World from Blur

Jiayan Qiu[1] Xinchao Wang[2] Stephen J. Maybank[3] Dacheng Tao[1]
[1] UBTECH Sydney AI centre, School of Compurter Science, FEIT, University of Sydney, Australia
[2] Department of Computer Science, Stevens Institute of Technology, USA
[3] Department of Computer Science and Information Systems, Birkbeck College, University of London, UK
jqiu3225@uni.sydney.edu.au xinchao.w@gmail.com sjmaybank@dcs.bbk.ac.uk dacheng.tao@sydney.edu.au

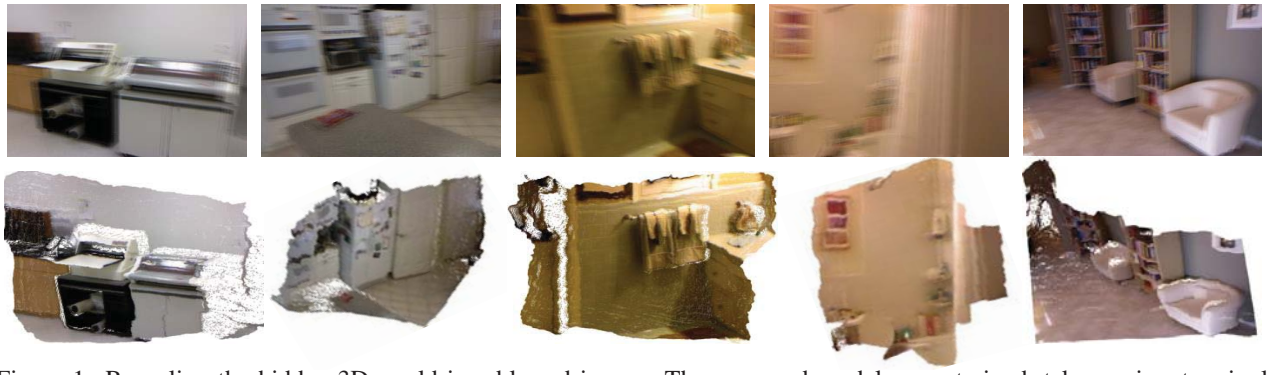


Figure 1: Revealing the hidden 3D world in a blurred image. The proposed model, once trained, takes as input a single blurred image and produces the reconstructed 3D scene concealed in the blurs.

Abstract

What can we tell from a single motion-blurred image? We show in this paper that a 3D scene can be revealed. Unlike prior methods that focus on producing a deblurred image, we propose to estimate and take advantage of the hidden message of a blurred image, the relative motion trajectory, to restore the 3D scene collapsed during the exposure process. To this end, we train a deep network that jointly predicts the motion trajectory, the deblurred image, and the depth one, all of which in turn form a collaborative and self-supervised cycle that supervise one another to reproduce the input blurred image, enabling plausible 3D scene reconstruction from a single blurred image. We test the proposed model on several large-scale datasets we constructed based on benchmarks, as well as real-world blurred images, and show that it yields very encouraging quantitative and qualitative results.

1. Introduction

Motion blur is caused by the relative motion between the scene objects and the camera during the exposure process. When the motion of the scene objects, or the camera, or both, is significant during the exposure time, the image tends to appear smeared along the direction of the relative motion. Motion-blurred images are in many cases favored by photographers and artists for aesthetic purpose, but seldom by computer vision researchers, as many standard vision tools including detectors, trackers, and feature extractors have a hard time dealing with the blurs.

Much effort has thus been made in the image processing and computer vision community to remove the "negative" influences of the blurs. A straightforward and crude way is to ignore blurred images, as done in SLAM systems [52] because matching algorithms tend to fail on blurred images. Another more analytical way is to conduct *deblurring*, which recovers a deblurred image from a blurred one. Over the past decades, there has been a series of seminal work along this line, demonstrating very promising and visually-pleasing results.

Despite the excellent results achieved, deblurring methods limit its goal to producing a blur-free image and omit the physical rationale behind the blurs. Since a blurred image is the result of relative motions, it actually encodes the motion information, though in a degraded way. The work of [35] pioneered to extract a sequence of deblurred images from a blurred one, yet still overlooked the motions con-

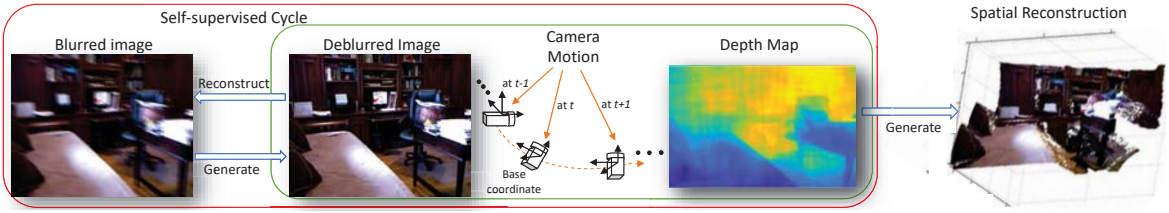


Figure 2: Illustration of our model. Given a blurred image, we construct a network with three modules to estimate camera motions, the deblurred image, and the depth map, all of which form a self-supervised cycle to reconstruct the input blurred image and further enable the 3D scene reconstruction from the blurs.

cealed.

We show in this paper that more hidden message can be revealed in a blurred image. As motion is encoded in the blurs, we propose to explicitly estimate the concealed motion trajectories buried under the smearings, based on which the static 3D scene can be restored, as demonstrated in Fig. 1. To this end, we train a collaborative network, which jointly infers motion trajectory and depth, both of which in absolute scale, as well as deblurred frame instant. All the three estimations, in turn, form a self-supervised cycle to reproduce the original blurred image, in aim to imitate the physical blurring process. Through this cycle, the different modules supervise and enhance one another, enabling plausible 3D reconstruction, as shown in Fig. 2.

Unarguably, estimating motion trajectories from a single blurred image is an inverse problem. To recover the most legitimate motion process while preserving a reasonable computational load, we approximate a blurred image, for which the creation process is continuous, as an average of a sequence of frames. In this regard, we construct datasets upon popular benchmarks, wherein each blurred image is generated by taking the average of a clean-frame sequence induced by deterministic motion. The constructed datasets thus provide us with ground truths for training the collaborative network, and allow us to conduct depth-, motion-, and frame-estimation, as well as the consequent 3D reconstruction. The proposed model, once trained, yields very promising results on synthetic and real-world blurred images.

Our contribution is therefore a novel approach that, for the first time, attempts to recover the absolute-scale 3D scene from a single blurred image. It is accomplished by training an innovative collaborative network that simultaneously estimates depth, clean images, and motion trajectories, each of which supervises another via a self-consistent cycle to reproduce the input blurred image, on large-scale datasets we build upon popular benchmarks. The proposed approach produces encouraging results on synthetic and real-world blurred images. Our code, model and datasets will be released.

2. Related work

There have been numerous reconstruction methods aiming to recover the 3D scene from one or multiple images, including but not limited to reconstruction from shading [95], from image texture [7, 8, 28], from camera motion [9], from stereo [49], from scene recognition [48, 22, 26], from tracking process [90, 50, 84, 83] and from focus [56].

Our approach, however, focuses on estimating 3D reconstruction from a single blurred image, not relies on the tracking process, which to our best knowledge is the first attempt along this line. As our cyclic strategy involves three modules, camera trajectory estimation, deblurring, and depth estimation, in what follows, we briefly review related work on these topics.

Camera trajectory estimation. Recent camera-trajectory estimation models can be broadly divided into three categories, based on the supervision level. The first category is fully-supervised methods. For example, Agrwal *et al*. [1] learn good visual features from moving cameras and predict the camera motion from a sequence of images. Wang *et al*. [80, 81] implement a recurrent ConvNet architecture for visual odometry estimation. Ummenhofer *et al*. [77] design an architecture to learn the depth and motion information from stereo images. The second category is weakly-supervised models. Examples include the approach of [34], which estimates the inter-frame motion by utilizing the stereo geometry known a priori. Approaches in the third category are unsupervised. For example, Vijayanarasimhan *et al*. [78] and Zhou *et al*. [97] propose unsupervised methods to estimate the camera ego-motion using the photometric error. The ones of [17, 18, 92] use stereo information to estimate the odometry from a sequence of images. Existing methods, however, conduct motion estimation from clean images, which differs from our focus on blurred images.

Deblurring. Blind deconvolution methods [62, 51, 4, 10, 88, 27, 93, 2, 19] for image deblurring have been widely studied and achieved promising results. . Recently, the models of [31, 32, 76, 61, 20, 60, 57, 53, 58] are designed to handle images with more than single-motion blurs. Another line of work focuses on video deblurring. For example, Zhang *et al*. [94] propose a method that jointly estimates the motions between consecutive frames, while Sellent *et al*. [71] instead utilize stereo information. Wieschollek *et al*. [86] introduce a recurrent ConvNet to deblur an image by using temporal information. Kim *et al*. [38] propose a method to simultaneously conduct deblurring and estimate

the optical flow between consecutive images. Ren *et al*. [66] exploit semantic information to guide the deblurring and optical flow estimation. Su *et al*. [75] propose a ConvNet for deblurring by utilizing inter-frame information. Pan *et al*. [63] jointly estimate the scene flow and deblur the image.

There are some approaches for estimating the spatial information from blurs, but they all focus on image sequences instead of a single image. For example, Park *et al*. [64] develop a method for the joint estimation of camera pose, depth, deblurring, and super-resolution from a sequence of blurred images. More recently, Jin *et al*. [35] propose a framework to extract a video sequence from a single blurred image, yet overlook the spatial information that enables 3D reconstruction.

Depth estimation. Earlier methods for depth estimation rely on geometry-based algorithms from stereo pairs [70, 14, 13]. Saxena *et al*. [68] first propose to exploit the monocular cues to estimate the scene depth, based on which many methods are proposed, yielding encouraging results [69, 29, 42, 45, 6, 39, 3, 73, 65, 16, 24, 91]. The methods of [98, 47, 36, 59, 89, 85], on the other hand, exploit not only local but also global image cues. Given the success of ConvNet in image processing, many deep learning based methods have been proposed [21, 96, 44, 54, 72, 82, 67, 46, 37, 11]. Thanks to multi-level contextual and structural information derived from deep networks, such as AlexNet [40], VGG [74], and ResNet [25], depth estimation has been boosted to a high-accuracy level [12, 17, 41, 43, 87, 79, 15]. Although these methods work well on single image depth estimation, they are not designed for estimating depth from a blurred image, which is the focus of our approach.

3. Preliminaries

Before introducing our model, we briefly review some preliminaries including the creation of a blurred image and the fundamental of 3D geometry, upon which we build our network and the self-supervised cycle.

Blurring Process. The process of image blurring is continuous within the exposure time t of the camera:

$$B = \frac{1}{t}\int^{t} I(t)\, dt, \tag{1}$$

where B is the resulting blurred image, t is the exposure time, and $I(t)$ is the clean image of the scene at time t. To model the blurring process in a computationally tractable way, we approximate this continuous process using the average of a sequence of $2n+1$ frames in the exposure process. We take the very middle frame, the $n+1$-frame, as the *reference frame*, and compute the relative motions at other frames with respect to this frame, as discussed in Sec. 4.

Vision geometry. Let p denote the 2D homogeneous coordinate of a pixel in image I, and P denote the corresponding 3D homogeneous coordinate in I's coordinate system. Also, let D denote the depth map of an image I, with $D(p)$ being the absolute distance between the camera's focal point and the real-world point P, whose projection on I is p. Finally, let T denote the transformation matrix that describes the absolute-scale motion of the camera, governed by six parameters, three for translation and three for rotation. For a pixel p, the corresponding 2D coordinate p' after the transformation T is computed as

$$p' = KTD(p)K^{-1}p, \tag{2}$$

where the intrinsic parameter matrix K of the camera is assumed known, as done in [92, 97]. In this process, the pixel p of the original image I is first inversely projected back to the 3D space, and then the obtained 3D point is transferred to a new 3D location according to the transformation matrix T. Finally, the new 3D point is re-projected to the new 2D scene by applying K to the coordinates of the 3D points.

4. Method

In this section, we introduce the proposed approach to recovering 3D scene from a single blurred image. We first give an overview of our approach, then discuss the modules of our network, and finally, show the self-supervised strategy to jointly optimize all the modules.

4.1. Overview

Our model comprises three modules for motion-estimation, deblurring, and depth-estimation, as well as an innovative self-supervised scheme that optimizes all modules together. The self-supervision is achieved by forming a cycle of three modules, all of which collaborate with each other, in aim to together reproduce the input blurred image. In other words, the input blurred image itself is utilized as a supervision signal for computing the reconstruction loss, during which process all the modules interact with and enhance each other.

We focus on static scene reconstruction and assume the relative motion is caused by the camera movement. We thus aim to estimate a static frame instant or reference frame, as well as a sequence of relative camera motions with respect to the reference that gives rise to the blurs. In our implementation, we take the frame instant in the very middle of the sequence as the reference frame, as discussed in Sec. 3.

We follow a two-stage training strategy, which we find to be more efficient and effective than the single-stage strategy that trains modules with the cycle all at once. In the first stage, we train the three modules independently, all in a supervised manner. In the second stage, we stack the three modules to form a self-supervised cycle, for which the goal is, again, to allow the predictions to reproduce the original blurred image so that the different modules can supervise and benefit one another. It is noteworthy that in the second stage, we provide ground truths for only the motion-estimation module but not the other two, in order to avoid

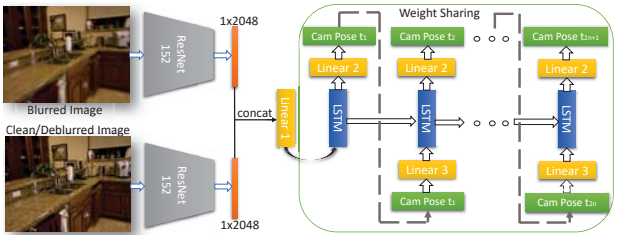


Figure 3: The architecture of our LSTM module for motion estimation. It takes as input a pair of images. The upper branch receives a blurred image, while the lower receives a clean or deblurred image, taken to be the reference frame. It outputs a sequence of $2n$ motions with respect to the reference. Notably, the lower branch is fed with clean images in the first training stage, and with deblurred images from the deblurring module in the cyclic self-supervision stage.

overfitting. Our experiments demonstrate that, compared to the single-stage strategy, the two-stage training converges much faster.

In what follows, we give more details on the three modules and the self-supervision strategy. To highlight the feasibility of inferring the static 3D scene from a blurred image, we mainly rely on compact networks to handle depth-, deblur-, and motion-estimation tasks. More sophisticated end-to-end networks can be readily applied as well and are likely to yield even better performances.

4.2. Motion-estimation Module

Our motion-estimation module, as depicted in Fig. 3, takes as input a blurred image, as well as a clean image or a deblurred one estimated by the deblurring module described in Sec. 4.3. It outputs a sequence of $2n+1$ relative camera motions with respect to the reference frame. This network architecture is motivated by the recent success of image captioning [5], whose goal is to produce a sequence of words describing an input image. The major difference is that our network is fed with a pair of images instead of one.

Specifically, we employ a ResNet152 [25] to extract the features from the second last fully-connected layer for both input images, and then concatenate the obtained features into one, which is fed as input to a Long-Short Term Memory (LSTM) network comprising $2n$ LSTM blocks with shared parameters. The LSTM network is expected to learn the temporal coherence of the camera motion and to output a sequence of $2n$ camera poses with respect to the reference frame. To unify the size of the feature vectors fed to the LSTM blocks, we introduce a *linear3* layer, which is implemented for the second to the last frame instants but not for the first one.

Recall that camera motion, described with the transformation matrix T, is characterized by a rotation vector $\mathbf{u} \in \mathbb{R}^3$ and a translation vector $\mathbf{v} \in \mathbb{R}^3$, where the former

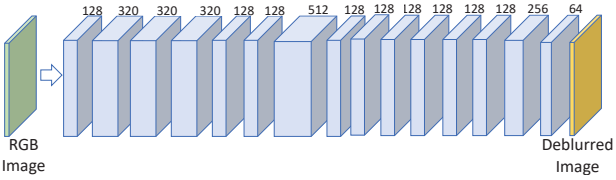


Figure 4: The architecture of our deblurring module.

one depicts the Pitch-, Yaw-, and Roll-rotation and the latter represents the translation along the X-, Y-, and Z-axis. Through out experiments, we find out that learning $\mathbf{u}$ and $\mathbf{v}$ separately leads to more favorable results. The losses of our LSTM for learning the two three-dimension variables are taken to be

$$\mathcal{L}_{\mathbf{u}} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{u}^i - \hat{\mathbf{u}}^i\|^2, \quad \mathcal{L}_{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{v}^i - \hat{\mathbf{v}}^i\|^2, \tag{3}$$

where $\mathbf{u}^i$, $\mathbf{v}^i$ are the ground truths of the i-th motion, $\hat{\mathbf{u}}^i$, $\hat{\mathbf{v}}^i$ are their estimations, and N is the number of samples.

4.3. Deblurring Module

The deblurring module takes the blurred image as input and produces a deblurred image, which we take to be the reference frame. In our implementation, we adopt the CNN-L15 model [30] that shows state-of-the-art performance yet comes in a compact size, with some minor modifications. The rough network structure shows in Fig. 4. We add batch normalization [33] on each layer except the last layer and change the active function of last layer from ReLU to be Tanh. The loss for deblurring is taken to be pixel-level square loss between the deblurred image and the ground truth:

$$\mathcal{L}_b = \frac{1}{N}\sum_{i=1}^{N}\|I^i - \hat{I}^i\|^2, \tag{4}$$

where I^i and $\hat{I}^i$ represent the i-th ground truth and the deblurred image respectively, and N denotes the number of samples.

4.4. Depth-estimation Module

The case for the depth-estimation module is slightly more complicated than the other two, as it has to handle heterogeneous inputs in the two stages of training. Recall that in the first stage we train the three modules separately all in the supervised way, yet in the second stage, as to be discussed in Sec. 4.5, we provide supervision signal only to the motion estimation and allow the cycle to enhance the depth and deblurring module. In other words, in the first stage the depth-estimation module is fed with clean images as input to produce depth, but in the second it is provided with *deblurred* images, which may still contain smearings.

The depth-estimation module is, therefore, expected to produce reasonable results even when the input images still contain blurs. To this end, we devise a two-branch network

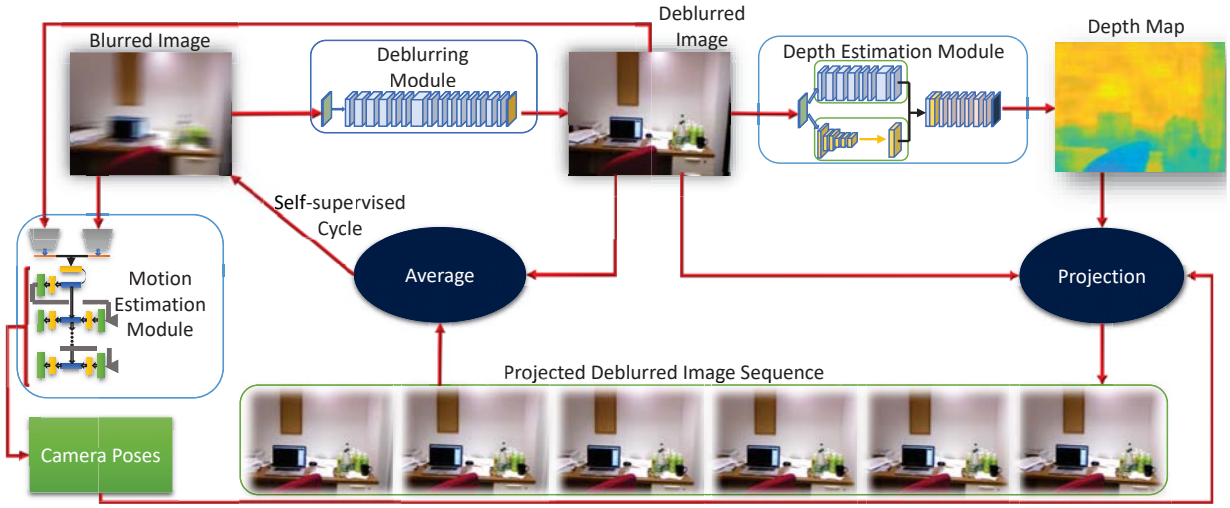


Figure 5: Illustration of the proposed cyclic self-supervision scheme. The predicted camera motions, deblurred image, and depth map are utilized to produce a sequence of frame instants, all of which are then averaged to reconstruct the input blurred image for computing the loss. The different modules, via this cycle, supervise and improve one another.

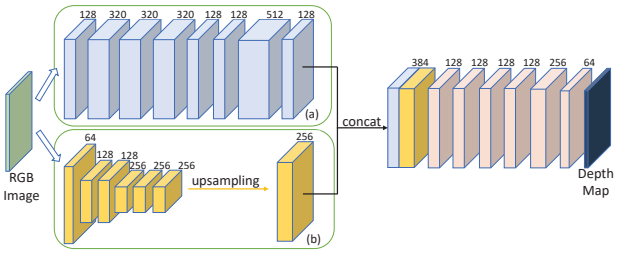


Figure 6: The architecture of our depth-estimation module. Branch (a) is inherited from the deblurring module to jointly handle the blur information and extract local depth clues. Branch (b) is implemented with the first six layers of VGG net to extract global depth clues.

for depth estimation, as shown in Fig. 6. The branch (a) has the same structure as the deblurring module, which simultaneously handles the remaining blur information in the input image and extracts local depth clues. The branch (b), on the other hand, focuses on extracting global depth clues. It is implemented by taking the first six layers from VGG [74], followed by upsampling the features to the same size as those in branch (a). The features from both branches are then concatenated and fed to a network of the same architecture as the deblurring one, with the only difference being that the activation function is ReLU in the last layer. The loss of the depth-estimation module is taken to be

$$\mathcal{L}_d = \frac{1}{N}\sum_{i=1}^{N} \|D^i - \hat{D}^i\|^2, \tag{5}$$

where D^i and $\hat{D}^i$ denote respectively the i-th ground truth and the prediction, and N is the number of samples.

4.5. Self-supervised Scheme

If the predictions of the motion-estimation module, the deblurring module, and the depth-estimation module are plausible, then together they should reconstruct the original blurred image. With this motivation, we stack the three modules in a cycle, for which the goal is to ensure all the predictions, in turn, reproduce the input blurred image. With the cycle, the blurred image itself is treated as the supervision signal, allowing the different modules to collaboratively supervise and benefit one another.

Our design for the cycle is depicted in Fig. 5. Intuitively, given an input blurred image, the deblurring module produces a deblurred image as the reference frame, which is then fed to both the depth module and the motion module. The former module outputs a depth map and the latter generates a motion sequence. Both outputs are, together with the deblurred reference frame, utilized to produce a sequence of clean images, which are further averaged to reproduce the input blurred image and for computing the loss.

Specifically, let p denote the homogeneous coordinate of a pixel in the deblurred reference frame. Given a camera motion $\hat{T}$ estimated by the motion module and depth map $\hat{D}$ estimated by the depth module, the corresponding pixel coordinate p' after undergoing the motion is computed, according to Eq. 2,

$$p' = K\hat{T}\hat{D}(p)K^{-1}p, \tag{6}$$

where again K is assumed to be given as done in previous works [92, 97]. We repeat this process for all the pixels by applying bilinear interpolation and in this way get a complete image, I', that undergoes a motion of $\hat{T}$ with respect to the reference frame I.

As the motion module estimates $2n$ relative motions, we

compute $2n$ such images using Eq. 6, all of which are then averaged to approximate the input blurred image for computing the loss. We write the cyclic-reconstruction loss as

$$\mathcal{L}_r = \frac{1}{N}\sum_{i=1}^{N} \|B^i - \hat{B}_r^i\|^2, \tag{7}$$

where B is the i-th input blurred image, $\hat{B}_r^i = \frac{1}{2n+1}\sum_{k=1}^{2n+1}\hat{B}^{i,k}$, with $\hat{B}^{i,k}$ being the image at k-th frame instant within the sequence, and N is the number of samples.

As discussed, we only utilize the motion supervision in the cycle-tuning stage. We thus have the final loss functions, one for rotation and one for translation, as follows,

$$\tilde{\mathcal{L}}_{\mathbf{u}} = \mathcal{L}_{\mathbf{u}} + \alpha\mathcal{L}_r, \quad \tilde{\mathcal{L}}_{\mathbf{v}} = \mathcal{L}_{\mathbf{v}} + \alpha\mathcal{L}_r, \tag{8}$$

where α is taken to be 10^{-3}.

5. Experiments

In this section, we provide our experimental setups and show the results. Since we are not aware of any existing work that performs exactly the same task as we do here, we mainly focus on showing the promise of the proposed network especially the self-supervised cycle design. We also compare part of our network with other popular models, and then substitute our module with others to verify the value of the cycle by comparing the performance of other models without and with the cycle.

Our goal is, again, to show the possibility of recovering the 3D scene from a blurred image, rather than trying to beat the state-of-the-art deblurring, depth- and trajectory-estimation, and 3D reconstruction models. More complicated networks, as long as they are end-to-end trainable, can be adopted in our cycle with possibly better performances.

5.1. Datasets and Implementation Details.

NYU Depth v2 [55]. It comprises 464 indoor scenes, among which we use 364 scenes for training and 100 for testing. Blurred images are created by averaging 7 consecutive frames. In total, we create 57K blurred samples for training and 13K for testing using about 420K frames. We adopt this dataset for constructing blurred images, because it provides a depth map for each video frame and the frame rate is high with respect to the camera motions. We also tried KITTI but found spatial gaps between two consecutive frames are too large, making the synthetic blurs unrealistic.

ICL-NUIM dataset [23]. It is smaller in size as compared to the NYU one. By following the same procedure as done for NYU, we create 706 blurred samples using 4.9K frames from two scenes for training and 604 samples using 4.2K frames from another two scenes for testing. Due to the limited training samples, we adopt the network pre-trained on NYU and finetune it on this dataset.

Term	Pre-NYU	C-NYU	Pre-ICL	C-ICL
Translation$_x$	3.589	2.584	3.813	2.961
Translation$_y$	3.735	2.746	3.796	3.142
Translation$_z$	2.446	1.492	2.452	2.112
Yaw	0.209	0.110	0.239	0.201
Pitch	0.184	0.084	0.185	0.147
Roll	0.180	0.082	0.206	0.144

Table 1: Results of the motion-estimation module. Translations are measured in centimeters and rotations in degrees. Pre-NYU refers to the network trained using ground-truth clean images on NYU, and C-NYU is the one with the self-supervised cycle, for which the input is the output of the deblur module. Pre-ICL and C-ICL refer to the corresponding networks on the ICL-NUIM dataset.

Term	Pre-NYU	C-NYU	Pre-ICL	C-ICL
PSNR	25.94	27.22	26.43	27.19
SSIM	0.8543	0.8931	0.8895	0.9206

Table 2: Results of deblurring without (Pre-NYU/ICL) and with (C-NYU/ICL) the self-supervised cycle on the two datasets.

Implementation. Our networks are implemented using PyTorch and with two Tesla V-100 SXM2 GPUs. The batch sizes for the motion estimation, deblurring and depth estimation module are 64, 4 and 4, respectively. During the cycle stage, the batch size is set to 2 for all modules due to the memory limitation. As our blurred dataset is trained by averaging 7 images, we train the LSTM model of Sec. 4.2 to predict 6 motions with respective to the reference frame.

5.2. Motion Estimation

Tab. 1 shows the absolute errors of translation (in centimeters), and of rotation angles along three axes (in degrees). It can be seen that with the self-supervised cycle, the errors on translations decrease about 1cm and those on rotations reduce up to 50%. It is noteworthy that the improvements on ICL are smaller than those on NYU, due to limited training samples.

5.3. Deblurring

We show the deblurring results in Tab. 2, where the self-supervised cycle again yields significant improvements. On the NYU dataset, the PNSR get increased by more than 1dB and the SSIM by 0.04. The same trend is observed on ICL, where PSNR improves more than 0.75dB and the SSIM improves more than 0.03. These results indicate that the self-supervised cycle enhances not only the pixel-based appearance of the deblurred images, but also the more global structural patterns, which are crucial for the succeeding depth estimation, motion estimation, and reconstruction tasks.

5.4. Depth Estimation

As shown in Tab. 3, the self-supervised cycle improves the performance of depth estimation by a large margin, in

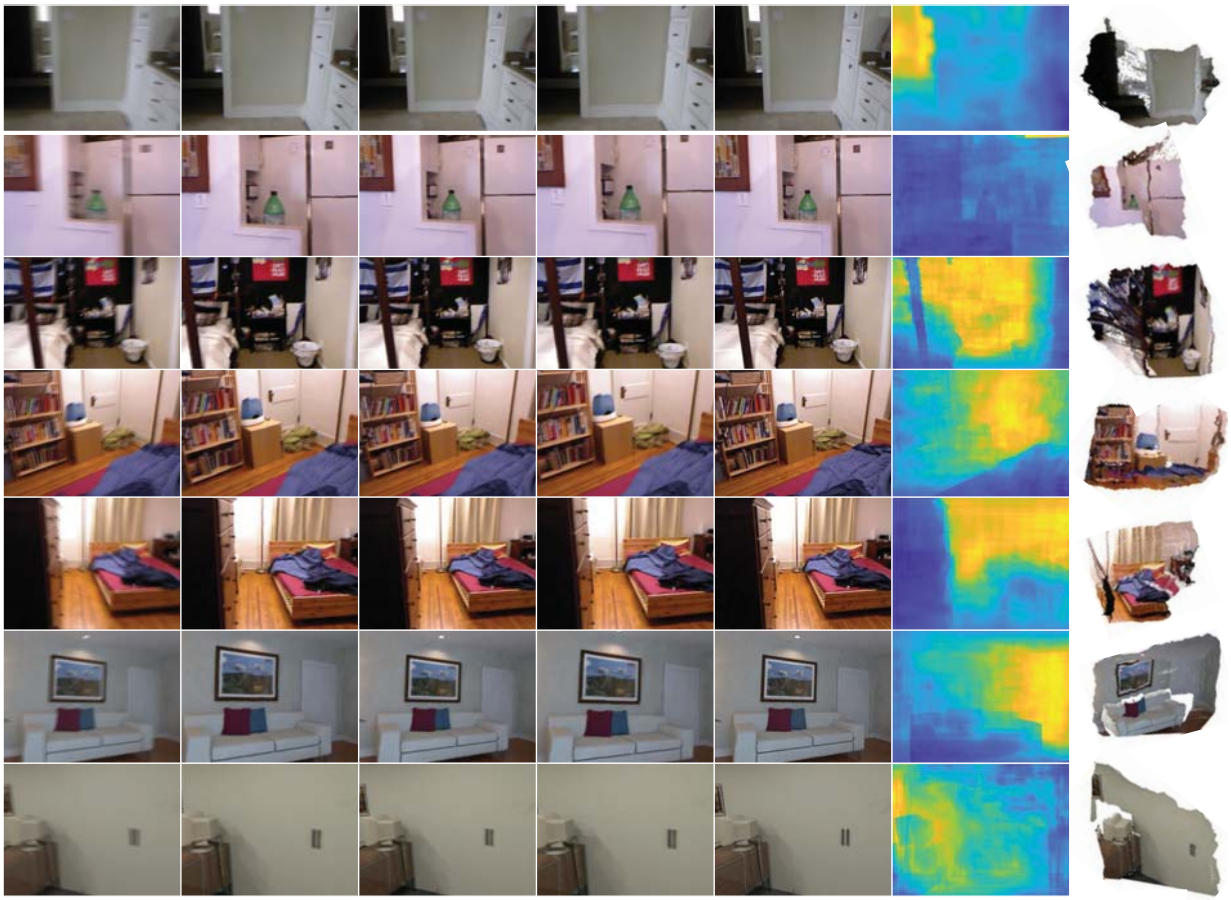


Figure 7: Results on the NYU dataset in the top 5 rows and the ICL dataset in the bottom 2 rows. Column (a) depicts the input blurred images, (c) depicts the deblurred reference frames, (b) and (d) are the first and the last projected frames, (e) corresponds to the ground-truth clean reference frames, (f) displays the predicted depth maps, and (g) demonstrates the 3D reconstruction results.

Term	Pre-NYU	C-NYU	Pre-ICL	C-ICL
Abs Rel	0.217	0.184	0.220	0.206
SqRel	0.213	0.156	0.216	0.180
RMSE	0.911	0.607	0.918	0.661
RMSE log	0.289	0.222	0.293	0.244
$\delta < 1.25$	0.607	0.733	0.603	0.684
$\delta < 1.25^2$	0.884	0.932	0.879	0.918
$\delta < 1.25^3$	0.969	0.982	0.961	0.972

Table 3: Results of the depth estimation without (Pre-NYU/ICL) and with (C-NYU/ICL) the self-supervised cycle on the two datasets.

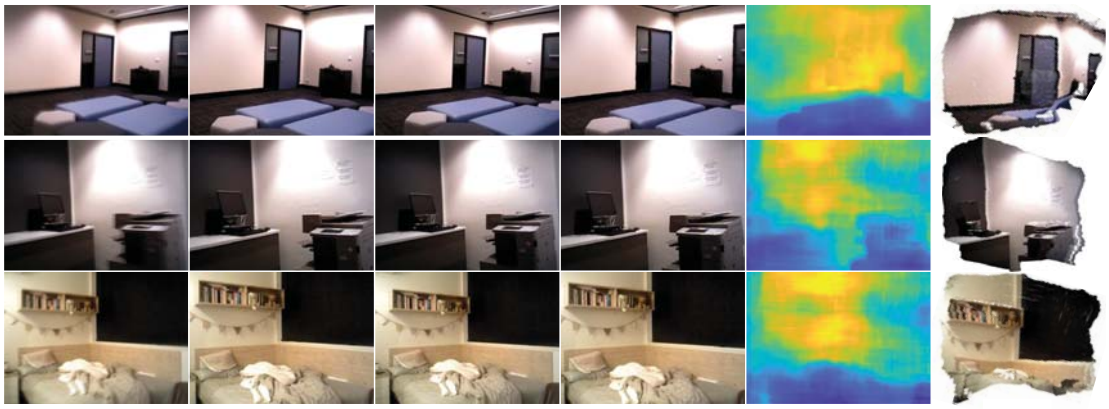
terms of both the error metrics including Abs Rel, SqRel, RMSE and RMSE log, and the accuracy ones $\delta < 1.25^n$. The large improvement on $\delta < 1.25$ shows that our cycle improves depth estimation on a large number of pixels across the image, indicating that the cycle benefits the global depth-estimation performance. Please note that, as discussed in Sec. 4.4, the pre-trained depth module is learned on clean images.

5.5. Analysis

Results on real-world blurred images. We show in Fig. 8 the results of our model on some real-world blurred images, taken by Asus Xtion Pro as camera parameters are close to those of Kinect v2. We show the blurred images on column (a), followed by three recovered clean frames, the ground-truth clean reference frames, depth maps, and 3D reconstructions. The results are visually pleasing despite not perfect.

Comparisons to other models. Here we conduct ablation studies to verify the performance of our depth estimation module and show why it fits our purpose. Specifically, we compare our network with a popular one from Eigen *et al*. [12]. When training Eigen's network, we followed the training strategy provided in [12]. All ablation experiments are conducted on the NYU dataset.

We compare the performances of Eigen's network and ours when trained on the clean image. As shown in Tab. 4, the results of the two models are very similar. When trained



(a) (b) (c) (d) (e) (f)

Figure 8: Results on real-world blurred images. Column (a) shows the blurred images, (c) corresponds to the deblurred reference frames, (b) and (d) are the first and the last projected frames, (e) shows the predicted depth maps, and (f) shows the 3D reconstruction results.

Term	Pre-Eigen's [12] on Clean	Pre-Ours on Clean	Pre-Eigen's on Deblurred	Pre-Ours on Deblurred	C-Eigen's	C-Ours
Abs Rel	0.215	0.217	0.231	0.224	0.198	0.184
SqRel	0.212	0.213	0.244	0.232	0.177	0.156
RMSE	0.907	0.911	0.921	0.917	0.651	0.607
RMSE log	0.285	0.289	0.291	0.290	0.237	0.222
$\delta < 1.25$	0.611	0.607	0.583	0.604	0.696	0.733
$\delta < 1.25^2$	0.887	0.884	0.869	0.880	0.922	0.932
$\delta < 1.25^3$	0.971	0.969	0.964	0.967	0.979	0.982

Table 4: Results of Eigen's depth network and ours under different setups. We compare the performances of the two networks pre-trained on clean images (Pre-Eigen's/Ours on Clean), the performances of the two networks trained using outputs of the deblur network (Pre-Eigen's/Ours on Deblurred), and those of the two using the proposed self-supervised cycle (C-Eigen's/Ours).

Term	With Eigen's depth	With our depth
PSNR	26.13	27.22
SSIM	0.8697	0.8931

Table 5: Comparing deblurring network after self-supervised cycle with ours depth estimation module and Eigen's depth network.

Term	With Eigen's depth	With our depth
Translation$_x$	2.762	2.584
Translation$_y$	3.008	2.746
Translation$_z$	1.699	1.492
Yaw	0.135	0.110
Pitch	0.107	0.084
Roll	0.096	0.082

Table 6: Comparing motion estimation module after self-supervised cycle with ours depth estimation module and Eigen's depth network.

on deblurred images and trained using the cycle, however, our network produces visibly better results, indicating that the proposed depth module with the two-branch architecture depicted in Fig. 6 can better handle blur information.

We further show the results of the deblurring and motion estimation using Eigen's depth network and ours in Tabs. 5 and 6 respectively. From both tables, we see that the proposed model yields superior results thanks to the better depth estimation. The results also indicate the important role that depth plays within the self-supervised cycle.

6. Conclusion

We show in this paper that given a blurred image, one can recover the 3D world hidden under the blurs given the camera intrinsic parameters. We accomplish this via training a deep network of three modules, one for motion estimation, one for deblurring, and one for depth estimation, all of which form a cycle to in turn reproduce the input blurred image and supervise one another. We construct datasets upon several large-scale benchmarks for training our model, and demonstrate the effectiveness of the proposed model on these datasets as well as real-world blurred images. In the future work, we will endeavor to estimate dynamic scenes from single blurred images, and incorporate more tasks like scene parsing into the framework.

Acknowledgement This research was supported by Australian Research Council Projects FL-170100117, DP-180103424, IH-180100002 and the startup funding of Stevens Institute of Technology. Xinchao Wang is the corresponding author of this paper.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 37–45, 2015.

[2] Y. Bahat, N. Efrat, and M. Irani. Non-uniform blind deblurring by reblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3294, 2017.

[3] M. H. Baig and L. Torresani. Coupled depth learning. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016.

[4] A. Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, pages 221–235. Springer, 2016.

[5] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431, 2015.

[6] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn. Depth analogy: Data-driven approach for single image depth estimation using gradient samples. *IEEE Transactions on Image Processing*, 24(12):5953–5966, 2015.

[7] A. Criminisi and A. Zisserman. Shape from texture: Homogeneity revisited. In *BMVC*, volume 1, page 2, 2000.

[8] E. Davis and G. Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[9] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 557–564. IEEE, 2000.

[10] J. Dong, J. Pan, Z. Su, and M.-H. Yang. Blind image deblurring with outlier handling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2478–2486, 2017.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[13] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[14] D. A. Forsyth and J. Ponce. A modern approach. *Computer vision: a modern approach*, pages 88–101, 2003.

[15] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[16] R. Furukawa, R. Sagawa, and H. Kawasaki. Depth estimation using structured light flow–analysis of projected pattern flow on an object's surface–. *arXiv preprint arXiv:1710.00513*, 2017.

[17] R. Garg, G. BGV Kumar, Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[18] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[19] D. Gong, M. Tan, Y. Zhang, A. Van den Hengel, and Q. Shi. Blind image deconvolution by automatic gradient activation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1827–1836, 2016.

[20] D. Gong, J. Yang, L. Liu, Y. Zhang, I. D. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *CVPR*, volume 1, page 5, 2017.

[21] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, volume 2, page 5, 2017.

[22] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision*, pages 482–496. Springer, 2010.

[23] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and automation (ICRA), 2014 IEEE international conference on*, pages 1524–1531. IEEE, 2014.

[24] C. Hane, L. Ladicky, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 381–389, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems*, pages 641–648, 2009.

[27] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. 2011.

[28] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE, 2005.

[29] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[30] M. Hradiš, J. Kotera, P. Zemcík, and F. Šroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, volume 10, page 2, 2015.

[31] T. Hyun Kim, B. Ahn, and K. Mu Lee. Dynamic scene deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3160–3167, 2013.

[32] T. Hyun Kim and K. Mu Lee. Segmentation-free dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2766–2773, 2014.

[33] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[34] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[35] M. Jin, G. Meishvili, and P. Favaro. Learning to extract a video sequence from a single motion-blurred image. *arXiv preprint arXiv:1804.04065*, 2018.

[36] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[37] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[38] T. H. Kim, S. Nah, and K. M. Lee. Dynamic scene deblurring using a locally adaptive linear blur model. *arXiv preprint arXiv:1603.04265*, 2016.

[39] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee. Learning-based, automatic 2d-to-3d image and video conversion. *IEEE Transactions on Image Processing*, 22(9):3485–3496, 2013.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[41] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[42] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014.

[43] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

[44] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy*, pages 22–29, 2017.

[45] X. Li, H. Qin, Y. Wang, Y. Zhang, and Q. Dai. Dept: depth estimation by parameter transfer for single still images. In *Asian Conference on Computer Vision*, pages 45–58. Springer, 2014.

[46] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016.

[47] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014.

[48] X. Liu, Y. Zhao, and S.-C. Zhu. Single-view 3d scene parsing by attributed grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–691, 2014.

[49] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[50] A. Maksai, X. Wang, F. Fleuret, and P. Fua. Non-markovian globally consistent multi-object tracking.

In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2544–2554, 2017.

[51] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798. Springer, 2014.

[52] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[53] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, volume 1, page 3, 2017.

[54] T. Narihira, M. Maire, and S. X. Yu. Learning lightness from human judgement on relative reflectance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2973, 2015.

[55] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[56] S. K. Nayar. Shape from focus. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1989.

[57] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *ICCV*, pages 4762–4770, 2017.

[58] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *German Conference on Pattern Recognition*, pages 65–77. Springer, 2017.

[59] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[60] J. Pan, J. Dong, Y.-W. Tai, Z. Su, and M.-H. Yang. Learning discriminative data fitting functions for blind image deblurring. In *ICCV*, pages 1077–1085, 2017.

[61] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, and M.-H. Yang. Soft-segmentation guided object motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2016.

[62] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.

[63] L. Pan, Y. Dai, M. Liu, and F. Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6987–6996. IEEE, 2017.

[64] 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 Conference on Computer Vision*, 2017.

[65] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4058–4066, 2016.

[66] W. Ren, J. Pan, X. Cao, and M.-H. Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. *arXiv preprint arXiv:1708.03423*, 2017.

[67] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016.

[68] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[69] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.

[70] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[71] A. Sellent, C. Rother, and S. Roth. Stereo video deblurring. In *European Conference on Computer Vision*, pages 558–575. Springer, 2016.

[72] E. Shelhamer, J. T. Barron, and T. Darrell. Scene intrinsics and depth from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–44, 2015.

[73] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *ACM Transactions on Graphics (TOG)*, 34(6):225, 2015.

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

[75] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *CVPR*, volume 2, page 6, 2017.

[76] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.

[77] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on computer vision and pattern recognition (CVPR)*, volume 5, page 6, 2017.

[78] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[79] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015.

[80] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017.

[81] S. Wang, R. Clark, H. Wen, and N. Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5):513–542, 2018.

[82] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[83] X. Wang, E. Türetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision*, pages 17–32. Springer, 2014.

[84] X. Wang, E. Türetken, F. Fleuret, and P. Fua. Tracking interacting objects using intertwined flows. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2312–2326, 2016.

[85] Y. Wang, C. Xu, J. Qiu, C. Xu, and D. Tao. Towards evolutionary compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2476–2485, 2018.

[86] P. Wieschollek, M. Hirsch, B. Schölkopf, and H. P. Lensch. Learning blind motion deblurring. In *ICCV*, pages 231–240, 2017.

[87] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016.

[88] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 6, 2017.

[89] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*, pages 1618–1625, 2017.

[90] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 469–484, 2018.

[91] X. You, Q. Li, D. Tao, W. Ou, and M. Gong. Local metric learning for exemplar-based object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(8):1265–1276, 2014.

[92] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[93] H. Zhang and D. Wipf. Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In *Advances in Neural Information Processing Systems*, pages 1556–1564, 2013.

[94] H. Zhang and J. Yang. Intra-frame deblurring by leveraging inter-frame camera motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4036–4044, 2015.

[95] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

[96] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2614–2622, 2015.

[97] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.

[98] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–622, 2015.