Supplementary material: LSD$_2$ – Joint Denoising and Deblurring of Short and Long Exposure Images with CNNs

Janne Mustaniemi[1]
janne.mustaniemi@oulu.fi
Juho Kannala[2]
juho.kannala@aalto.fi
Jiri Matas[3]
matas@cmp.felk.cvut.cz
Simo Särkkä[4]
simo.sarkka@aalto.fi
Janne Heikkilä[1]
janne.heikkila@oulu.fi

[1] Center for Machine Vision and Signal Analysis, University of Oulu, Finland
[2] Department of Computer Science, Aalto University, Finland
[3] Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
[4] Department of Electrical Engineering and Automation, Aalto University, Finland

This document contains additional examples from the same datasets shown in the paper. Images are best viewed electronically and zoomed-in. Figures 1 - 5 show the results on realworld images (real motion blur and noise). Figures 6 - 8 show the results on synthetically corrupted images. Additional details of the exposure fusion method are given at the end of this document.

1 Additional results

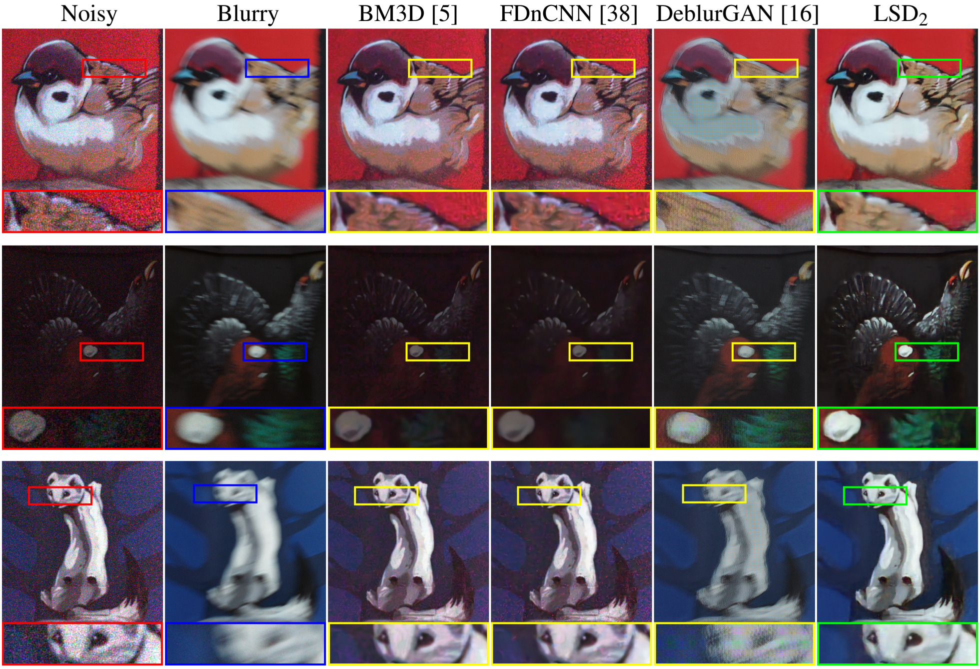

Figure 1: Static scene performance (*sparrow, capercaillie, weasel*).

Figure 2: Static scene performance (*bear, duck, fox, frog*).

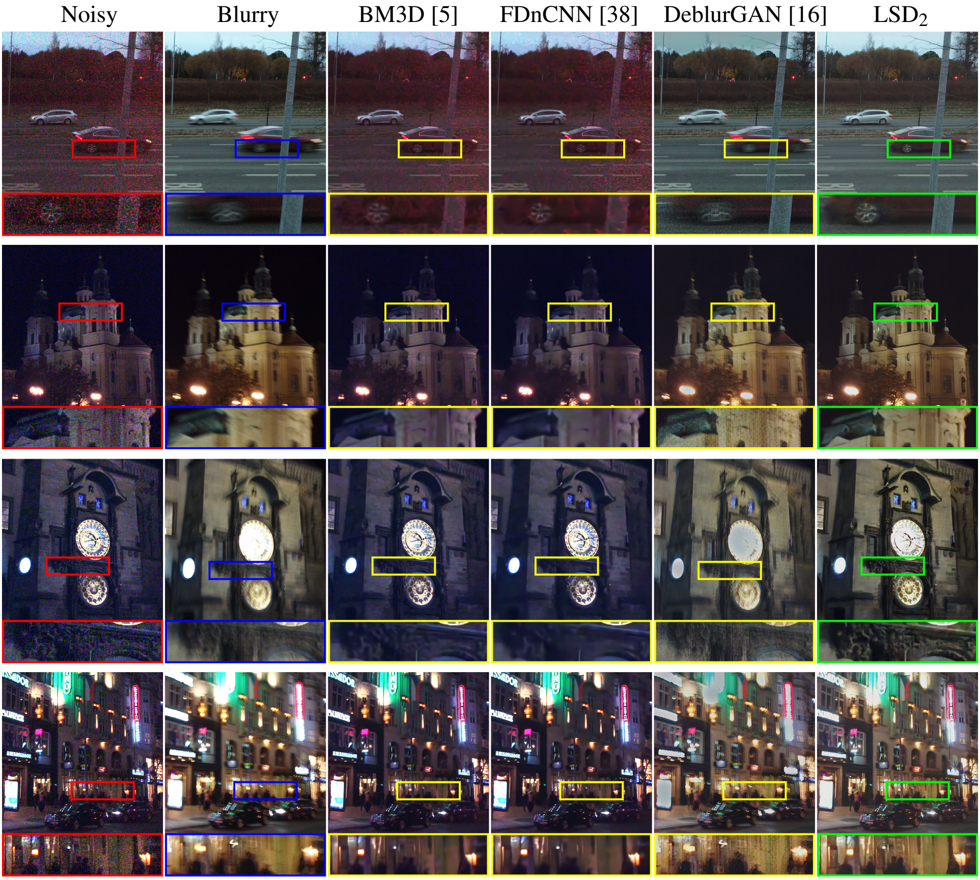

Figure 3: Dynamic scene performance and low-light performance including saturated pixels (*cars, church, clock, street*).

Figure 4: A comparison with Aittala and Durand [1]. A noisy image and a burst of blurry images (blue). The results of [1] obtained using a growing number of input images: 1, 3 and 6 (red). The result of LSD_2 (green). Notice that LSD_2 is able to recover more details (see for example the cone shaped wooden ornament in the first test case).

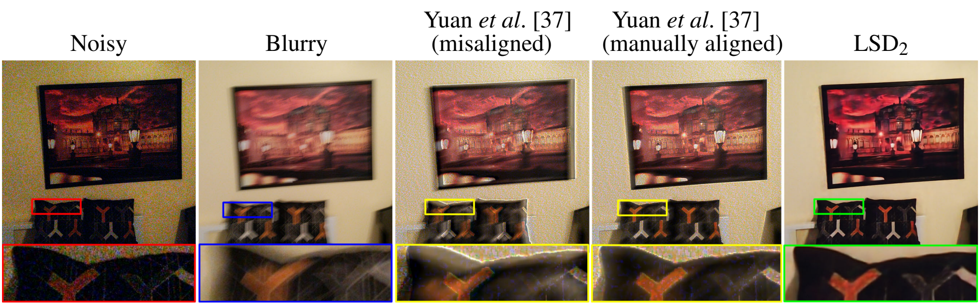

Figure 5: A comparison of LSD_2 and Yuan *et al.* [37].

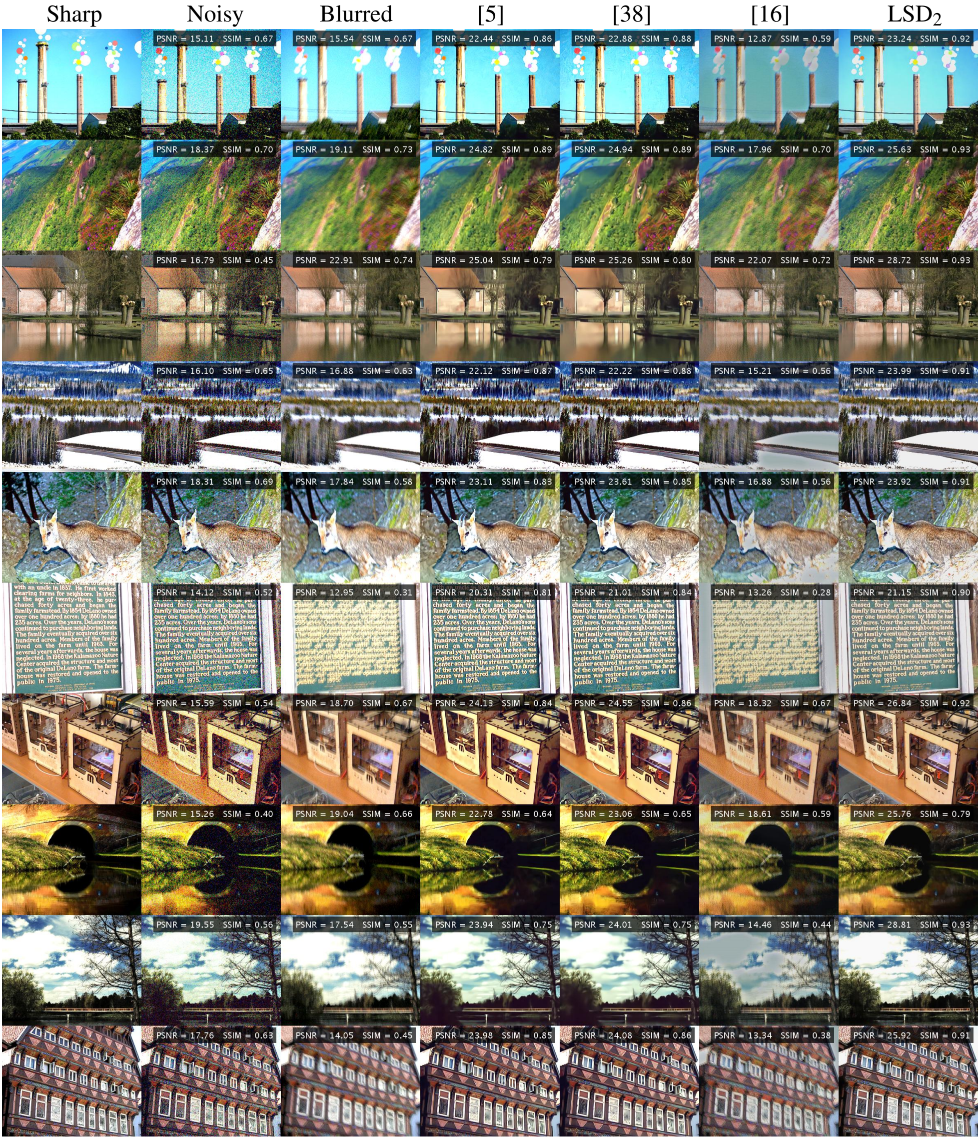

Figure 6: Results on synthetically corrupted images (1-10). Noisy images and the results of BM3D [5] and FDnCNN [38] have been normalized so that the mean intensity of each color channel matches the blurred image.

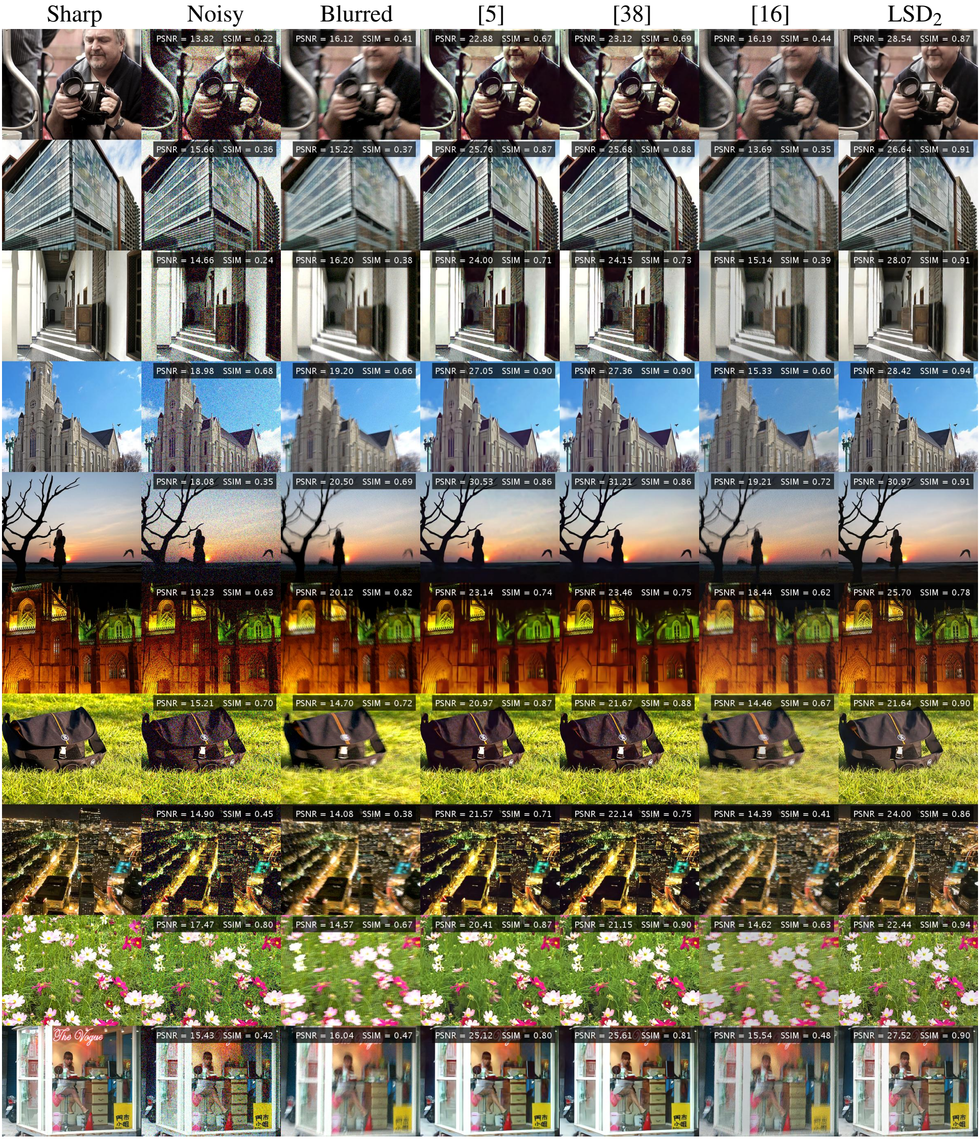

Figure 7: Results on synthetically corrupted images (11-20). Noisy images and the results of BM3D [5] and FDnCNN [38] have been normalized so that the mean intensity of each color channel matches the blurred image.

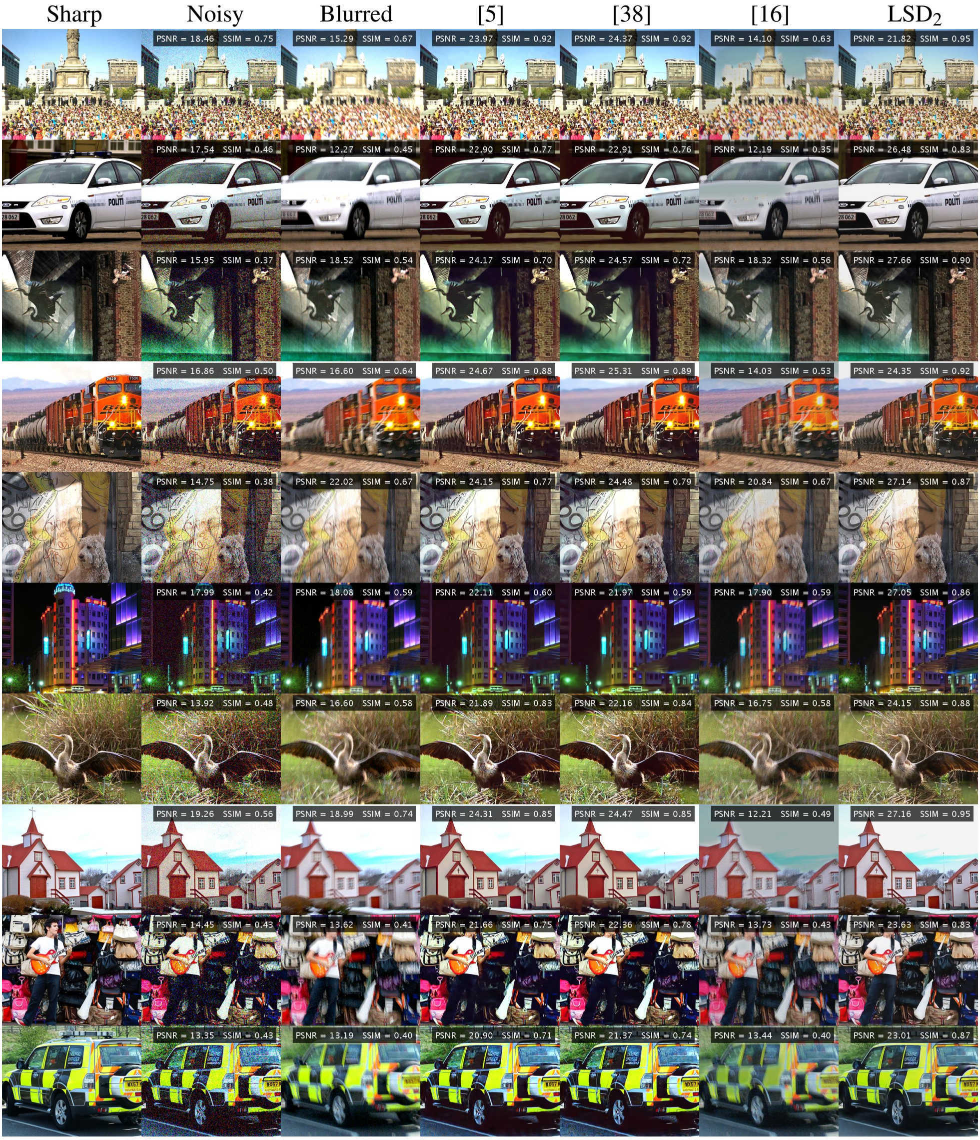

Figure 8: Results on synthetically corrupted images (21-30). Noisy images and the results of BM3D [5] and FDnCNN [38] have been normalized so that the mean intensity of each color channel matches the blurred image.

2 Exposure Fusion

The proposed exposure fusion network takes a pair of short and long exposure images as input. Let I_S and I_L denote the short and long exposure images, respectively. In our case, I_L is produced by the LSD$_2$ method. The output of the exposure fusion network is a weight map W, which is used to produce the fused image

$$\hat{I}_F(i,j,k) = W(i,j) \cdot I_L(i,j,k) + [1 - W(i,j)] \cdot I_S(i,j,k), \tag{1}$$

where (i,j,k) refers to pixel (i,j) in the k-th color channel. We then compute the mean squared error loss given the ground truth image I_F, presumably taken with "good exposure". In the following sections, we provide details of the network architecture and training.

2.1 Architecture

The network consists of 7 convolutional layers connected in a sequential manner. The input of the network is a pair of short and long exposure images I_S and I_L (stacked). The output is a weight map W with the same size as the input images (single channel). All convolutional layers use a 3×3 window, except the last layer, which is a 1×1 convolution. The number of feature maps is 16 for the layers 1, 2, 5 and 6, and 32 for the layers 3 and 4. Even though the network is very simple, it produces surprisingly good results as shown in Fig. 7 of the main paper. We note that alternative network architectures might provide further improvements.

2.2 Training

The network was trained on 50k images taken from an online image collection [12]. The training was done using synthetic long and short exposure image pairs as described in Sections 3.1.1 and 3.1.2 of the main paper. The resolution of the images was 270×480 pixels. We used the Adam [15] optimizer. The learning rate was set to 0.00002 and the network was trained for 5 epochs.