Learning to See through Reflections

Meiguang Jin
University of Bern
Switzerland
jin@inf.unibe.ch

Sabine Süsstrunk
École Polytechnique Fédérale de Lausanne
Switzerland
sabine.susstrunk@epfl.ch

Paolo Favaro
University of Bern
Switzerland
favaro@inf.unibe.ch

Abstract

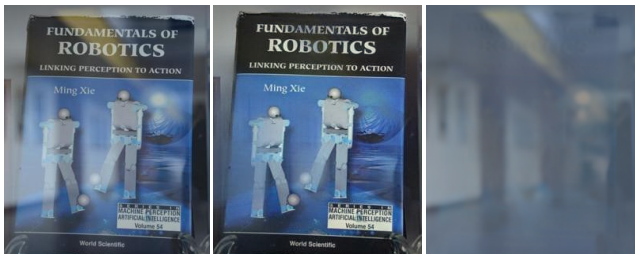
Pictures of objects behind a glass are difficult to interpret and understand due to the superposition of two real images: a reflection layer and a background layer. Separation of these two layers is challenging due to the ambiguities in assigning texture patterns and the average color in the input image to one of the two layers. In this paper, we propose a novel method to reconstruct these layers given a single input image by explicitly handling the ambiguities of the reconstruction. Our approach combines the ability of neural networks to build image priors on large image regions with an image model that accounts for the brightness ambiguity and saturation. We find that our solution generalizes to real images even in the presence of strong reflections. Extensive quantitative and qualitative experimental evaluations on both real and synthetic data show the benefits of our approach over prior work. Moreover, our proposed neural network is computationally and memory efficient.

1. Introduction

Glass is a common material that we often encounter in daily life. What makes it a fundamental component in the design of products, *e.g.*, windows in architecture and transportation, is that it allows light transmission due to its transparency. However, glass also reflects incoming light, so that pictures of objects through a window result in the superposition of two layers composed of transmitted and reflected images. More formally, an observed *superimposed* image $\mathbf{I}$ can be modeled as the superposition of two layers, that is,

$$\mathbf{I} = \mathbf{B} + \mathbf{R}, \tag{1}$$

where $\mathbf{B}$ is the background layer and $\mathbf{R}$ is the reflection layer. Unfortunately, this superposition challenges our understanding of objects in a scene due to the difficulty of assigning structural patterns to one layer or the other. Reflection removal is thus a layer separation problem [3]. It aims to restore $\mathbf{B}$ and $\mathbf{R}$ from a single image $\mathbf{I}$ (see, for example, Figure 1). Without additional information, this is an ill-posed task since $\mathbf{B}$ and $\mathbf{R}$ are perfectly interchangeable. One popular constraint used to distinguish the two layers is to assume that the reflection layer is blurrier than the background one, *i.e.*, that



$\mathbf{I}$ $\mathbf{B}$ $\mathbf{R}$

Figure 1. From a single, real, superimposed image $\mathbf{I}$, we obtain the reconstructed background $\mathbf{B}$ layer and reflection $\mathbf{R}$ layer.

$$\mathbf{I} = \mathbf{B} + \mathbf{R} * h \tag{2}$$

where h is a Gaussian blur and $*$ denotes convolution. While this assumption helps, it does not fully determine to which layer smooth regions belong to. This is particularly evident for the assignment of the *average color* of each layer (see Figure 2).

To address these ambiguities, previous methods seek to get additional information either through a custom hardware design [13, 14, 22, 23] or multiple images [4, 7, 8, 10, 17, 20, 27, 28, 29, 31]. However, reflection removal using a single image remains a very appealing and useful problem, which is gaining attention in the scientific community [2, 5, 15, 16]. A number of approaches introduce priors or constraints on the image gradients. The main benefit of using gradients is that they are invariant to the average color ambiguity and have limited sensitivity to smooth texture variations. However, one challenge of gradient-based constraints is that they provide *local* priors. Therefore, they cannot encourage the joint assignment of extended regions of texture to the same layer based on how compatible their content is. We call this ambiguity the *context ambiguity* (see Figure 2 third row). To address the context ambiguity, one

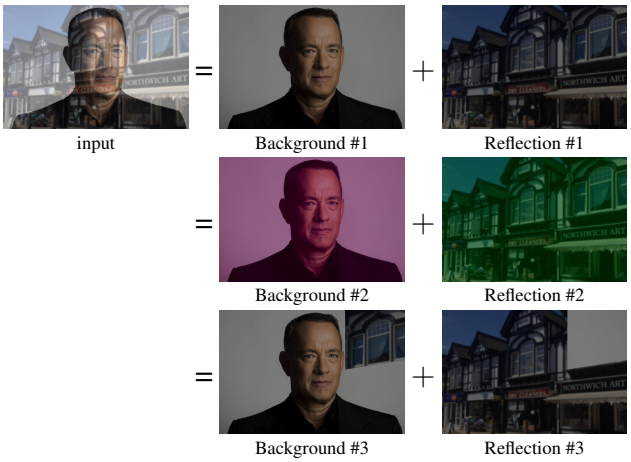


Figure 2. An input image can be represented as the superposition of the ground truth Background #1 and Reflection #1 layers. Due to the average color ambiguity, it can also be the superposition of Background #2 and Reflection #2 layers. Using local priors does not address the context ambiguity. As shown with Background #3 and Reflection #3 layers, we do not know to which layer the house facade should be assigned to until we impose global consistency of the content. The house detail in the reflection layer could also belong to the background layer as a wall poster.

could introduce global priors, for example, by using a neural network model with a large receptive field. The recent approach of Fan *et al*. [5] use, for the first time, a convolutional neural network (CNN) to remove reflections and to handle artifacts due to saturation. However, their method does not explicitly handle the average color ambiguity. Our approach combines the strengths of prior work: we introduce a neural network model that can capture global priors by using a resampling strategy and we avoid the average color ambiguity by explicitly assigning the average color of the input image to one of the two layers. Moreover, the resampling strategy increases the receptive field without sacrificing computational and data-storage efficiencies.

Our contributions can be summarized as follows:

1. We propose a novel synthetic data generation model that explicitly handles the average color ambiguity and saturation due to over and under exposure.

2. We address the context ambiguity by designing a neural network with a large receptive field via resampling.

3. We illustrate the benefits of our network through both quantitative and qualitative comparisons on real and synthetic data with state-of-the-art approaches.

4. The proposed network requires fewer parameters and is computationally more efficient than the only other existing neural network approach [5].

2. Prior Work

Existing reflection removal approaches can be divided into two categories: single image-based and multiple image-based approaches.

Multiple image approaches. Reflection removal is a highly ill-posed problem. Several reflection removal algorithms address the ill-posedness by using multiple images. Some methods minimize the correlation between the two layers by capturing multiple images with a different polarization [13, 14, 22, 23]. Agrawal *et al*. [1] separate the layers by using a pair of flash and non-flash images. Schechner *et al*. [21] separate the layers by using a pair of images captured with different focus settings. Others separate the layers by exploiting motion cues and minimizing the layer correlation in a short video sequence [4, 7, 8, 10, 17, 18, 20, 27, 28, 29, 31].

Single image approaches. Levin and Weiss [15] propose a user-assisted approach in which the user indicates which gradients belong to each layer to guide the optimization process. Shih *et al*. [25] propose to explore ghosting cues to separate the layers. They introduce a model of the ghosting reflection by using a double-impulse convolution kernel. Li and Brown [16] apply a smooth gradient prior to the reflection layer and a sparse gradient prior to the background layer, by making use of the observation that the reflection layer is generally blurrier than the background layer. Wan *et al*. [30] assume that pixels with higher depth of field (DOF) confidence belong to the desired background layer, and introduce a multi-scale strategy to classify the DOF confidence of edge pixels. Arvanitopoulos *et al*. [2] propose to suppress reflections by using a Laplacian data fidelity term and by imposing an L^0 gradient sparsity term to the background layer. Fan *et al*. [5] propose a new model to generate realistic synthetic superimposed images. These images are then used to train an edge-based deep neural network to output the corresponding background layer.

Despite the remarkable progress on single image reflection removal, current methods are still challenged by real images. One important limitation is due to the ambiguities discussed in the introduction, as they prevent the correct separation of structures in the input image. In addition, real images have clipped regions due to over and under exposure for very bright or very dark reflection, which results in saturated channels. This destroys the original content of the layers and therefore requires inpainting over extended regions. We show experimentally that our neural network trained with our proposed synthetic imaging model can handle the context and average color ambiguities as well as perform inpainting where saturation occurs.

(a) Background (b) Reflection (c) Convex model (d) Saturation model (e) Brightness [5] model

Figure 3. (c-e) Images generated by different data models with the given (a) background and (b) reflection layers.

3. Data Generation

As discussed in the previous sections, we propose to solve the reflection removal problem by using a neural network as a model. The next step in the training of such networks is the specification of the dataset of input-output image examples, *i.e.*, a set of ($\mathbf{I}$,$\mathbf{B}$) samples. Unfortunately, no dataset with real images is currently available and capturing a large number of real superimposed and background-only image pairs is not practical. An alternative way is to generate synthetic superimposed images with the corresponding ground truth background layers. The challenge, however, is in ensuring that the performance of a neural network trained on the synthetic images carries over to real images. In the next sections, we introduce different image data-generation models. We will validate our observations on each model in the experiments section by comparing the performance of neural networks trained on the corresponding dataset. To motivate our choices in defining the image data-generation model, in the next section we present an analysis on when a unique pair of background and reflection layers can be identified from the superimposed image.

3.1. Reconstruction Ambiguities

As shown in Figure 2, the reconstruction of the background and reflected layers from a single superimposed image is not unique. To illustrate the reconstruction ambiguities, we show that there exist a background layer $\hat{\mathbf{B}} \neq \mathbf{B}$ and a reflection layer $\hat{\mathbf{R}} \neq \mathbf{R}$ such that

$$\mathbf{I} = \hat{\mathbf{B}} + \hat{\mathbf{R}} * h = \mathbf{B} + \mathbf{R} * h, \tag{3}$$

where $\mathbf{B}$ and $\mathbf{R}$ are the ground truth images, and h is a Gaussian blur[1] (thus, h is nonnegative and integrates to 1). The justification for applying a Gaussian kernel to the reflection layer is that, in practice, a photographer tends to bring the background object into focus. In fact, this assumption has been used often in previous work [2, 8, 16, 19, 29, 32]. One family of ambiguities can be obtained by letting

$$\hat{\mathbf{B}} = \mathbf{B} + \mathbf{C} \quad \text{and} \quad \hat{\mathbf{R}} = \mathbf{R} - \mathbf{C}, \tag{4}$$

[1]We could consider ambiguities where also the blur may change. We leave this case to future work.

Table 1. Quantitative comparison between networks trained with and without harmonic components. We measure the performance with NCC, SSIM, PSNR and PSNR-harmonic metrics on 10 real images captured with ground truth background layers.

network\performance	NCC	SSIM	PSNR	PSNR-harmonic
harmonic	0.940	0.615	18.12	18.84
zero-mean	**0.947**	**0.656**	**18.21**	**19.66**

where $\mathbf{C}$ is a constant intensity image. To see that it satisfies eq. (3), we use the fact that $\mathbf{C} * h = \mathbf{C}$. We call this ambiguity the *average color* ambiguity.

More in general, however, it has been shown [6] that a harmonic function $\mathbf{H}$, *i.e.*, such that the Laplacian equation $\Delta\mathbf{H} = 0$, satisfies the equation $\mathbf{H} * h = \mathbf{H}$ for any circularly symmetric blur h (which includes Gaussian blurs). These harmonic functions include smooth shadings such as a linear change of intensity (in the pixel coordinates). Since all images can be split in harmonic and non-harmonic components, we can write $\mathbf{B} = \mathbf{B}_H + \mathbf{B}_{NH}$ and $\mathbf{R} = \mathbf{R}_H + \mathbf{R}_{NH}$, where $\Delta\mathbf{B}_H = 0 = \Delta\mathbf{R}_H$. Thus, one can define a more general family of ambiguities with $\Delta\mathbf{C}_H = 0$

$$\hat{\mathbf{B}} = \mathbf{B}_{NH} + \gamma\mathbf{B}_H + \beta\mathbf{R}_H + \mathbf{C}_H \tag{5}$$

$$\hat{\mathbf{R}} = \mathbf{R}_{NH} + (1-\gamma)\mathbf{B}_H + (1-\beta)\mathbf{R}_H - \mathbf{C}_H \tag{6}$$

for any γ, β.

So far we have discussed the reconstruction ambiguities based on measurable blur differences between the two images $\mathbf{B}$ and $\mathbf{R}$. Our neural network model, however, could learn to separate/disambiguate these two layers based also on their content. While in the case of the average ambiguity it is not possible to learn a meaningful disambiguation, in the case of harmonic components, it may be possible to achieve it. We then need to evaluate empirically what ambiguities the network is able to resolve. Towards this objective we train a network (introduced later in Sec. 4.1) on image pairs where only the average ambiguity has been removed and a second network on image pairs where the harmonic ambiguity has been removed. In the first case, superimposed images are generated by adding a background image to a zero-mean reflection image. In the second case, superimposed images are generated by adding a background image to the non-harmonic component of a reflection image (obtained as $\mathbf{R}_{NH} \simeq (1-w) * \mathbf{R}$, with a large blur kernel

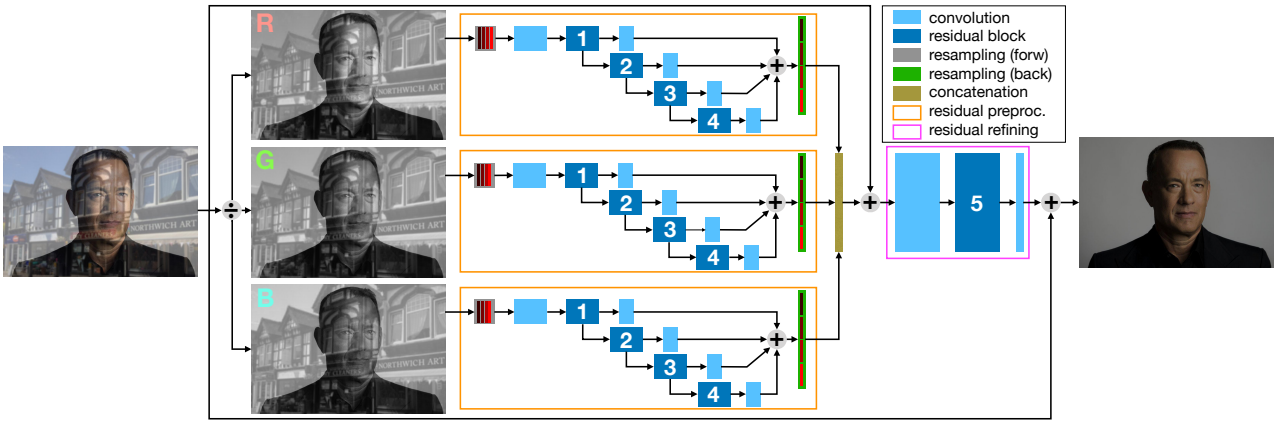


Figure 4. Network architecture.

w). In both cases the background image is the network output. We evaluate these networks on 10 real superimposed images that we captured together with the corresponding ground truth background layer. We show the reconstruction performance with NCC/SSIM/PSNR metrics in Table 1. To better evaluate the performance of these networks we also consider a PSNR metric where the harmonic components in both the ground truth background image and the reconstructed background image are removed, and call it PSNR-harmonic. We observe that the network trained on data without the harmonic component does not perform as well as the network trained on data without the average color. This may be due to the fact that the network can learn to correctly assign the harmonic components to the corresponding images. Thus, in the rest of the paper we consider dealing only with the average ambiguity. Notice that the ability of the network to identify the harmonic components requires a *large receptive field* (we describe in Sec. 4.1 how to achieve that).

Finally, we consider $\mathbf{C} = \bar{\mathbf{R}}$, where $\bar{\mathbf{R}}$ is the average of the reflection image and obtain

$$\hat{\mathbf{B}} = \mathbf{B} + \bar{\mathbf{R}} \quad \text{and} \quad \hat{\mathbf{R}} = \mathbf{R} - \bar{\mathbf{R}}. \tag{7}$$

Since this choice makes $\hat{\mathbf{R}}$ zero-mean, the mean of $\mathbf{I}$ will match the mean of $\hat{\mathbf{B}}$ without ambiguities. However, given $\hat{\mathbf{B}}$ and $\hat{\mathbf{R}}$ it will not be possible to find the original average color of $\mathbf{B}$ and $\mathbf{R}$ and thus distinguish $\hat{\mathbf{B}}$ from $\mathbf{B}$.

3.2. Data Generation Models

In the following, we introduce several image data generation models based on the considerations laid out above.

Convex Model. The most direct way of generating superimposed images is to apply the basic model of eq. (2)

$$\mathbf{I} = \mathbf{B} + \mathbf{R} * h. \tag{8}$$

We extend this model by introducing data augmentation through an arbitrary convex combination of the two layer images

$$\mathbf{I} = \alpha\mathbf{B} + (1-\alpha)\mathbf{R} * h, \quad 0 < \alpha < 1, \tag{9}$$

and then train on sample pairs $(\mathbf{I}, \mathbf{B})$. Figure 3 (c) shows a synthetic image generated with the above model where $\alpha = 0.6$. However, as argued in the previous section, networks trained with image pairs $(\mathbf{I}, \mathbf{B})$ will not generalize well to real images due to the ambiguity of the reconstruction task. We show this problem in the experiment and it has also been reported in recent work [5].

Zero-Mean Model. To remove the average color ambiguity we introduce the data generation model

$$\mathbf{I} = \mathbf{B} + \delta(\mathbf{R} - \bar{\mathbf{R}}) * h, \tag{10}$$

where δ is a positive scalar for data augmentation. As shown in the experiments, our neural network trained with samples $(\mathbf{I}, \mathbf{B})$ obtained through this model generalizes better than when trained with model (9).

Saturation Model. Although model (10) can already generalize better than model (9), there is an important limitation. Due to the fixed range of luminances that an imaging sensor can capture (we assume that the intensities of background and reflection images are normalized within $[0, 1]$), real superimposed images may exhibit over and under exposure resulting in saturated channels. Therefore, to make our generated superimposed images more realistic, we propose the following saturation model

$$\mathbf{I} = f(\mathbf{B} + \delta(\mathbf{R} - \bar{\mathbf{R}}) * h), \tag{11}$$

where f is a clipping function to saturate its argument at 0 when negative and at 1 when above 1. A visual example generated with the saturation model is shown in Figure 3 (d), where $\delta = 1$ is used.

Brightness Model. Recently, Fan *et al*. [5] proposed to increase the realism of the generated images by boosting the overall brightness in the data model. They argue that in a realistic superimposed image the reflection layer contributes

input/ground truth convex brightness [5] zero-mean saturation

Figure 5. Comparison between different data generating models on real data. Leftmost column: The superimposed image (top) and ground truth reflection layer (bottom). We show the background (top) and reflection (bottom) layers estimated after training our network on data generated with the convex model (second column), the brightness model (third column), the zero-mean model (fourth column) and the saturation model (fifth column).

to the brightest portion of the image. To account for this effect, they propose the following model

$$\mathbf{I} = f(\mathbf{B} + f(\mathbf{R} * h - \theta)), \tag{12}$$

where θ is a positive scalar adjusted depending on the intensities of $\mathbf{B}$ and $\mathbf{R}$. A synthetic superimposed image generated by the brightness model (12) is shown in Figure 3 (e). Although this model generates realistic data, it does not remove the average color ambiguity. We find experimentally that neural networks trained on data generated with this data model are not able to handle strong saturation effects.

4. Implementation

The design of our network needs to address the context and average color ambiguities. Thus, we strive for a large receptive field. To avoid the high computational cost of employing very large convolutional filters, we use resampling (that is, the rearrangement of an image as a tensor, where each channel collects pixels from the original image on a regular lattice with the same spacing, but different shift on the original image domain). Resampling has the advantage of enlarging the receptive field and at the same time of reducing the number of model parameters, because convolutions applied to a resampling channel are equivalent to strided dilated convolutions on the original image. This is a benefit to the overall computational efficiency and memory footprint of the model during execution, and allows to deal with input images at a much larger resolution than the existing neural network solution [5] (see Table 6).

4.1. Network Design

Inspired by the success of ResNet [9], our network employs a residual learning framework. The overall structure of our network is shown in Figure 4. Our proposed architecture comprises of two parts: three identical residual preprocessing sub-networks and a residual refining sub-network. Each residual preprocessing sub-network takes a color channel of the superimposed image as input and predicts the corresponding residual. Weights of the three residual preprocessing sub-networks are shared. The refining sub-network takes as input a color image obtained through the concatenation of the outputs of the preprocessing sub-networks. The refining sub-network compensates possible misalignments of the independently generated color channels. In a residual preprocessing sub-network, the first two layers are a resampling layer (gray box in Figure 4) with resampling factor $\sigma = 4$, followed by a convolutional layer (light blue box in Figure 4) with 48 filters of size $5 \times 5 \times 16$. The resampling operation creates $\sigma^2 = 16$ sub-sampled images. Each sub-sampled image is obtained by sampling the input image channel one pixel every σ pixels (along both axes). Every sub-sampled image differs by the initial sampled pixel on the original input (up to σ^2 possible initial positions). The resampling operation helps enlarge the receptive field very quickly in the beginning and also reduces the memory footprint. Then, the convolutional layer is followed by 12 residual blocks. Each dark blue box $(1, 2, 3, 4)$ of the residual preprocessing sub-network in Figure 4 contains 3 residual blocks. Besides two 3×3 convolutional layers, an additional 1×1 convolutional layer is used in our proposed residual block. All convolutions in the residual blocks use 48 feature channels and are followed by batch

Table 2. Quantitative comparison of our neural network output when trained on data from each of the data generation models and tested on data generated by all models. The accuracy is measured with the SSIM and the NCC metrics.

train\test (SSIM\NCC)	convex	zero-mean	saturation	brightness	average
convex	.789**.987**	.704**.984**	.561\.941	.751\.964	.701\.969
zero-mean	.759\.981	**.732**\.979	.646\.950	.806\.972	.736\.971
saturation	**.790**\.982	.719\.980	**.719****.974**	.809\.977	**.759****.978**
brightness	.724\.971	.697\.976	.599\.942	**.837****.984**	.714\.968

normalization and ReLU layers (except for the last layer in each sub-network). The middle 6 residual blocks (dark blue boxes #2 and #3 in Figure 4) use dilated convolutions to further enlarge the receptive field. More precisely, 12 convolutional layers use dilations $1, 2, 2, 4, 4, 8, 8, 4, 4, 2, 2, 1$ respectively. The overall receptive field of our network is 465×465 pixels, while the receptive field of [5] is 128×128 pixels. Since our network is deep, in addition to the output of the last residual block, we use the outputs of the intermediate blocks. As the outputs of all residual blocks are at the sub-sampled image resolutions, we go back to the original image resolution by inverting the resampling operation (green box in Figure 4), as done in [24]. Our refining sub-network contains 2 convolutional layers and a residual block (dark blue box 5 in Figure 4). A skip connection from the input of the residual blocks to the input and output of the refining sub-network completes the network. We emphasize the importance of a large receptive field through an ablation study in the experimental section.

4.2. Network Training

Our network is trained with an ℓ_1 loss on the image gradients and a perceptual loss [11] on the original images. While imposing a gradient constraint is common practice in prior work [2, 5, 16], introducing a perceptual loss on the background layer is novel. We find that the perceptual loss not only helps the training converge faster, but also significantly improves the restoration quality, as shown in the experimental section. More specifically, we use the following loss

$$\mathcal{L} = |\phi(\mathbf{I}) - \mathbf{B}|_1 + \mu \sum_{(j,i)\in S} |V_i(D_j(\phi(\mathbf{I}))) - V_i(D_j(\mathbf{B}))|_2^2, \tag{13}$$

where $\mu = 0.008$, the function ϕ is our proposed network, V_1, V_2 are `relu2_2` and `relu3_3` layers of the `vgg16` network [26], D_0, D_1, D_2 are three downsampling operations with scales 1, 0.5 and 0.25, respectively, and $S = \{0, 1, 2\} \times \{1, 2\}$. Our network is trained with the ADAM optimizer [12]. The learning rate is initialized at 0.001 and decreased to 0.0001 after 32000 iterations. Overall training takes 64000 iterations with a mini-batch size of 20 images at each iteration.

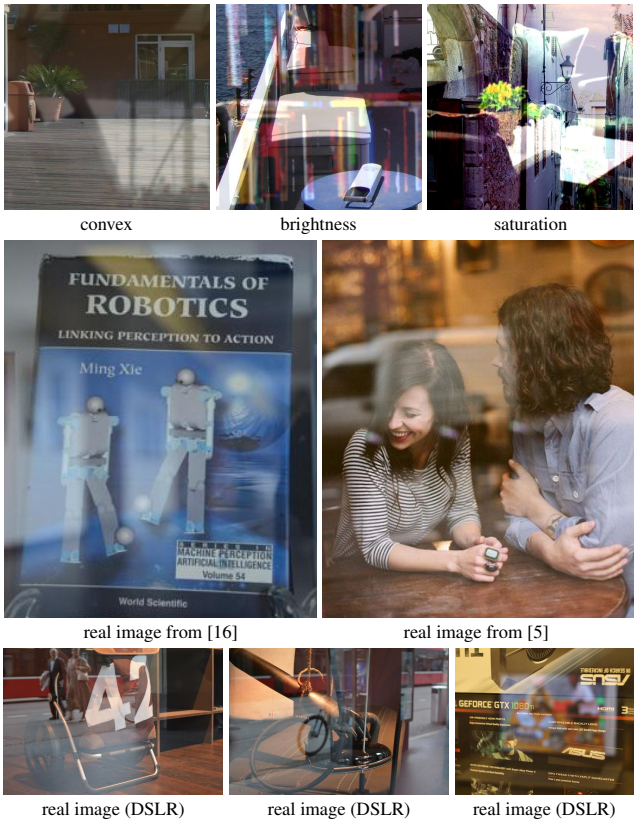


Figure 6. Superimposed images used in visual comparisons. The three synthetic images in the first row are generated with the convex, brightness and saturation data models. The real images on the second row are from [16] and [5]. The three images in the third row were captured with our DSLR camera.

5. Experiments

We evaluate the different data-generating models both quantitatively and qualitatively. This shows that the proposed saturation model, which avoids the average color ambiguity while making the synthetic data realistic, yields the best training data set. Based on this data, we compare different network architectures to illustrate the role of the receptive field, the network capacity and the loss functions. Finally, we compare our proposed neural network with state-of-the-art approaches on both real and synthetic data.

Visualization. In the visual comparisons we compute the reflection layer by subtracting the background layer predictions from the input superimposed images. Additionally, we add the average intensities of the superimposed images to the reflection layer images to get a better visualization.

Comparison of the Data Generation Models. To evaluate the effectiveness of each data simulation model, we

Table 3. Quantitative comparison between different network designs and training choices on the NCC and SSIM metrics.

network\performance	NCC	SSIM
baseline (32 channels)	0.9726	0.7160
24 channels	0.9678	0.6895
no-dilated conv.	0.9656	0.6959
loss ℓ_1	0.9690	0.6773
loss ℓ_1+VGG single-scale	0.9720	0.7152
proposed (48 channels)	**0.9750**	**0.7192**

Table 4. Quantitative comparison between different sub-sampling choices on the NCC, SSIM and PSNR metrics with 10 real images captured with ground truth background layers.

network\performance	NCC	SSIM	PSNR
non sub-sampling	0.920	0.583	17.49
sub-sampling 2	0.925	0.601	17.83
sub-sampling 3	0.938	0.632	18.10
[16]	0.901	0.484	15.93
[2]	0.903	0.490	15.74
[5]	0.943	0.613	18.15
proposed	**0.947**	**0.656**	**18.21**

train our network (see Figure 4) on four different training datasets generated from: the convex model, the zero-mean model, the saturation model and the brightness model of [5]. Due to the average color ambiguity, the peak signal-to-noise ratio (PSNR) is not a good metric for reflection removal algorithms. Thus, we use the normalized cross-correlation (NCC) and structural similarity (SSIM) metrics. Quantitative results are shown in Table 2. We observe that the zero-mean model introduces a substantial improvement in the average performance compared to the convex model, as predicted by the ambiguity analysis. Training on data generated from the saturation model introduces a further improvement that can be best appreciated when the trained network is tested on real data. We find that in this case the network generalizes well to all other datasets. Figure 5 shows a visual comparison of these trained networks on real images. The first column shows the superimposed input image (top) and ground truth reflection layer (bottom), which we captured with our DSLR camera. The following columns show the background layers predicted by the network (top) and the corresponding reflection layers (bottom) obtained as described above, from the convex model, the brightness model, the zero-mean model and the saturation model, respectively. Visual inspection reveals that training on data from the saturation model leads to the best layer separation.

Ablation Study (Network Architecture). In Table 3 we report the performance (measured by NCC and SSIM) of different network configurations. This shows the impact of each component in our network architecture design and training. The baseline network employs 32 feature channels in all residual convolutional layers instead of 48 for faster training and dilated convolutions are used in the middle 6 residual blocks (dark blue boxes 2 and 3 of Figure 4). Training is based on the loss (13) and the training data is generated by the saturation data model with $\delta \in (1, 1.25)$. The evaluation uses a test set with 100 synthetic superimposed images also generated from the saturation model. To evaluate the importance of the network capacity, receptive field and training loss, we modify the baseline network as follows: 1) we reduce the number of feature channels from 32 to 24; 2) we replace the dilated convolutional layers with the standard convolution (same number of filter elements), which give a smaller receptive field; 3) we train the baseline network with only the ℓ_1 loss; 4) we train the baseline network with ℓ_1 and a single-scale perceptual loss. Our proposed network uses 48 feature channels and is shown in the last row of Table 3. The quantitative results in Table 3 show that increasing the network capacity (row 2 vs row 1 vs row 6) and the receptive field (row 3 vs row 1) give a steady improvement. We also see that employing the perceptual loss gives a significant performance boost (row 4 vs row 5 vs row 1). Additionally, in Table 4 we also study the impact of the sub-sampling factor. We retrain three variants of our proposed network: non sub-sampling and sub-sampling with factors 2 and 3. From both synthetic and real image evaluations, we find that these variants are worse than our proposed network (with factor 4). The non sub-sampling variant has the worst performance, which we attribute to the fact that it has the smallest receptive field. Quantitative performance evaluations (NCC/SSIM/PSNR) on real images for these three variants and our proposed network are shown in Table 4.

Table 5. Quantitative comparison with the state-of-the-art methods on synthetic test sets generated with the convex, brightness [5] and saturation models.

data	convex		brightness [5]		saturation	
	NCC	SSIM	NCC	SSIM	NCC	SSIM
[16]	0.9485	0.6684	0.9549	0.7182	0.9184	0.4991
[2]	0.9480	0.6621	0.9628	0.7965	0.9180	0.5790
[5]	0.9563	0.7136	0.9735	**0.8102**	0.9311	0.6004
ours	**0.9822**	**0.7899**	**0.9765**	0.8092	**0.9739**	**0.7192**

Table 6. Comparison of the number of parameters, execution time and memory footprint of the proposed network with the one of Fan *et al*. [5] on three images at different resolutions. The evaluation is carried out on an NVIDIA TITAN X GPU.

		execution time (s)			memory (GB)		
method	# param	512^2	768^2	1024^2	512^2	768^2	1024^2
[5]	2.28M	0.26	0.62	1.44	3.6	7.9	>12
ours	**0.64M**	**0.02**	**0.05**	**0.09**	**0.4**	**1.0**	**1.4**

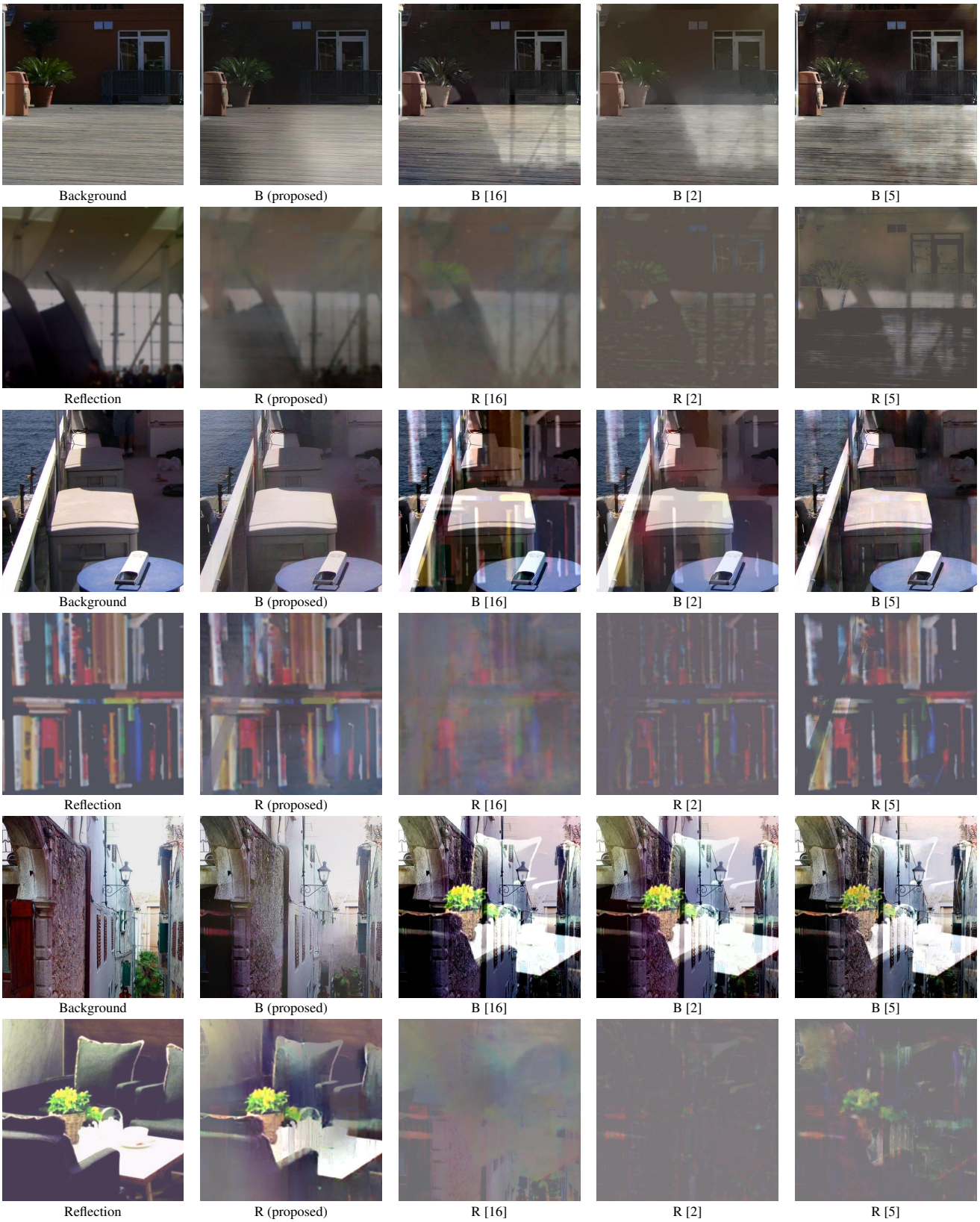
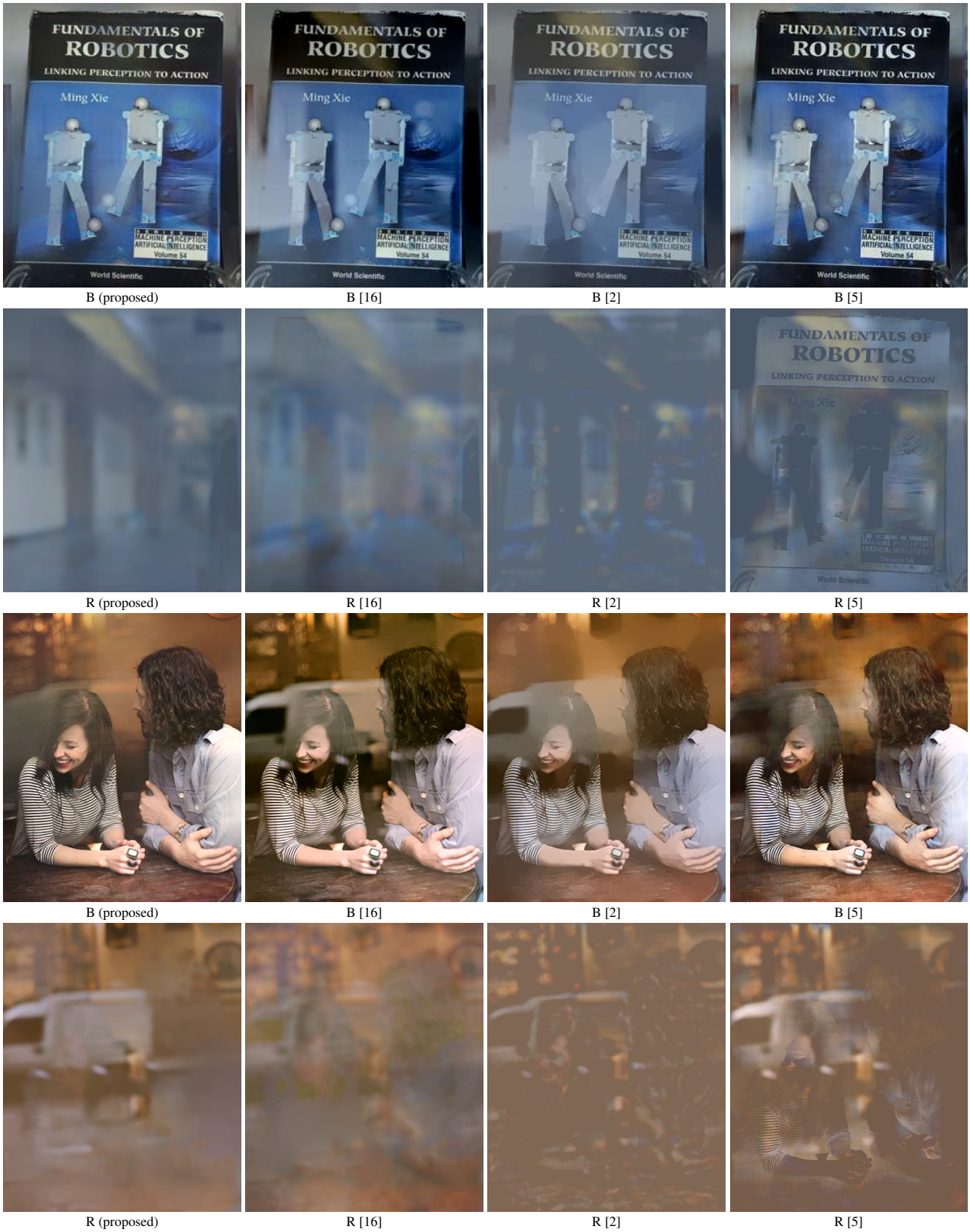


Figure 7. Synthetic comparison with state of the art approaches [2, 5, 16] on 3 superimposed images generated with the convex (top two rows), brightness (thrid and fourth rows) and saturation (last two rows) data models.



B (proposed) B [16] B [2] B [5]

R (proposed) R [16] R [2] R [5]

B (proposed) B [16] B [2] B [5]

R (proposed) R [16] R [2] R [5]

Figure 8. Visual comparison on real images from prior work.

Comparisons with the State-of-the-Art. We train the proposed network with synthetic superimposed images generate through the saturation model. We crop 320K non overlapping patches with size 448×448 pixels from the Places dataset [33] and select two random patches, one as the background layer and the other as the reflected layer, to generate each superimposed image $\mathbf{I}$. We compare our proposed network with the state-of-the-art optimization-based approaches [2, 16] and the neural network-based approach [5]. We evaluate all methods quantitatively on three synthetic dataset, each one containing 100 images generated from the convex, brightness [5] and saturation data models. Results in Table 5 show that, although our proposed network is trained with the saturation model, it can also generalize to other data models better than existing state-of-the-art methods. Three visual comparisons on synthetic images generated from the convex, brightness and saturation data models are shown in Figure 7. The ground truth background and reflection layers are shown on the leftmost column of Figure 7 and the corresponding three superimposed input images are shown on the first row of Figure 6. We observe that our proposed network reconstructs more plausible background and reflection layers compared to other methods. In particular, we see in the last two rows in Figure 7 that our neural network has learned to inpaint saturated regions as well as correctly separate content (the vase with the green plant and the desk) based on the context. Comparisons on two real examples are shown in Figure 8. The corresponding input superimposed images are shown on the second row of Figure 6: one is from Li *et al*. [16] and the other one is from Fan *et al*. [5]. Additionally, we also captured two outdoor real images and an indoor scene, where the ground truth background and reflection layers have been captured with a DSLR camera. The corresponding three input images are shown on the third row of Figure 6 and visual comparisons are shown in Figure 9. We observe that our proposed network trained on synthetic data generalizes to real data, thus confirming the validity of our analysis on the data generation models.

Computational and Memory Efficiency. An advantage of the resampling scheme is its computational and memory efficiency. To demonstrate the extent of its efficiency, we run our proposed network on three images at different resolutions. Execution time and memory footprint are shown in Table 6. Compared with the neural network of Fan *et al*. [5], our network is more than 10 times faster and requires much less memory storage. Notice that [5] takes more than 12GB of the memory to run on a 1024×1024 pixels image, which is not feasible with a TITAN X GPU. To process such large images with [5], we split them into several tiles, which introduces additional computational overhead. In contrast, our network can handle up to 11 megapixel images within 1.1s. These capabilities make our network suitable for mobile devices.

6. Conclusions

We have presented a novel deep learning approach to automatically separate two superimposed layers, a background and a reflection layer, from a single image. We describe analysis to make the layer separation problem well-posed and used it to devise a synthetic data generation model and to design a neural network with a wide receptive field. We trained our neural network on synthetic data generated through this model and showed that it generalizes well to real images and outperforms prior work.

Acknowledgements. MJ and PF acknowledge support from the Swiss National Science Foundation on project 200021_153324.

References

[1] A. K. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.*, 2005.

[2] N. Arvanitopoulos, R. Achanta, and S. Susstrunk. Single image reflection suppression. In *CVPR*, 2017.

[3] H. Barrow. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978.

[4] E. Be'ery and A. Yeredor. Blind separation of superimposed shifted images using parameterized joint diagonalization. *IEEE TIP*, 2008.

[5] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *ICCV*, 2017.

[6] P. Favaro. Shape from focus and defocus: Convexity, quasiconvexity and defocus-invariant textures. In *ICCV*, 2007.

[7] K. Gai, Z. Shi, and C. Zhang. Blind separation of superimposed moving images using image statistics. *IEEE TPAMI.*, 2012.

[8] X. Guo, X. Cao, and Y. Ma. Robust separation of reflection from multiple images. In *CVPR*, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 1994.

[11] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2015.

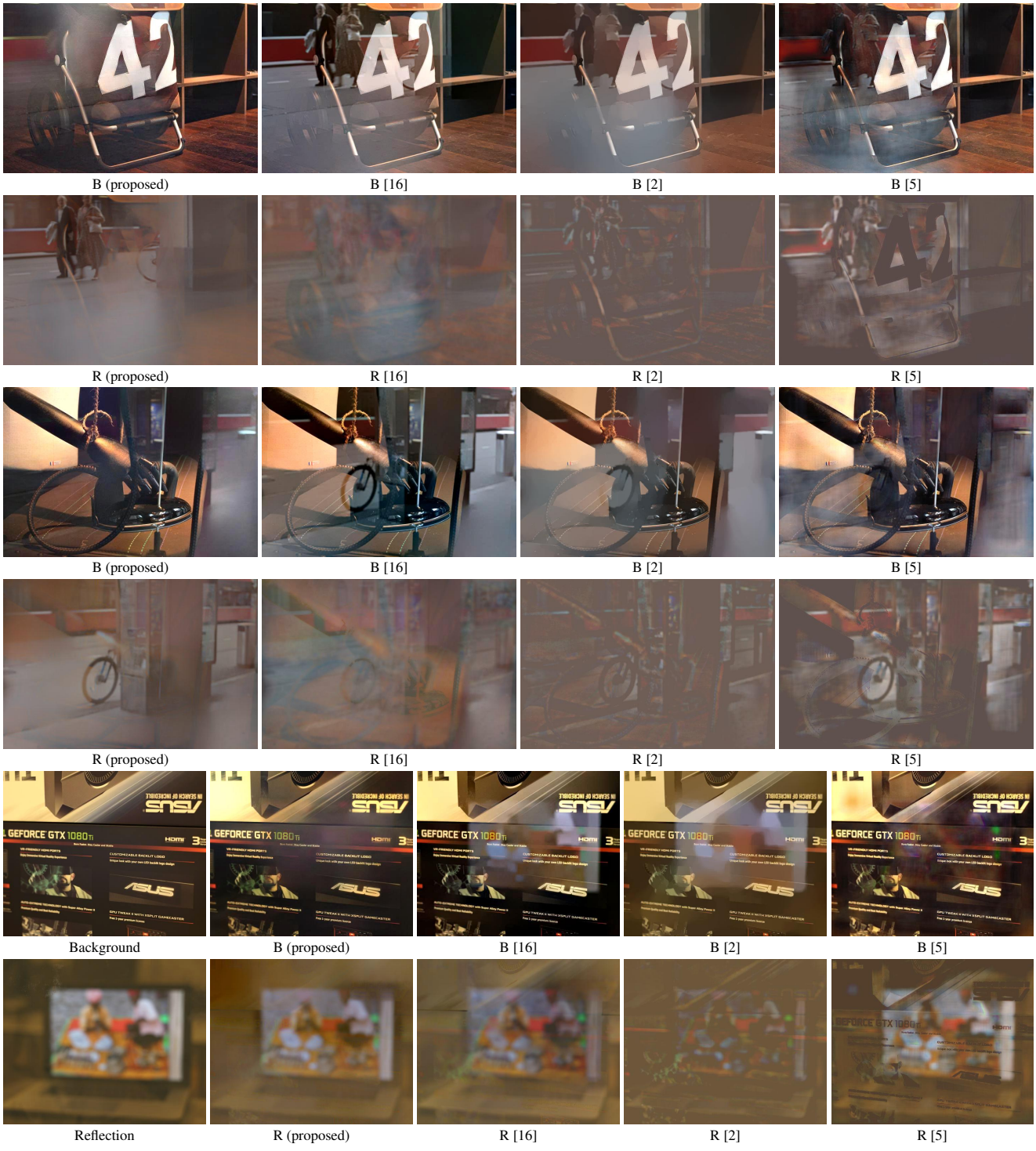


Figure 9. Visual comparison on real images captured with our DSLR camera.

[13] N. Kong, Y. Tai, and J. S. Shin. A physically-based approach to reflection separation: From physical modeling to constrained optimization. *IEEE TPAMI.*, 2014.

[14] J. Kopf, F. Langguth, D. Scharstein, R. Szeliski, and M. Goesele. Image-based rendering in the gradient domain. *ACM Trans. Graph.*, 2013.

[15] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE TPAMI.*, 2007.

[16] Y. Li and M. S. Brown. Single image layer separation

using relative smoothness. In *CVPR 2014*.

[17] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, 2013.

[18] A. Nandoriya, M. Elgharib, C. Kim, M. Hefeeda, and W. Matusik. Video reflection removal through spatio-temporal optimization. In *ICCV*, 2017.

[19] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *ECCV*, 2004.

[20] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV*, 2005.

[21] Y. Y. Schechner, N. Kiryati, and R. Basri. Separation of transparent layers using focus. *IJCV*, 2000.

[22] Y. Y. Schechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *ICCV*, 1999.

[23] Y. Y. Schechner, J. Shamir, and N. Kiryati. Polarization and statistical analysis of scenes containing a semireflector. *J. Opt. Soc. Am. A*, 2000.

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

[25] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *CVPR*, 2015.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[27] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Trans. Graph.*, 2012.

[28] C. Sun, S. Liu, T. Yang, B. Zeng, Z. Wang, and G. Liu. Automatic reflection removal using gradient intensity and motion cues. In *ACM, MM*, 2016.

[29] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, 2000.

[30] R. Wan, B. Shi, A. Tan, and A. C. Kot. Depth of field guided reflection removal. In *ICIP*, 2016.

[31] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman. A computational approach for obstruction-free photography. *ACM Trans. Graph.*, 2015.

[32] J. Yang, H. Li, Y. Dai, and R. T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *CVPR*, 2016.

[33] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.