig. 5: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

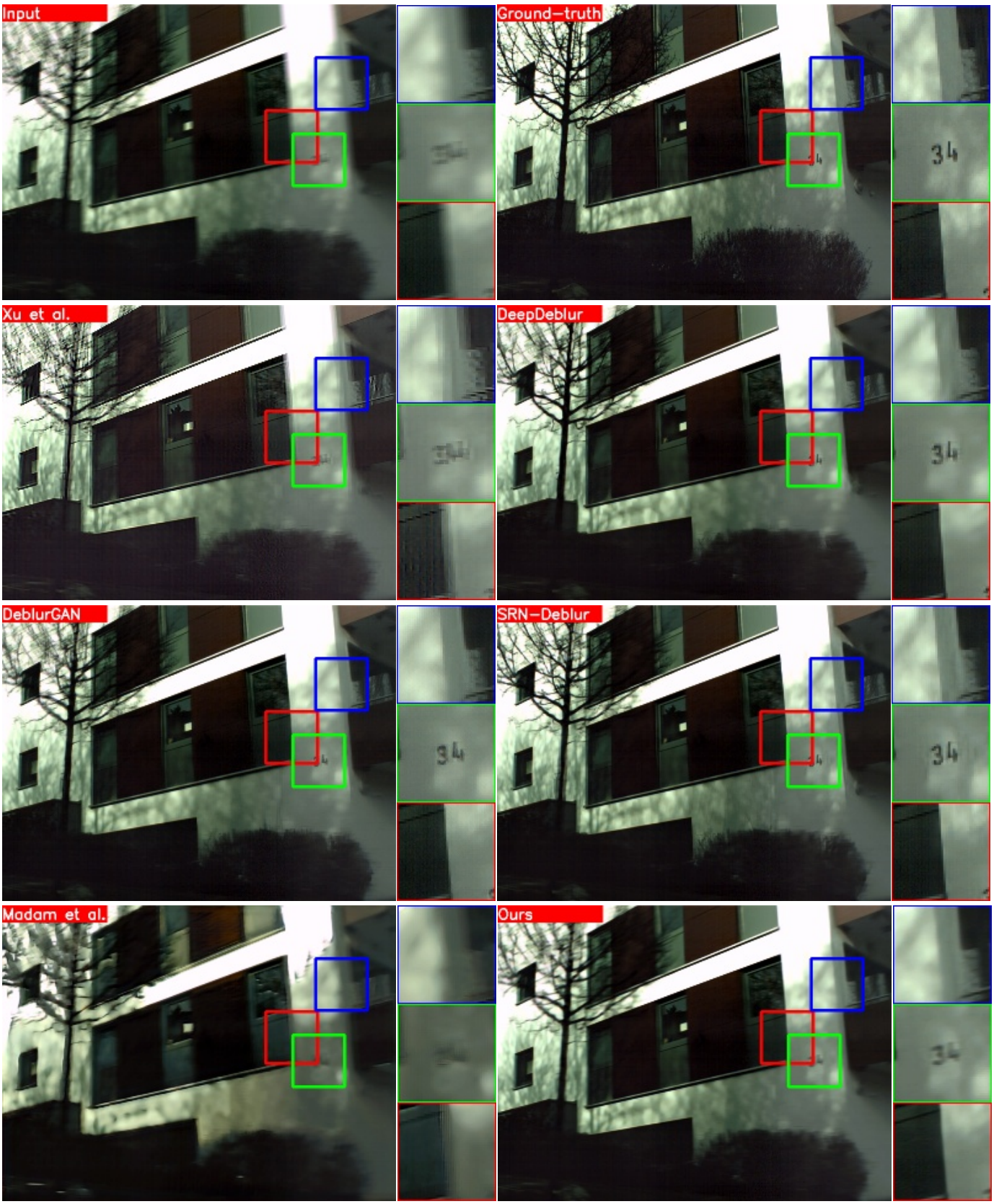

Fig. 6: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

Fig. 7: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

Fig. 8: **Qualitative comparisons on the Fastec dataset.** All the baseline networks are retrained on our Fastec dataset.

Fig. 9: **Temporal consistency on the Fastec dataset (frame 1-5). Left:** Ground truth. **Middle:** Blurry image. **Right:** Deblurred image by our network.

Fig. 10: **Temporal consistency on the Fastec dataset (frame 6-10). Left:** Ground truth. **Middle:** Blurry image. **Right:** Deblurred image by our network.

Fig. 11: **Temporal consistency on the Fastec dataset (frame 11-15). Left:** Ground truth. **Middle:** Blurry image. **Right:** Deblurred image by our network.

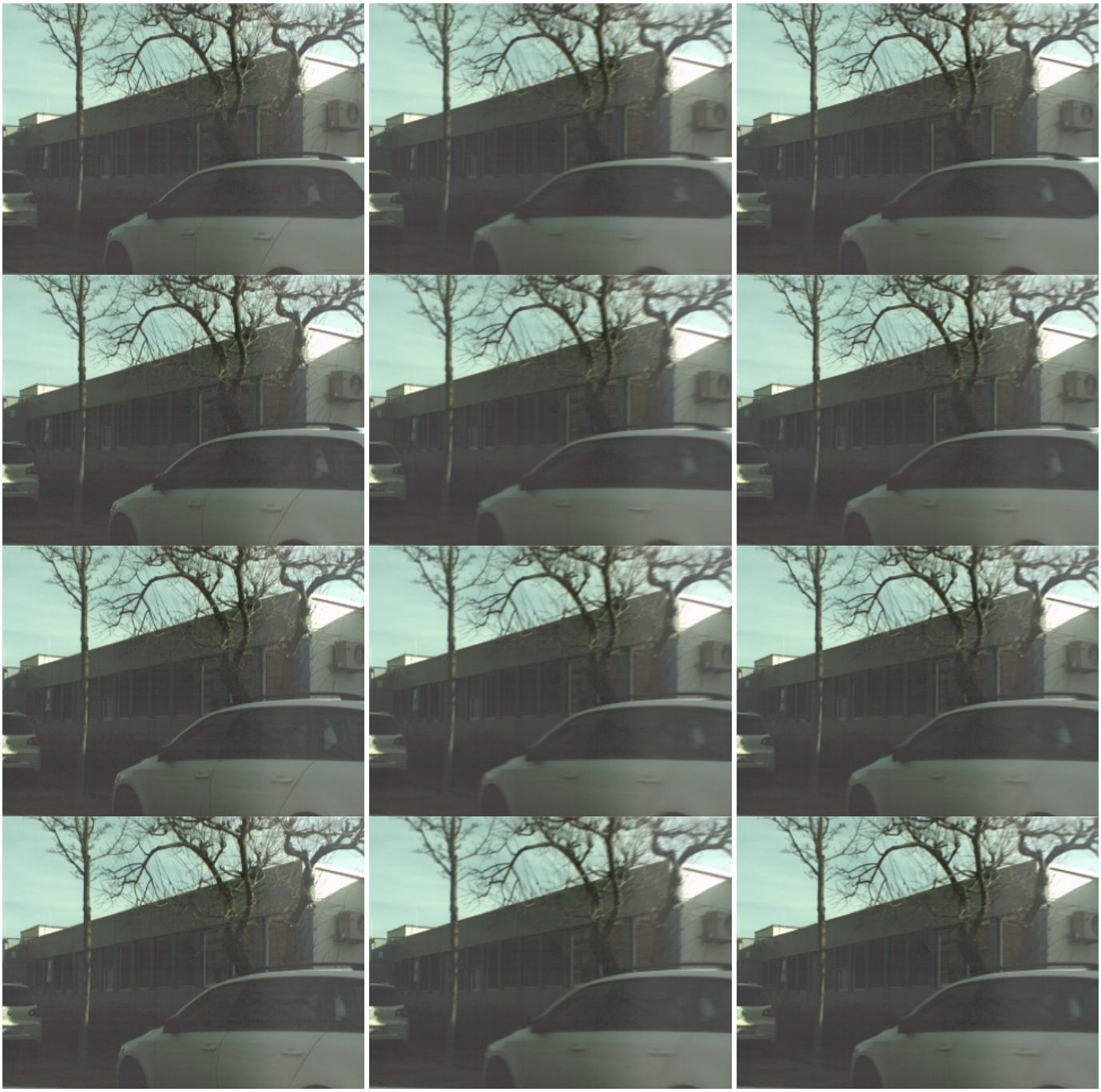

Fig. 12: **Temporal consistency on the Fastec dataset (frame 16-19). Left:** Ground truth. **Middle:** Blurry image. **Right:** Deblurred image by our network.

Fig. 13: **Temporal consistency on the real dataset. Odd rows:** Blurry image. **Even rows:** Deblurred image by our network.

Fig. 14: **Temporal consistency on the real dataset. Odd rows:** Blurry image. **Even rows:** Deblurred image by our network.