Spatially Variant Linear Representation Models for Joint Filtering Supplemental Material

Jinshan Pan[1] Jiangxin Dong[2] Jimmy Ren[3] Liang Lin[4] Jinhui Tang[1] Ming-Hsuan Yang[5,6]
[1]Nanjing University of Science and Technology [2]Dalian University of Technology
[3]SenseTime Research [4]Sun Yat-Sen University [5]UC Merced [6]Google Cloud

Overview

In this document, we first present the derivation details of the equation (12) in the main paper in Section 1. We then discuss why we use deep convolutional neural networks (CNNs) instead of the conventional models with hand-crafted priors to estimate the spatially variant linear representation coefficients in Section 2. In Section 3, we analyze the effectiveness of the proposed algorithm and compare it against methods based on end-to-end trainable networks. We further demonstrate the convergence property of the proposed algorithm in Section 4. Finally, we provide more experimental results of the proposed algorithm against the state-of-the-art deblurring methods in Section 5.

1. Derivations of (12) in the Manuscript

If we take $\varphi(\alpha)$ and $\phi(\beta)$ as $\mu\alpha^2$ and $\eta\beta^2$, the objective function (6) in the manuscript is

$$\mathcal{E}(\alpha,\beta) = \|\alpha G + \beta - I\|^2 + \mu\alpha^2 + \eta\beta^2, \tag{1}$$

where μ and η are positive weight parameters.

The gradients of $\mathcal{E}(\alpha,\beta)$ with respect to α and β are

$$\frac{\partial\mathcal{E}(\alpha,\beta)}{\partial\alpha} = 2G(\alpha G + \beta - I) + 2\mu\alpha, \tag{2a}$$

$$\frac{\partial\mathcal{E}(\alpha,\beta)}{\partial\beta} = 2(\alpha G + \beta - I) + 2\eta\beta. \tag{2b}$$

By setting $\frac{\partial\mathcal{E}(\alpha,\beta)}{\partial\alpha} = 0$, $\frac{\partial\mathcal{E}(\alpha,\beta)}{\partial\beta} = 0$, we can obtain

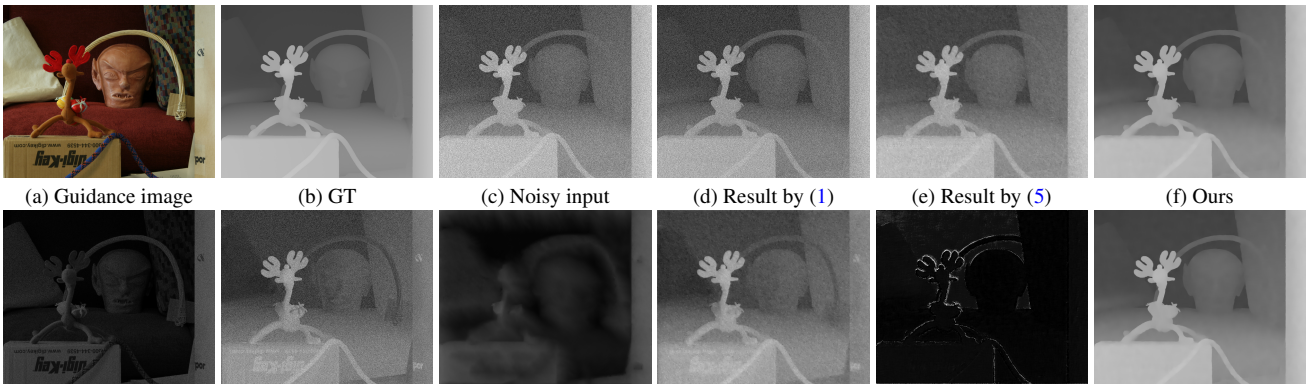
$$\alpha = \frac{GI - G\beta}{G^2 + \mu}, \qquad \beta = \frac{I - \alpha G}{1 + \eta}. \tag{3}$$

Based on (3), we can minimize (1) by solving

$$\alpha = \frac{\eta GI}{\eta G^2 + \mu + \mu\eta}, \qquad \beta = \frac{I - \alpha G}{1 + \eta}. \tag{4}$$

2. Why Using Deep CNNs Instead of Hand-crafted Priors for the Coefficient Estimation?

As discussed in Section 4.2 of the manuscript, it is not trivial to determine $\varphi(\alpha)$ and $\phi(\beta)$ as it is quite difficult to describe the statistical properties of α and β. In Section 6 of the manuscript, we have shown that using the model (6) based on commonly used priors (i.e., $\mu\alpha^2$ in [4]) does not always generate good results. However, one may wonder if the performance is mainly due to the effect of $\mu\alpha^2$ and $\eta\beta^2$ as these constraints are less effective to image noise. To answer this question, we



(a) Guidance image (b) GT (c) Noisy input (d) Result by (1) (e) Result by (5) (f) Ours

(g) $\alpha(G, I)$ by (1) (h) $\beta(G, I)$ by (1) (i) $\alpha(G, I)$ by (5) (j) $\beta(G, I)$ by (5) (k) Our $\alpha(G, I)$ (l) Our $\beta(G, I)$

Figure 1. Comparisons of the depth denoising results with different hand-crafted priors. Modeling the properties of the coefficients by hand-crafted priors is not a trivial task as it is quite difficult to describe the statistical properties of the linear representation coefficients. Thus, the models based on the commonly used hand-crafted priors do not generate clear images. In contrast, we develop a deep CNN which is constrained by the SVLRM to estimate the coefficients. With the estimated linear representation coefficients, the proposed method generates better denoised results (Best viewed on high-resolution display with zoom-in).

further use the sparsity of the image gradient (e.g., $\|\nabla\alpha\|_1$) as the constraint in the model (6) because this constraint is more robust to image noise and is able to preserve the main structures of the images. Thus, the objective function becomes

$$\mathcal{E}(\alpha, \beta) = \|\alpha G + \beta - I\|^2 + \mu\|\nabla\alpha\|_1 + \eta\|\nabla\beta\|_1. \tag{5}$$

We use the gradient descent method (7) of the manuscript to solve (5). At each iteration, we need to solve

$$\alpha^t = \alpha^{t-1} - \lambda \left(\frac{\partial \mathcal{E}(\alpha, \beta^{t-1})}{\partial \alpha} \right)_{\alpha=\alpha^{t-1}}, \tag{6a}$$

$$\beta^t = \beta^{t-1} - \lambda \left(\frac{\partial \mathcal{E}(\alpha^{t-1}, \beta)}{\partial \beta} \right)_{\beta=\beta^{t-1}}, \tag{6b}$$

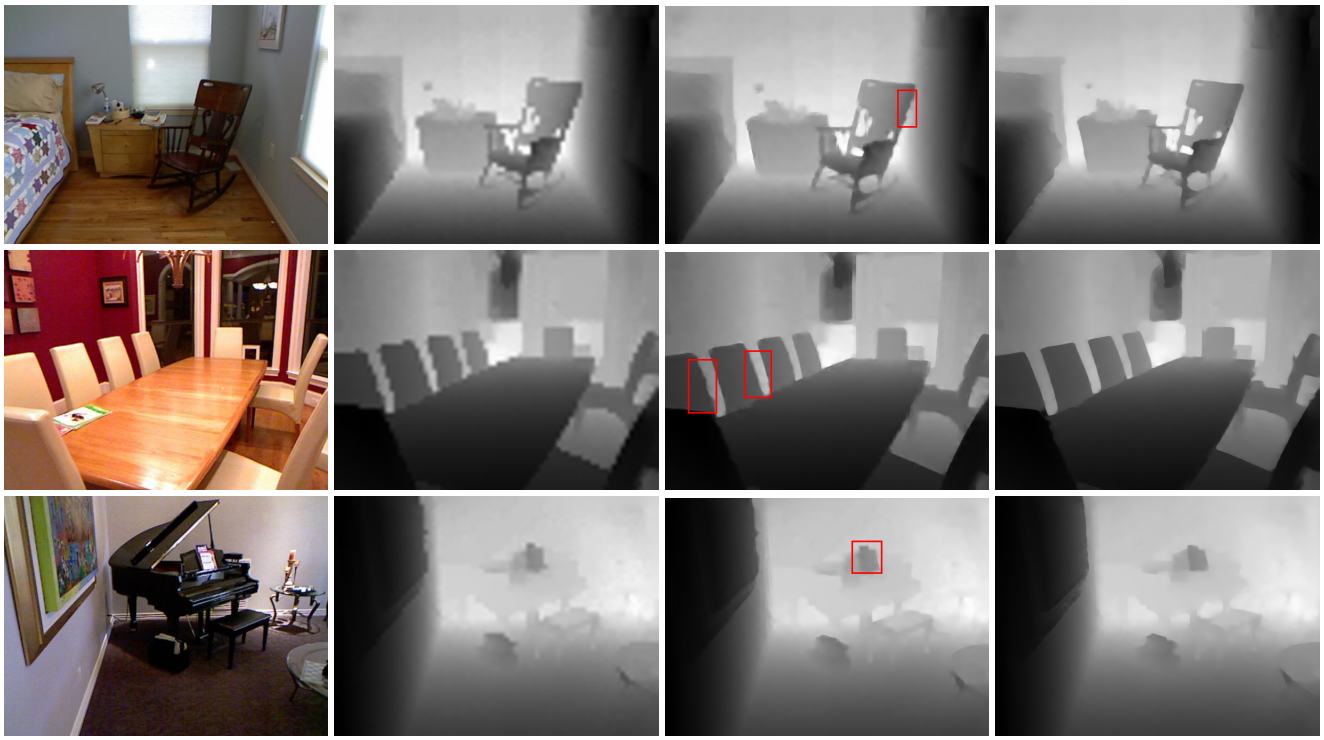
where $\frac{\partial \mathcal{E}(\alpha,\beta)}{\partial \alpha}$ and $\frac{\partial \mathcal{E}(\alpha,\beta)}{\partial \beta}$ are the partial derivatives w.r.t. α and β. We empirically set $\lambda = 0.01$, $t = 200$, and $\mu = \eta = 0.2$ for fair comparisons.

Figure 1 shows the comparisons of the results with different hand-crafted priors. Although using the sparsity of the image gradient as the constraint of the linear representation coefficients generates better results than those with the commonly used prior $\mu\alpha^2$ [4], the generated results still contain significant noise. Instead of using hand-crafted priors, we develop a deep CNN to estimate the linear representation coefficients. The use of deep CNNs to estimate the linear representation coefficients is motivated by the gradient descent method (7) of the manuscript as stated in Section 4.2. The proposed deep CNN is constrained by the SVLRM, which is able to estimate the linear representation coefficients (Figure 1(k) and (l)). Thus, the proposed algorithm is able to remove noise and generate better denoised results as shown in Figure 1(f).

3. Why Using the SVLRM Instead of the End-to-end Trainable Networks?

We note that several methods develop deep CNNs for joint image upsampling, e.g., [8, 5]. The target images are directly estimated by a deep CNN in a regression way. As the deep CNNs used in joint filtering are less effective for the details restoration [6], this accordingly leads to results containing halo effect or over-smoothed boundaries (see Figure 2(c)). In contrast, we develop a deep CNN to estimate the linear representation coefficients instead of the target images. The linear coefficients can determine whether the structures of the guidance image should be transferred to the target image or not as stated in Section 3 of the manuscript. Therefore, under the guidance of the linear representation coefficients, the SVLRM is able to generate the results with sharp edges (see Figure 2(d)).

In Section 6 of the manuscript, we have analyzed the effectiveness of the proposed algorithm against the methods based on end-to-end trainable networks. The results in Figure 2 further demonstrate the effectiveness of the proposed algorithm.



(a) Guidance images (b) Bicubic (c) End-to-end trainable network [8] (d) Ours

Figure 2. Comparisons of the depth upsampling results ($\times 8$) with end-to-end trainable networks. The boundaries of the parts enclosed in the red boxes in (c) are not preserved well. The proposed method generates the depth images with sharper boundaries (Best viewed on high-resolution display with zoom-in).

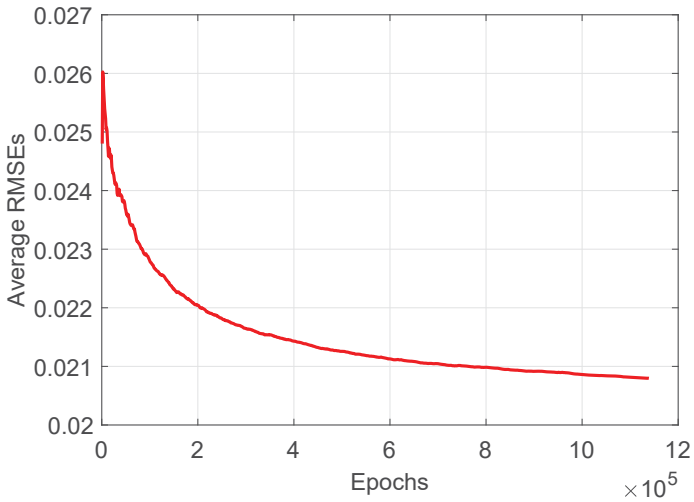


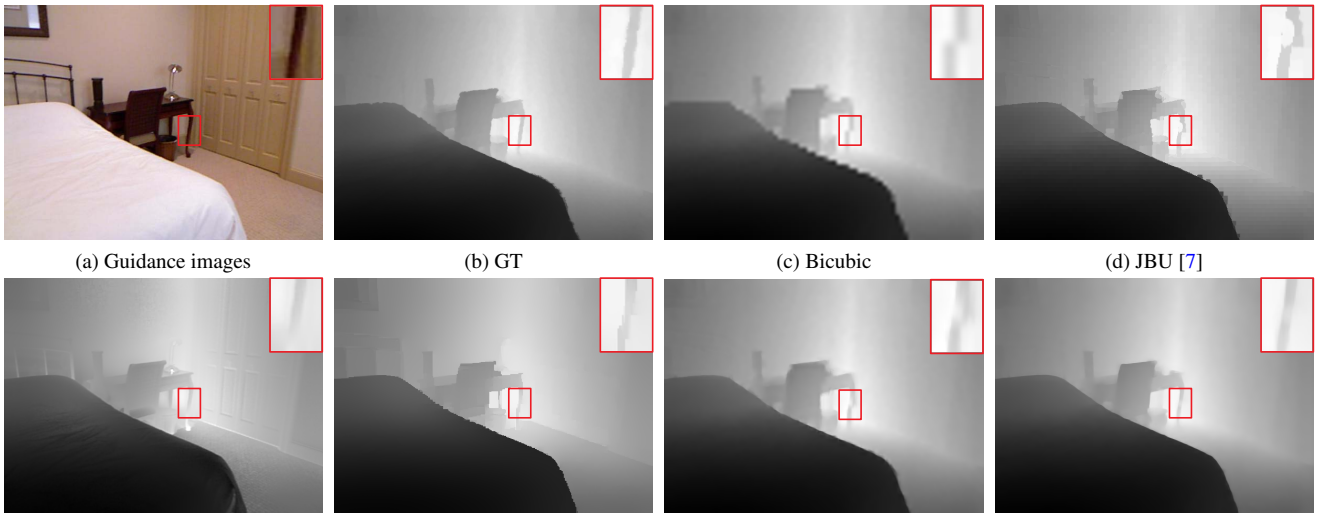
Figure 3. Quantitative evaluation of the convergence property on the depth image upsampling test dataset used in the manuscript. The deep CNN used for the spatially linear representation coefficient estimation converges well.

4. Convergence Property

To quantitatively evaluate the convergence properties of the proposed algorithm, we evaluate our method on the depth image upsampling test dataset used in the manuscript. Figure 3 shows that the proposed network converges well.

5. More Experimental Results

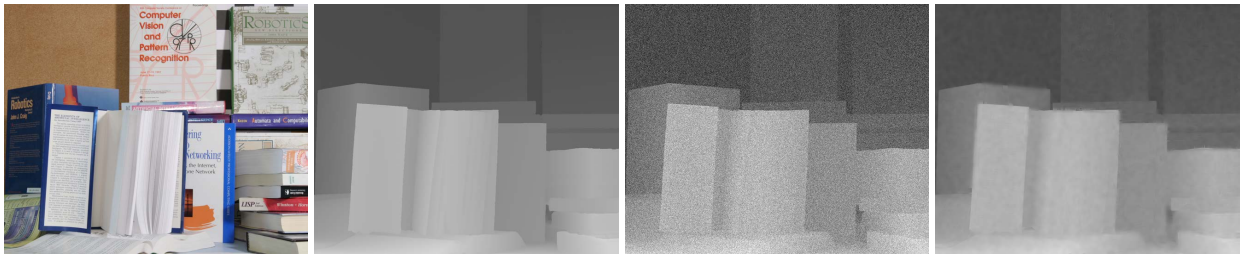
In this section, we provide more visual comparisons with state-of-the-art methods.



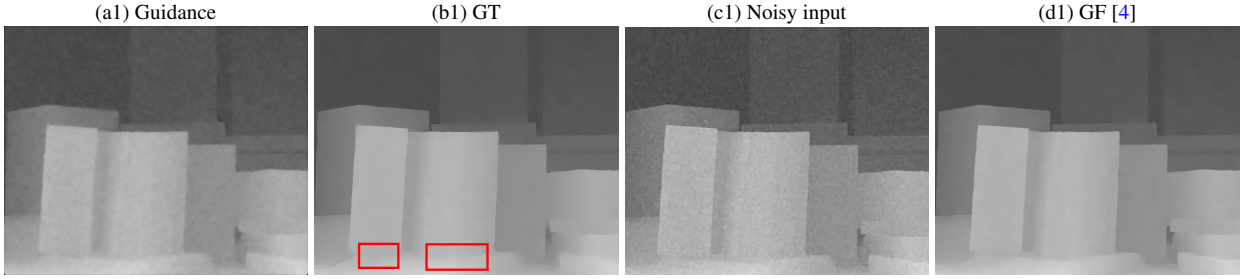
(a) Guidance images (b) GT (c) Bicubic (d) JBU [7]

(e) GF [4] (f) SDF [3] (g) DJF [8] (h) Ours

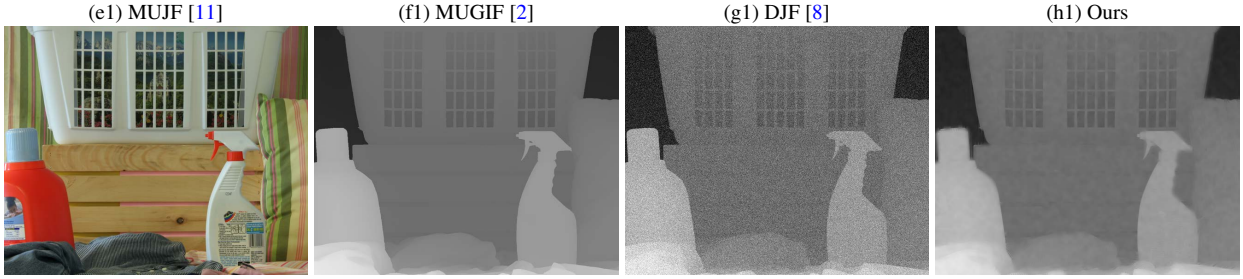
Figure 4. On the depth image upsampling application ($\times 8$). The proposed method generates the depth images with sharper boundaries and preserves the main structures well (Best viewed on high-resolution display with zoom-in).



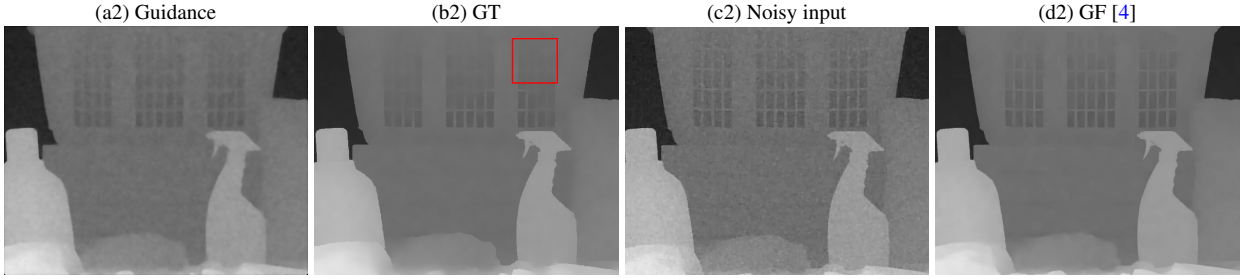
(a1) Guidance (b1) GT (c1) Noisy input (d1) GF [4]



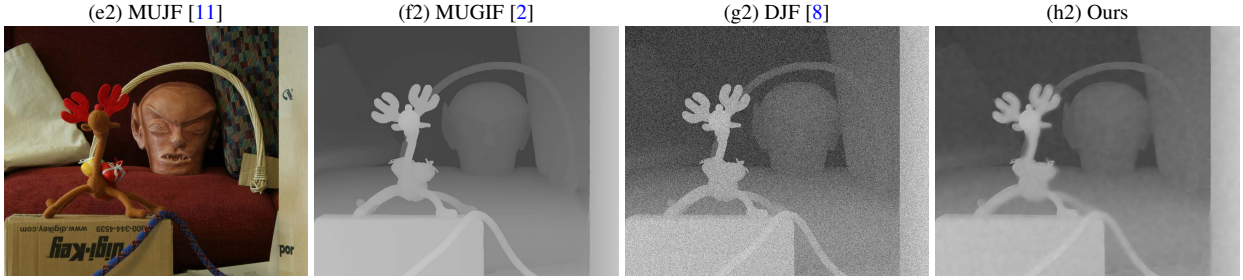
(e1) MUJF [11] (f1) MUGIF [2] (g1) DJF [8] (h1) Ours



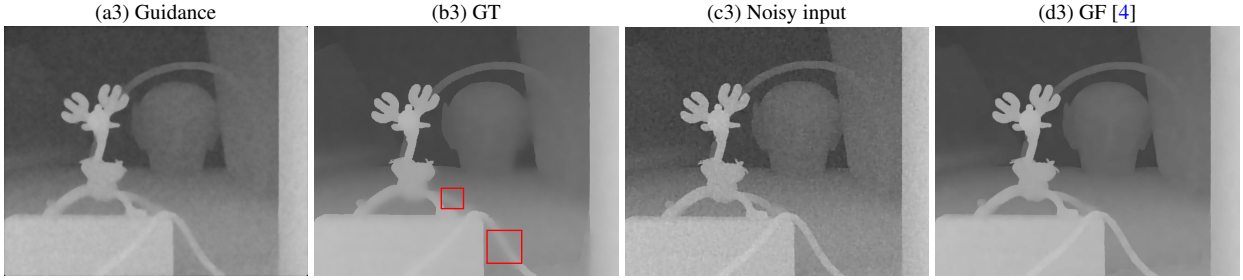
(a2) Guidance (b2) GT (c2) Noisy input (d2) GF [4]



(e2) MUJF [11] (f2) MUGIF [2] (g2) DJF [8] (h2) Ours

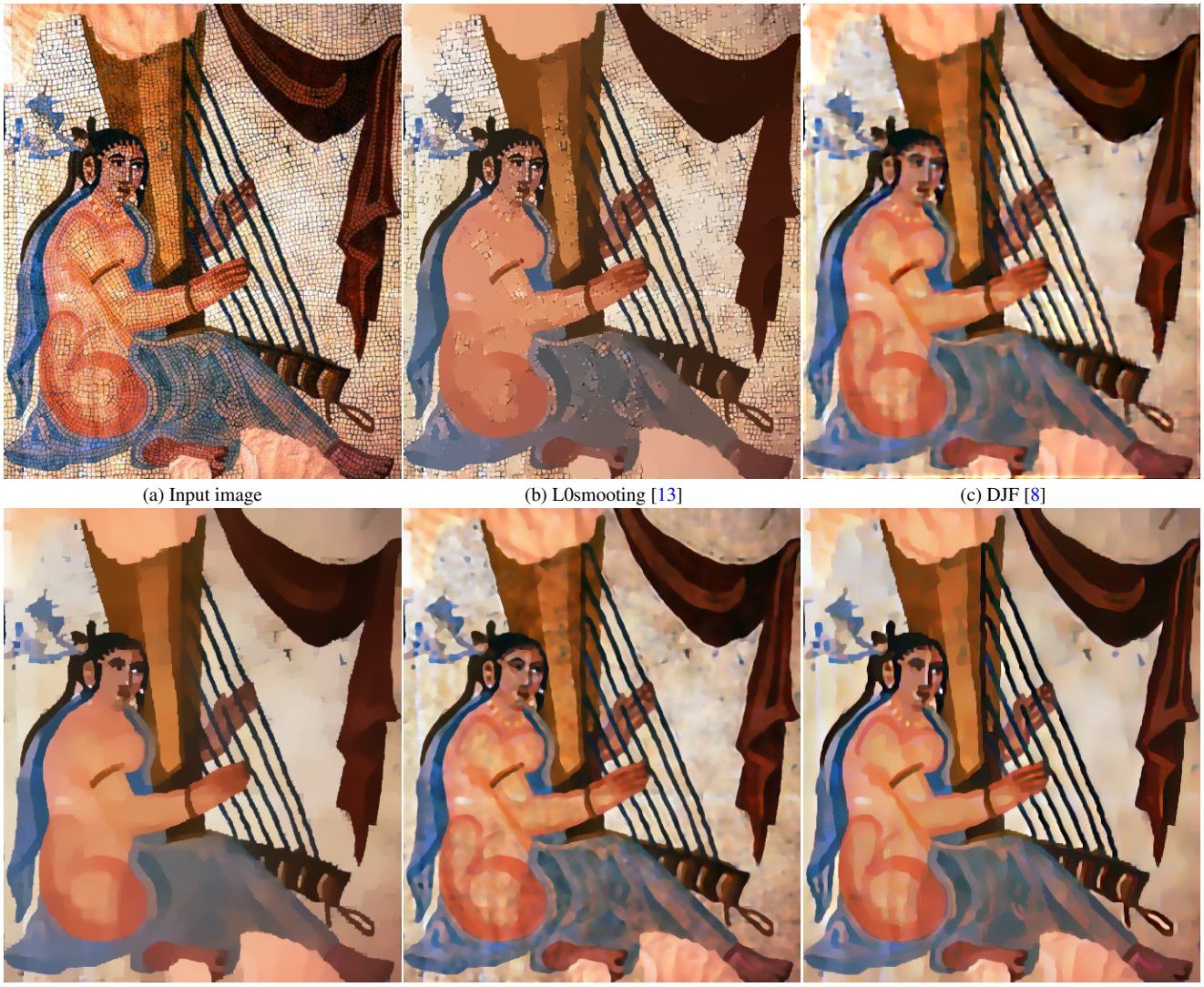


(a3) Guidance (b3) GT (c3) Noisy input (d3) GF [4]



(e3) MUJF [11] (f3) MUGIF [2] (g3) DJF [8] (h3) Ours

Figure 5. On the depth image restoration application. The parts enclosed in the red boxes in (f) are over-smoothed. The proposed method generates the depth images with sharper boundaries (Best viewed on high-resolution display with zoom-in).



(a) Input image (b) L0smooting [13] (c) DJF [8]

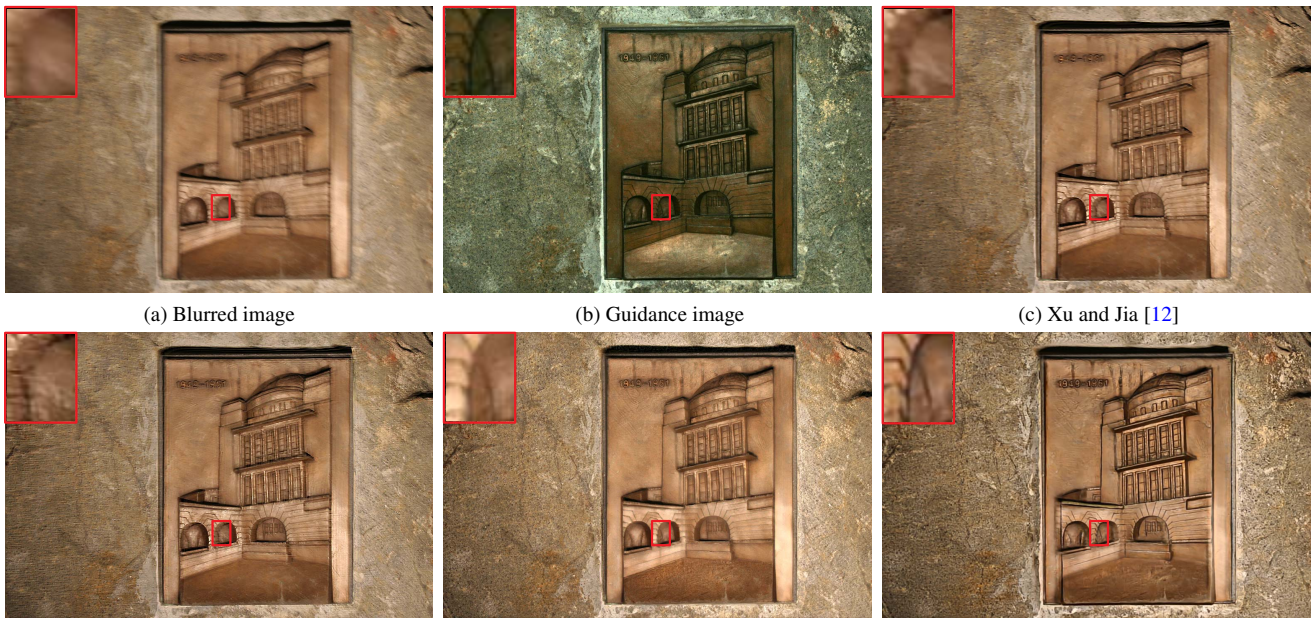
(d) RTV [14] (e) RGF [16] (f) Ours

Figure 6. On the scale-aware filtering application. The proposed algorithm is able to remove small-scale structures while preserving the main sharp edges (Best viewed on high-resolution display with zoom-in).

(a) Noisy input (b) EPLL [18] (c) CSF [10]

(d) MLP [1] (e) IRCNN [15] (f) Ours

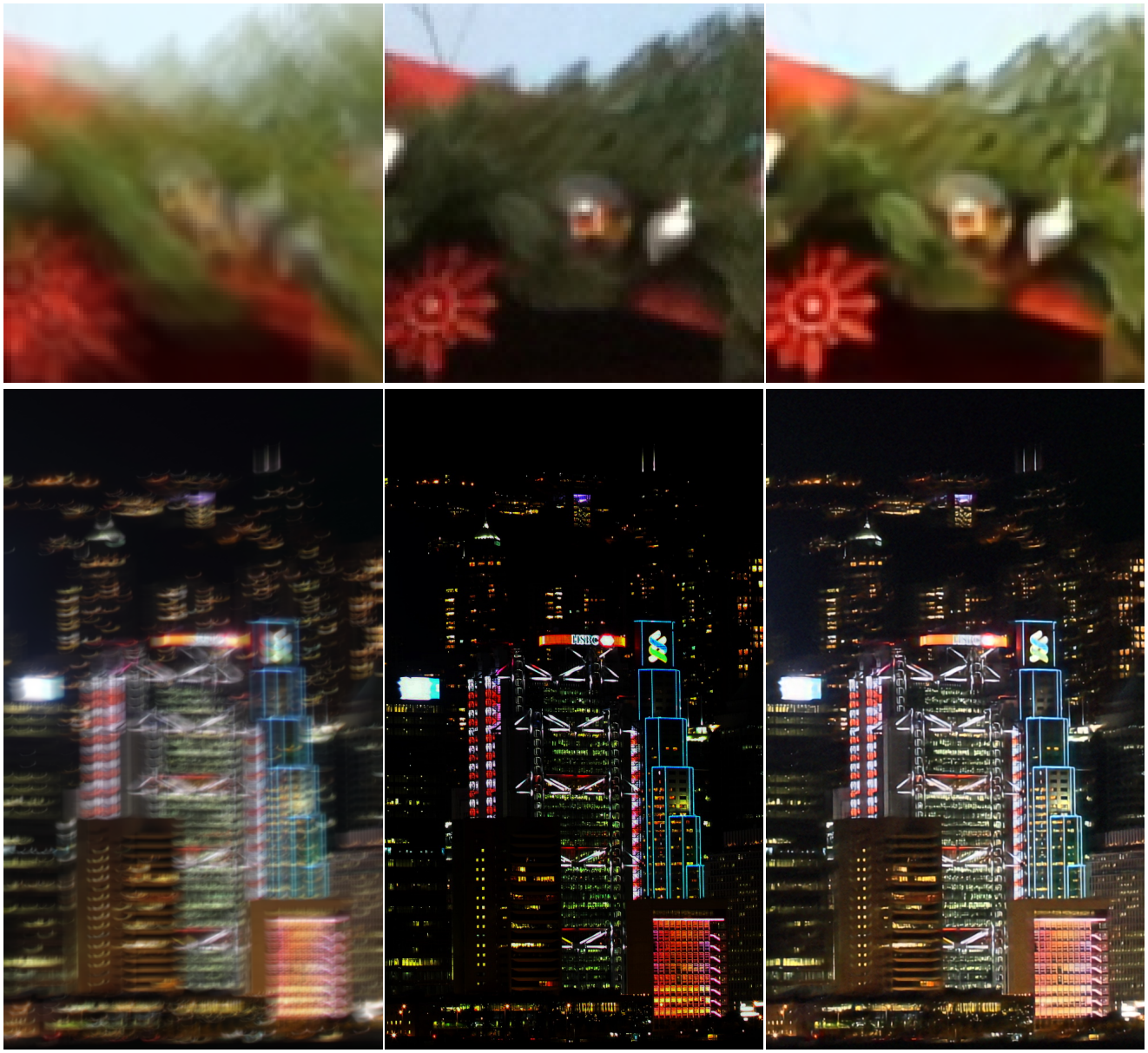
Figure 7. On the image denoising application. The proposed method generates a clearer image, where the structural details are preserved well (Best viewed on high-resolution display with zoom-in).



(a) Blurred image (b) Guidance image (c) Xu and Jia [12]

(d) Pan et al. [9] (e) Zhuo et al. [17] (f) Ours

Figure 8. On the image deblurring application. The proposed algorithm is able to generate much clearer images with finer structures and textures. (Best viewed on high-resolution display with zoom-in).



(a) Blurred image (b) Guidance image (c) Ours

Figure 9. More image deblurring results. The proposed algorithm is able to generate much clearer images with finer structures and textures. (Best viewed on high-resolution display with zoom-in).

References

[1] Harold Christopher Burger, Christian J. Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, pages 2392–2399, 2012. 7

[2] Xiaojie Guo, Yu Li, and Jiayi Ma. Mutually guided image filtering. In *ACM MM*, pages 1283–1290, 2017. 5

[3] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust guided image filtering using nonconvex potentials. *IEEE TPAMI*, 40(1):192–207, 2018. 4

[4] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE TPAMI*, 35(6):1397–1409, 2013. 1, 2, 4, 5

[5] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *ECCV*, pages 353–369, 2016. 2

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 2

[7] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matthew Uyttendaele. Joint bilateral upsampling. *ACM TOG*, 26(3):96, 2007. 4

[8] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *ECCV*, pages 154–169, 2016. 2, 3, 4, 5, 6

[9] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636, 2016. 7

[10] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, 2014. 7

[11] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. *IJCV*, 125(1-3):19–33, 2017. 5

[12] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010. 7

[13] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via L_0 gradient minimization. *ACM TOG*, 30(6):174:1–174:12, 2011. 6

[14] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM TOG*, 31(6):139:1–139:10, 2012. 6

[15] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, pages 2808–2817, 2017. 7

[16] Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia. Rolling guidance filter. In *ECCV*, pages 815–830, 2014. 6

[17] Shaojie Zhuo, Dong Guo, and Terence Sim. Robust flash deblurring. In *CVPR*, pages 2440–2447, 2010. 7

[18] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 7