Supplementary Material: Monte Carlo Denoising via Auxiliary Feature Guided Self-Attention

JIAQI YU, South China University of Technology, China
YONGWEI NIE*, South China University of Technology, China
CHENGJIANG LONG, JD Finance America Corporation, USA
WENJU XU, OPPO US Research Center, InnoPeak Technology Inc, USA
QING ZHANG, Sun Yat-sen University, China
GUIQING LI, South China University of Technology, China

ACM Reference Format:
Jiaqi Yu, Yongwei Nie, Chengjiang Long, Wenju Xu, Qing Zhang, and Guiqing Li. 2021. Supplementary Material: Monte Carlo Denoising via Auxiliary Feature Guided Self-Attention. *ACM Trans. Graph.* 40, 6, Article 1 (December 2021), 16 pages. https://doi.org/10.1145/3478513.3480565

1 SELF-ATTENTION IS ESSENTIALLY NON-LOCAL MEANS

Wang et al. [Wang et al. 2018] have shown that self-attention is actually non-local means filtering in the embedding space. Here, we describe their connection for the convenience of readers. Please refer to [Wang et al. 2018] for more details.

The non-local means filter was first proposed in [Buades et al. 2005]. Its discrete version [Wang et al. 2018] can be written as:

$$\boldsymbol{y}_{i,j} = \frac{1}{C(i,j)} \sum_{(a,b)\in\Omega} f(a,b,i,j)\boldsymbol{x}_{a,b}, \tag{1}$$

where Ω is the whole position lattice of an image, (i, j) and (a, b) denote pixel positions in the lattice, $\boldsymbol{x}_{a,b}$ represents the original color, $\boldsymbol{y}_{i,j}$ indicates the output color and $f(a, b, i, j)$ is a weighting function measuring the relationship between pixels at (a, b) and (i, j), defined as follows:

$$f(a,b,i,j) = e^{-\frac{d(\Phi(\boldsymbol{x}_{i,j}),\Psi(\boldsymbol{x}_{a,b}))}{\sigma^2}}, \tag{2}$$

where $\Phi(\boldsymbol{x}_{i,j})$ and $\Psi(\boldsymbol{x}_{a,b})$ denote two patches centered at $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{a,b}$ respectively, and $d(\Phi(\boldsymbol{x}_{i,j}), \Psi(\boldsymbol{x}_{a,b}))$ calculates the similarity of two patches, defined as the average of squared distances of pixels between two patches. Finally, $C(i, j)$ is used for normalization:

$$C(i,j) = \sum_{(a,b)\in\Omega} f(a,b,i,j). \tag{3}$$

Now, instead of viewing $\Phi(\boldsymbol{x}_{i,j})$ and $\Psi(\boldsymbol{x}_{a,b})$ as two patches centered at $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{a,b}$, we treat them as two vectors of $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{a,b}$ in a high-dimensional embedding space:

$$\begin{aligned} \Phi(\boldsymbol{x}_{i,j}) &= \boldsymbol{W}_\Phi \boldsymbol{x}_{i,j}, \\ \Psi(\boldsymbol{x}_{a,b}) &= \boldsymbol{W}_\Psi \boldsymbol{x}_{a,b}. \end{aligned} \tag{4}$$

where $\boldsymbol{W}_\Phi, \boldsymbol{W}_\Psi$ are learnable linear embedding mappings, and define $f(a, b, i, j)$ as:

$$f(a,b,i,j) = e^{\Phi(\boldsymbol{x}_{i,j})\Psi(\boldsymbol{x}_{a,b})^\top}. \tag{5}$$

Further, by replacing $\boldsymbol{x}_{a,b}$ by a vector $\Gamma(\boldsymbol{x}_{a,b}) = \boldsymbol{W}_\Gamma \boldsymbol{x}_{a,b}$ in the high-dimensional embedding space with $\boldsymbol{W}_\Gamma$ being a different learnable mapping, the non-local means filter formulation in Equation 1 then becomes:

$$\boldsymbol{y}_{i,j} = \mathrm{Softmax}_{a,b}(\Phi(\boldsymbol{x}_{i,j})\Psi(\boldsymbol{x}_{a,b})^\top)\Gamma(\boldsymbol{x}_{a,b}), \tag{6}$$

which is exactly the form of self-attention in [Vaswani et al. 2017] if setting query $Q = \Phi(\boldsymbol{x}_{i,j})$, key $K = \Psi(\boldsymbol{x}_{a,b})$, and value $V = \Gamma(\boldsymbol{x}_{a,b})$. This indicates that self-attention mechanism is essentially a special type of non-local means filter that operates in the high-dimensional embedding space rather than the original image space. Thereby, self-attention mechanism is inherently suited for denoising task.

2 QUANTITATIVE COMPARISONS WITH STATE-OF-THE-ART METHODS ON 4 TO 1024 SPP NOISY IMAGES

In the main paper, we only show the comparisons between our method and other three methods on 4 to 128 spp testing images due to space limit. In this material, Table 1 gives the comparison results on 4 to 1024 spp testing images. For most of the results, NFOR is the worst. However, for 1024 spp input, NFOR achieves the best results on PSNR and 1-SSIM metrics. This may be due to the reason that the three deep learning methods are trained on a 32 spp dataset. Our method overwhelms the other methods on the RMSE and PSNR metrics. For the 1-SSIM metric, our method outperforms the other methods on lower spp inputs (4 to 32 spp), and is comparable to other methods on higher spp inputs (128 to 1024 spp). The worse performance of our method on the 1-SSIM metric compared to the

*Yongwei Nie is the corresponding author.

Authors' addresses: Jiaqi Yu, Yongwei Nie, and Guiqing Li are with the School of Computer Science and Engineering, South China University of Technology. Chengjiang Long and Wenju Xu are with the JD Finance America Corporation. Qing Zhang is with the School of Computer Science and Engineering, Sun Yat-sen University. Authors' e-mails: kakeiyuscut@gmail.com, nieyongwei@scut.edu.cn, cjfykx@gmail.com, xuwenju123@gmail.com, zhangqing.whu.cs@gmail.com, ligq@scut.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
0730-0301/2021/12-ART1 $15.00
https://doi.org/10.1145/3478513.3480565

Table 1. Quantitative comparisons with NFOR [Bitterli et al. 2016], KPCN [Bako et al. 2017] and ACFM [Xu et al. 2019] on the testing set. Learning-based model are trained on the 32 spp training set. The "re" after the model name means that the model is retrained by ourselves. For ACFM, the results of their published model are also reported. The RMSE, PSNR, and 1-SSIM metrics are measured for each compared model. Bold numbers are the best, while numbers with a hat are the second best.

Model	SPP	RMSE↓	PSNR↑	1-SSIM↓
NFOR [2016]	4	27.469	29.2585	0.1447
	8	16.010	31.6135	0.1016
	16	8.460	33.6525	0.0751
	32	4.682	35.7583	0.0544
	128	1.748	39.2909	0.0323
	256	1.157	40.8636	0.0263
	512	0.879	42.$\hat{2}$375	0.0223
	1024	0.695	**43.4317**	**0.0198**
KPCN [2017] (re)	4	11.512	30.8417	0.1219
	8	6.631	32.9486	0.0888
	16	4.132	34.9944	0.0669
	32	2.840	36.6687	0.0509
	128	1.443	39.6035	0.0300
	256	1.116	40.6951	0.0260
	512	0.901	41.6172	0.0231
	1024	0.758	42.3305	0.0218
ACFM [2019]	4	10.668	31.5617	0.1166
	8	6.019	33.6399	0.0823
	16	3.697	35.6662	0.0593
	32	2.383	37.4461	0.0446
	128	1.192	40.2327	**0.0281**
	256	0.$\hat{9}$21	41.$\hat{2}$826	**0.0241**
	512	0.$\hat{7}$60	42.0705	**0.0215**
	1024	0.$\hat{6}$57	42.6611	**0.0198**
ACFM (re)	4	10.$\hat{2}$17	31.$\hat{9}$081	0.1$\hat{0}$76
	8	5.$\hat{3}$65	33.$\hat{9}$380	0.0$\hat{7}$81
	16	3.$\hat{5}$45	35.$\hat{7}$812	0.0$\hat{5}$86
	32	2.$\hat{3}$69	37.$\hat{5}$137	0.0$\hat{4}$43
	128	1.$\hat{1}$89	40.$\hat{2}$571	0.0$\hat{2}$83
	256	0.927	41.1544	0.0$\hat{2}$45
	512	0.763	42.0127	0.0$\hat{2}$18
	1024	0.661	42.5239	0.0$\hat{2}$01
Ours	4	**9.979**	**32.3770**	**0.1023**
	8	**4.855**	**34.6834**	**0.0713**
	16	**2.766**	**36.7625**	**0.0521**
	32	**1.839**	**38.3812**	**0.0410**
	128	**1.051**	**40.8028**	0.0286
	256	**0.840**	**41.8505**	0.0251
	512	**0.717**	**42.7167**	0.0226
	1024	**0.631**	43.$\hat{3}$798	0.0210

RMSE and PSNR metrics may be due to the overly aggressive edge-preserving effect of our method as shown in the last row of Figure 11 in the main paper.

3 ADDITIONAL QUALITATIVE COMPARISONS WITH STATE-OF-THE-ART METHODS

We show more qualitative results of our model from Figure 1 to 12 and compare with NFOR [Bitterli et al. 2016], KPCN [Bako et al. 2017] and ACFM [Xu et al. 2019]. We choose RMSE, PSNR and SSIM as evaluation metrics and show the RMSE heatmap between the output denoised image and the corresponding ground truth image. In the RMSE heat map, a darker red color indicates a larger error for that pixel, while a darker blue color indicates a more accurate color value.

4 ADDITIONAL SELF-ATTENTION MAP VISUALIZATIONS

We choose additional examples and visualize their attention maps from Figure 13 to 14 to show the effects of our Transformer blocks.

REFERENCES

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* 36, 4 (2017), 97–1.

Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. 2016. Nonlinearly weighted first-order regression for denoising Monte Carlo renderings. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 107–117.

Antoni Buades, Bartomeu Coll, and J-M Morel. 2005. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 60–65.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7794–7803.

Bing Xu, Junfei Zhang, Rui Wang, Kun Xu, Yong-Liang Yang, Chuan Li, and Rui Tang. 2019. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.* 38, 6 (2019), 224–1.

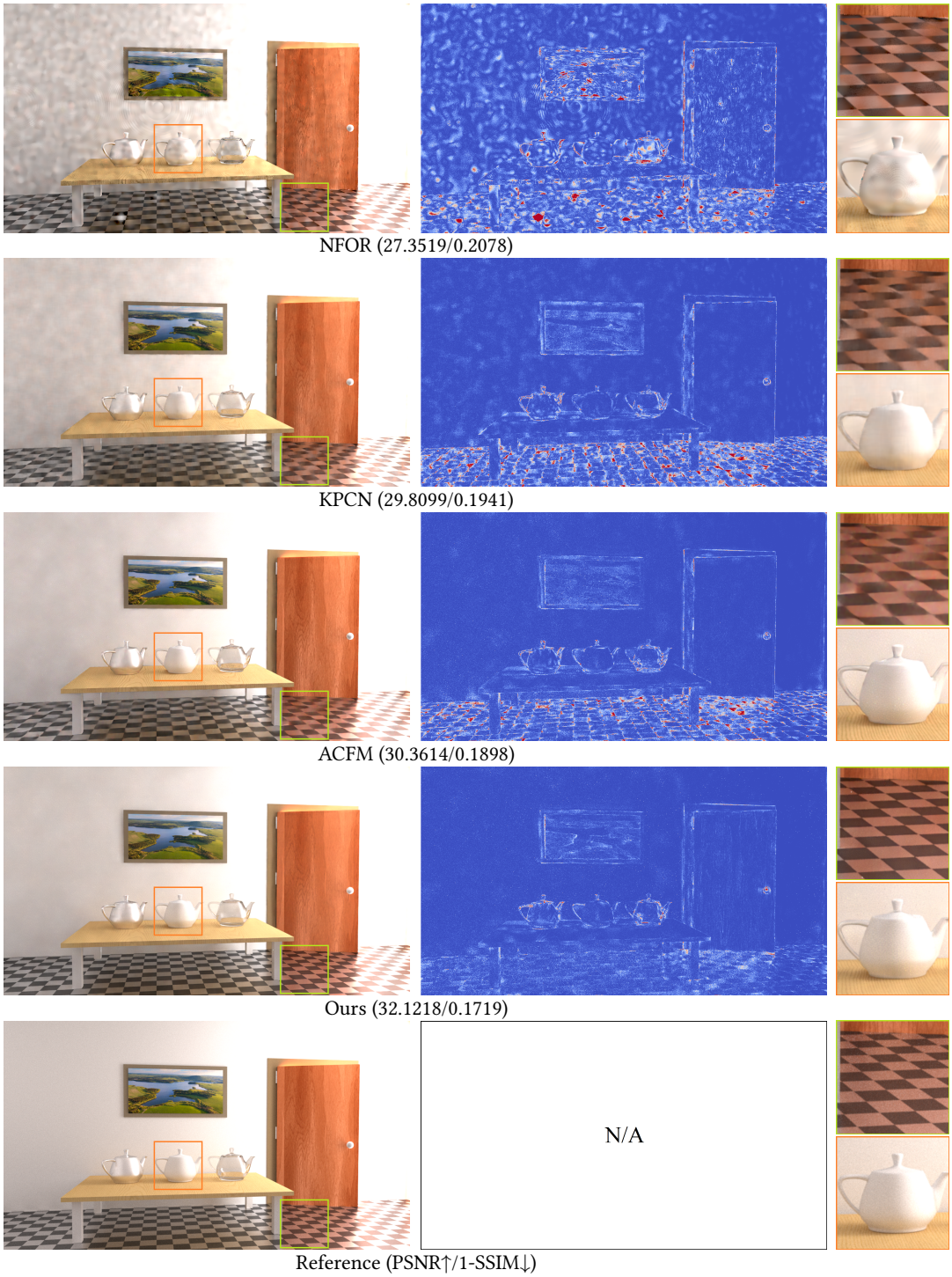


Fig. 1. Denoising results of scene "VeachAjar" with 32 spp noisy input images. Our model can better restore the details and structures.

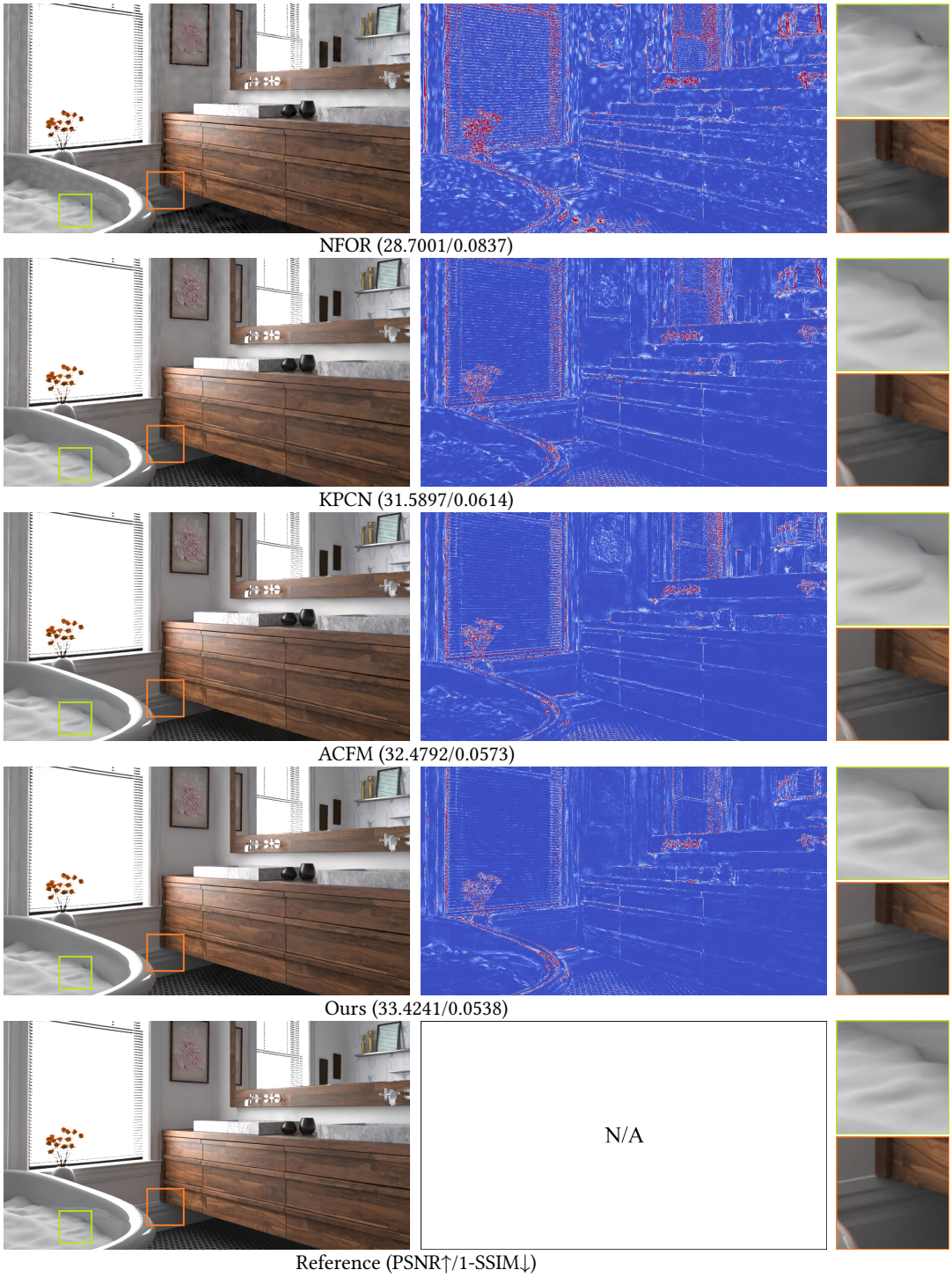


Fig. 2. Denoising results of scene "Bathroom" with 32 spp noisy input images.

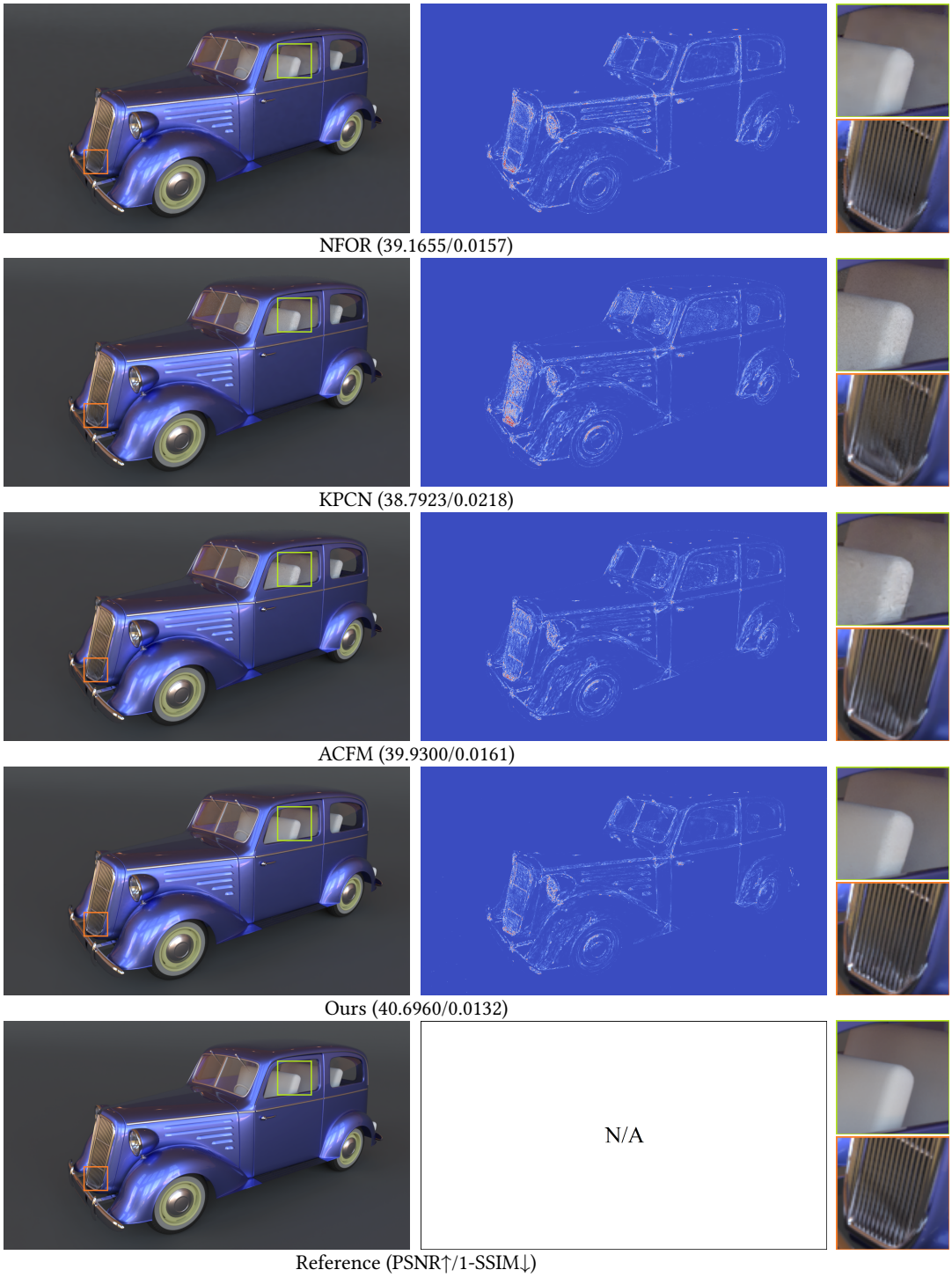


Fig. 3. Denoising results of scene "VintageCar" with 32 spp noisy input images. Our model can better restore the structures of heat sink and produce smooth seat.

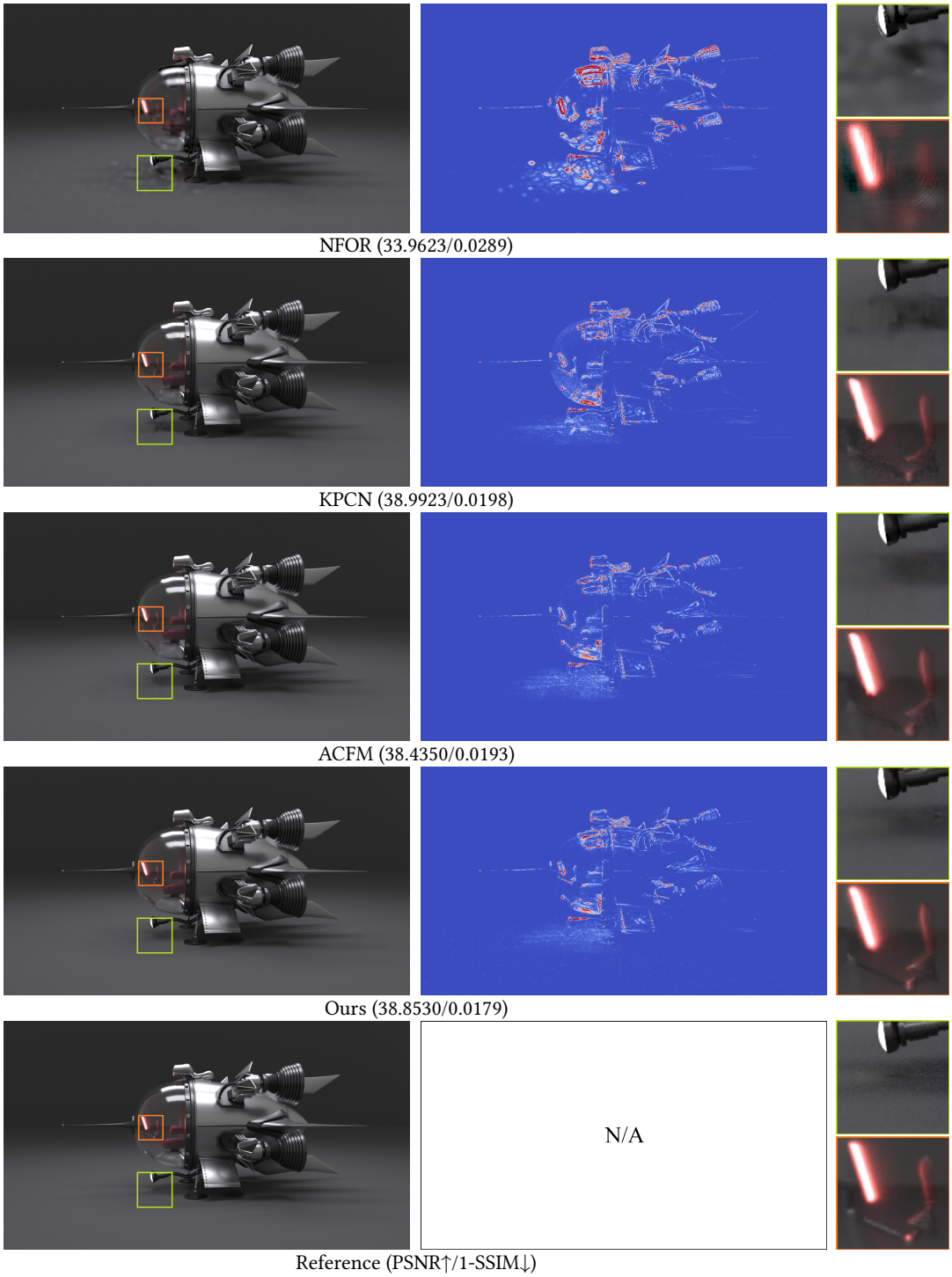


Fig. 4. Denoising results of scene "Spaceship" with 32 spp noisy input images.

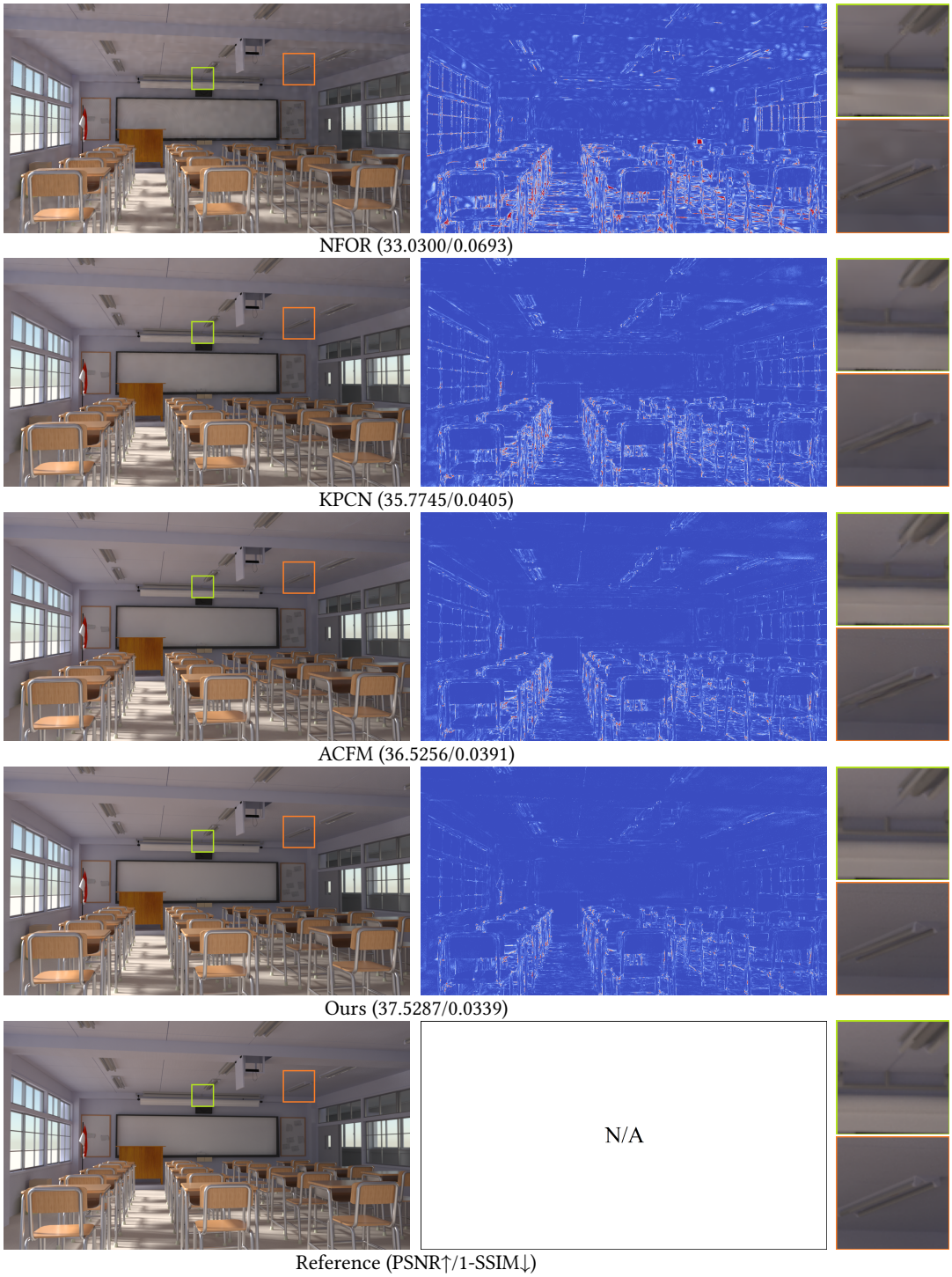


Fig. 5. Denoising results of scene "Classroom" with 32 spp noisy input images.

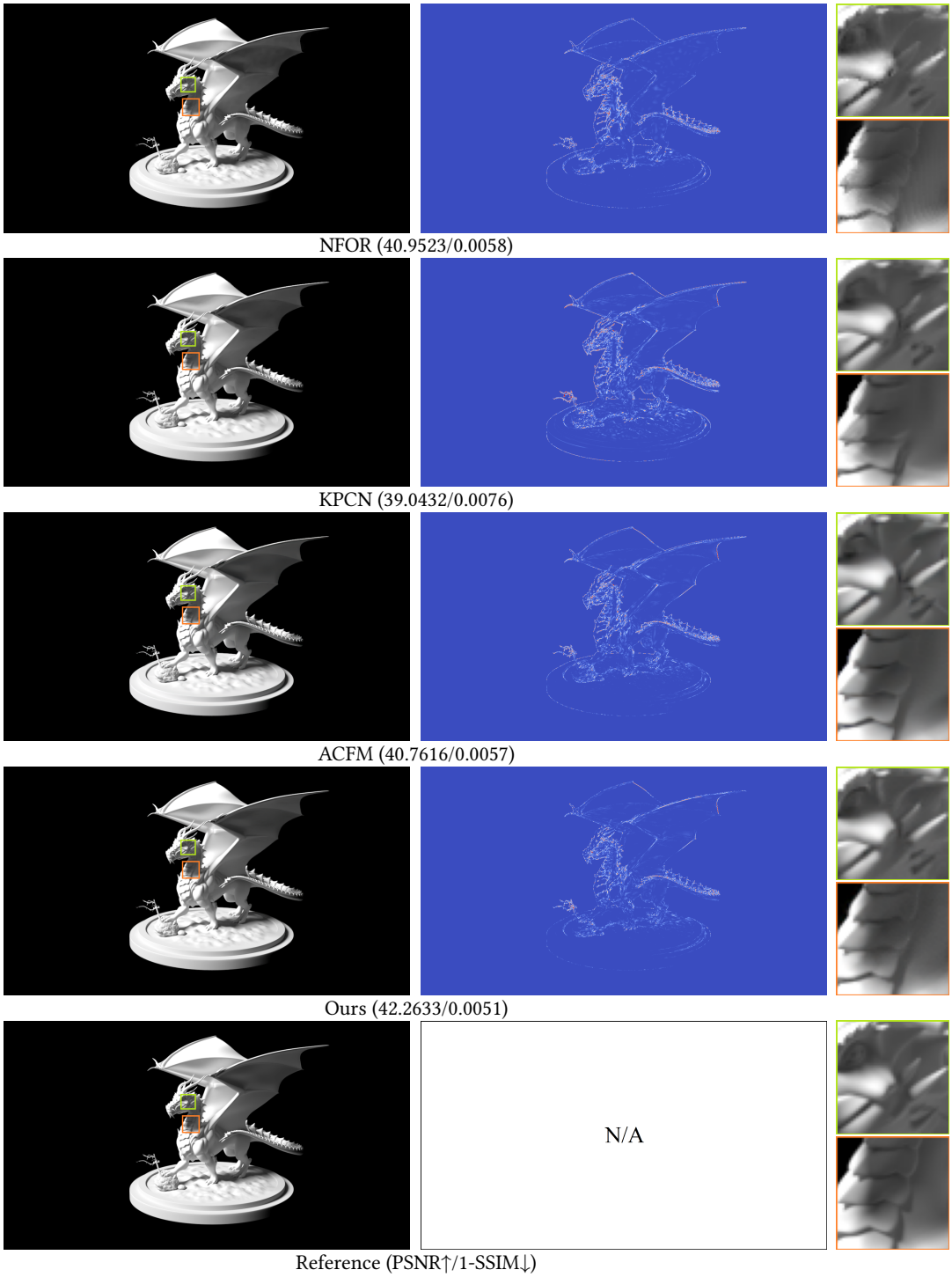
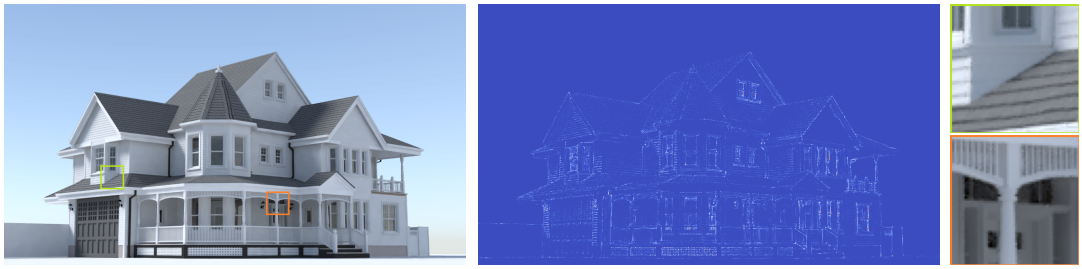
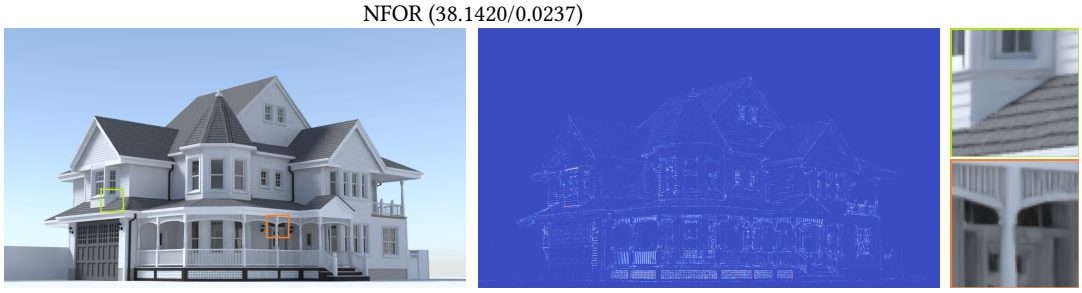
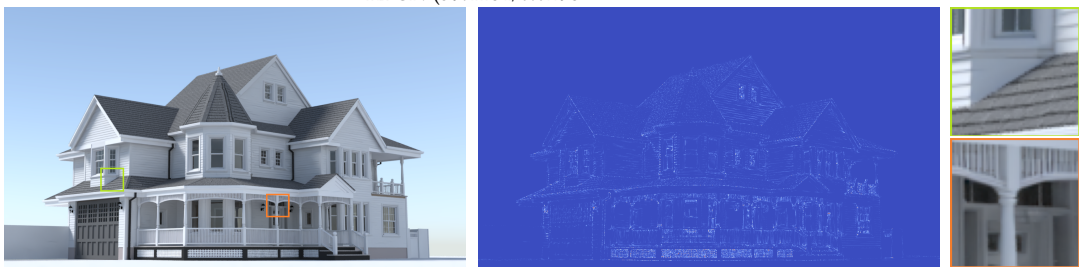
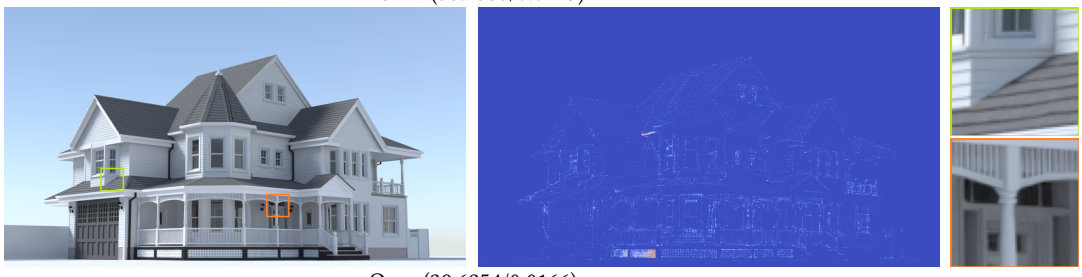


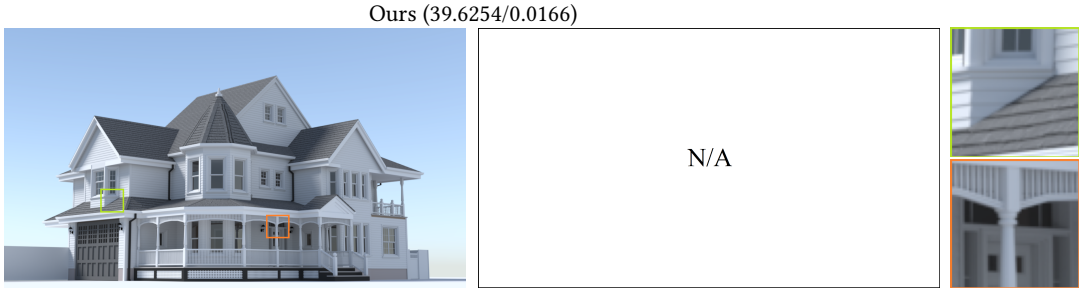
Fig. 6. Denoising results of scene "Dragon" with 32 spp noisy input images.


NFOR (38.1420/0.0237)


KPCN (37.4164/0.0276


ACFM (38.4383/0.0217)


Ours (39.6254/0.0166)



Reference (PSNR↑/1-SSIM↓)

Fig. 7. Denoising results of scene "House" with 32 spp noisy input images.

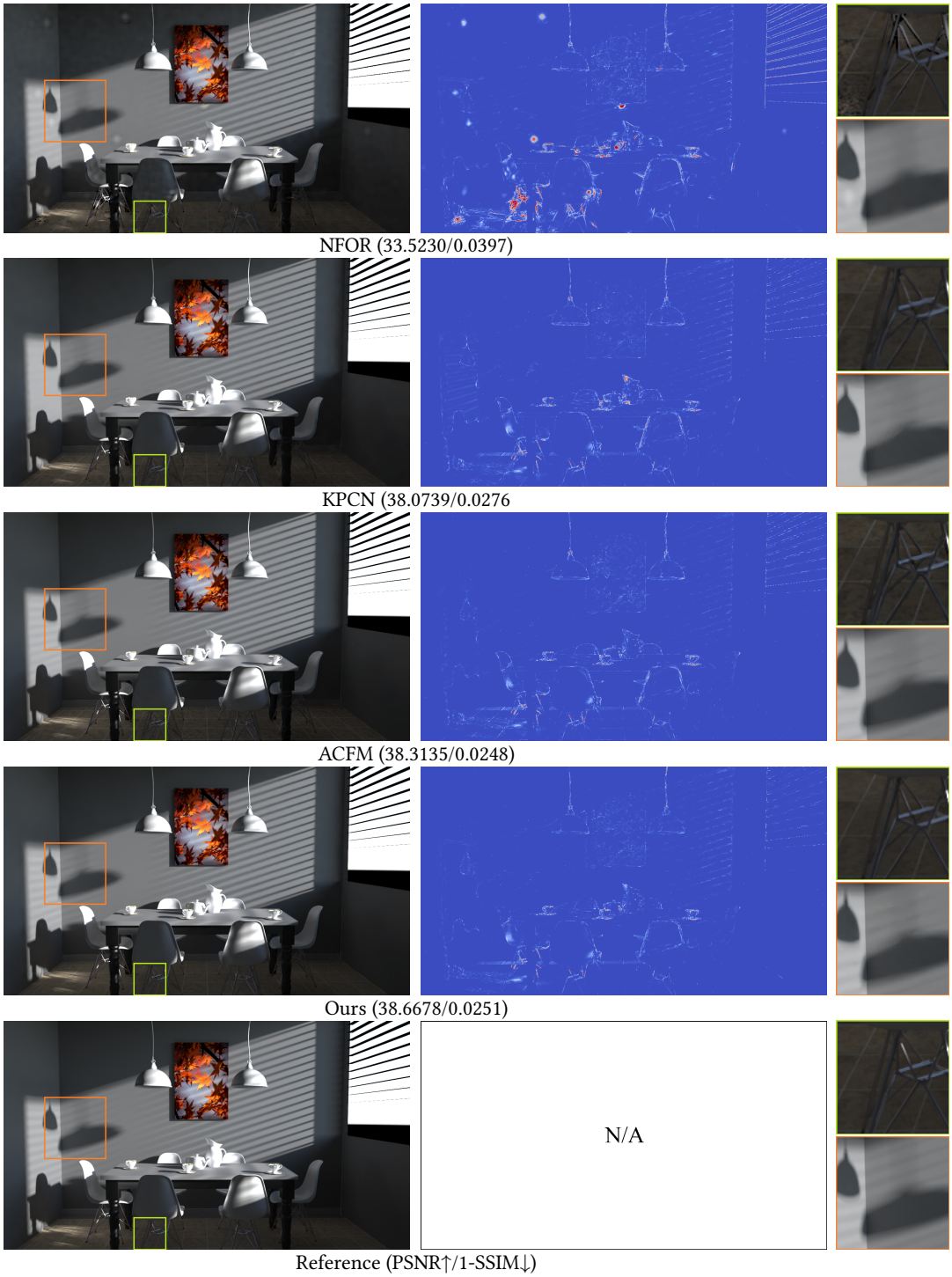
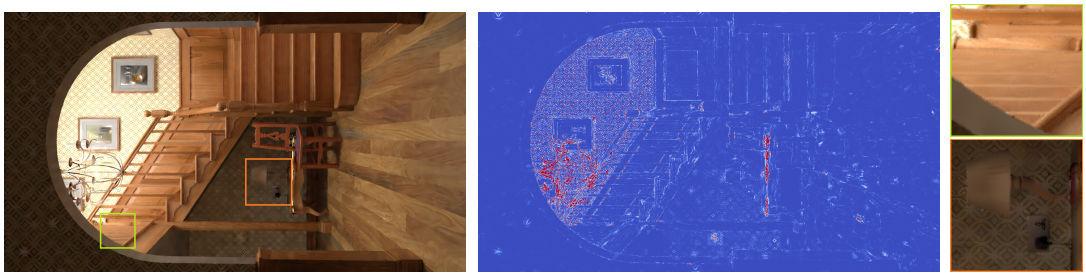
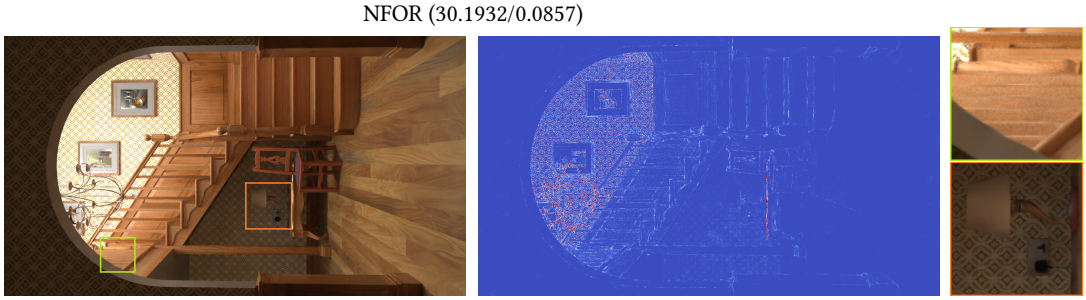


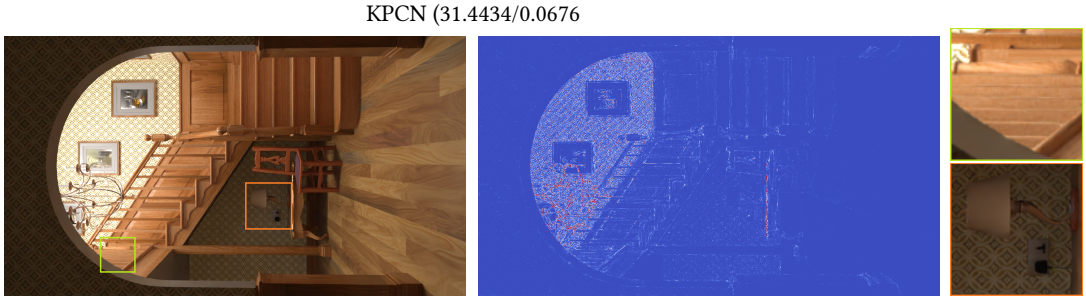
Fig. 8. Denoising results of scene "DiningRoom" with 32 spp noisy input images.



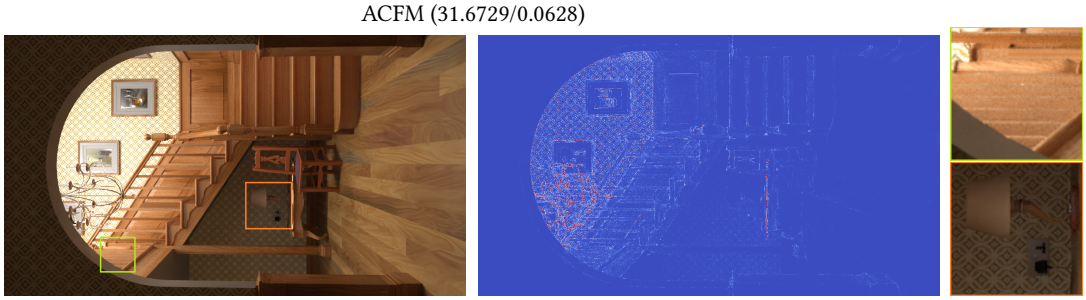
NFOR (30.1932/0.0857)



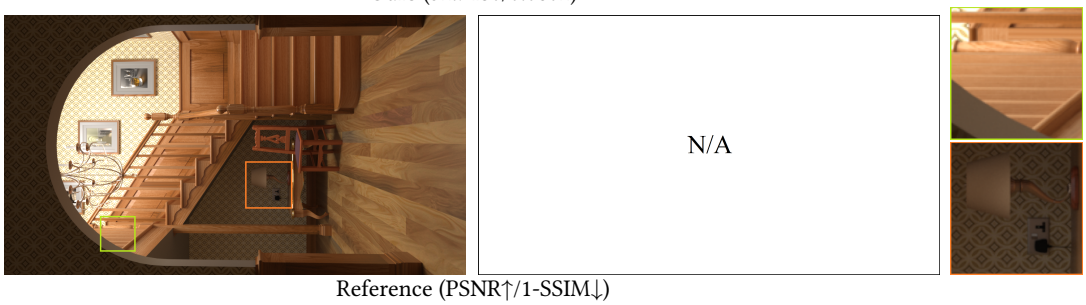
KPCN (31.4434/0.0676



ACFM (31.6729/0.0628)



Ours (32.9480/0.0602)



Reference (PSNR↑/1-SSIM↓)

Fig. 9. Denoising results of scene "Staircase" with 4 spp noisy input images.

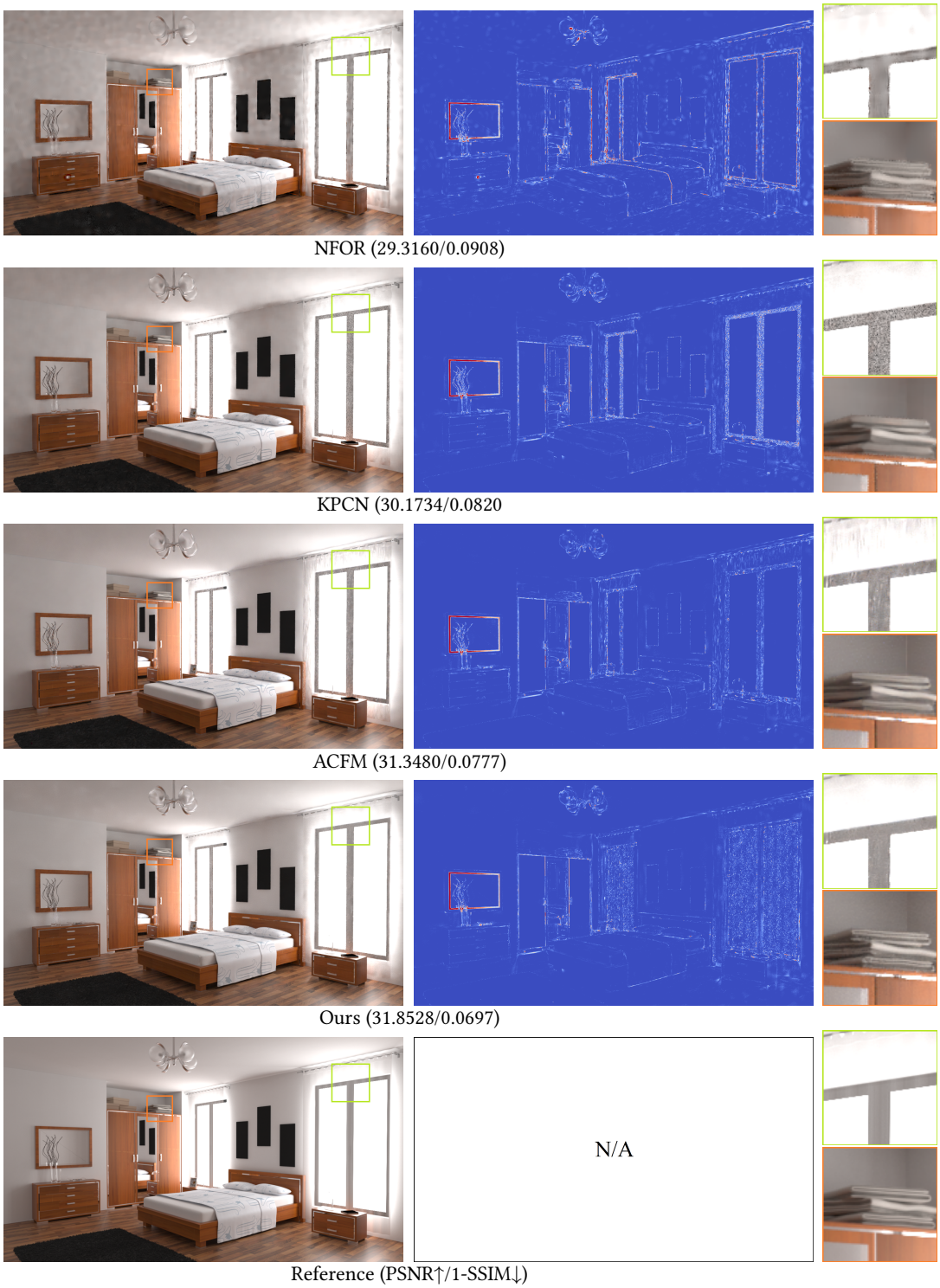


Fig. 10. Denoising results of scene "Bedroom" with 8 spp noisy input images.

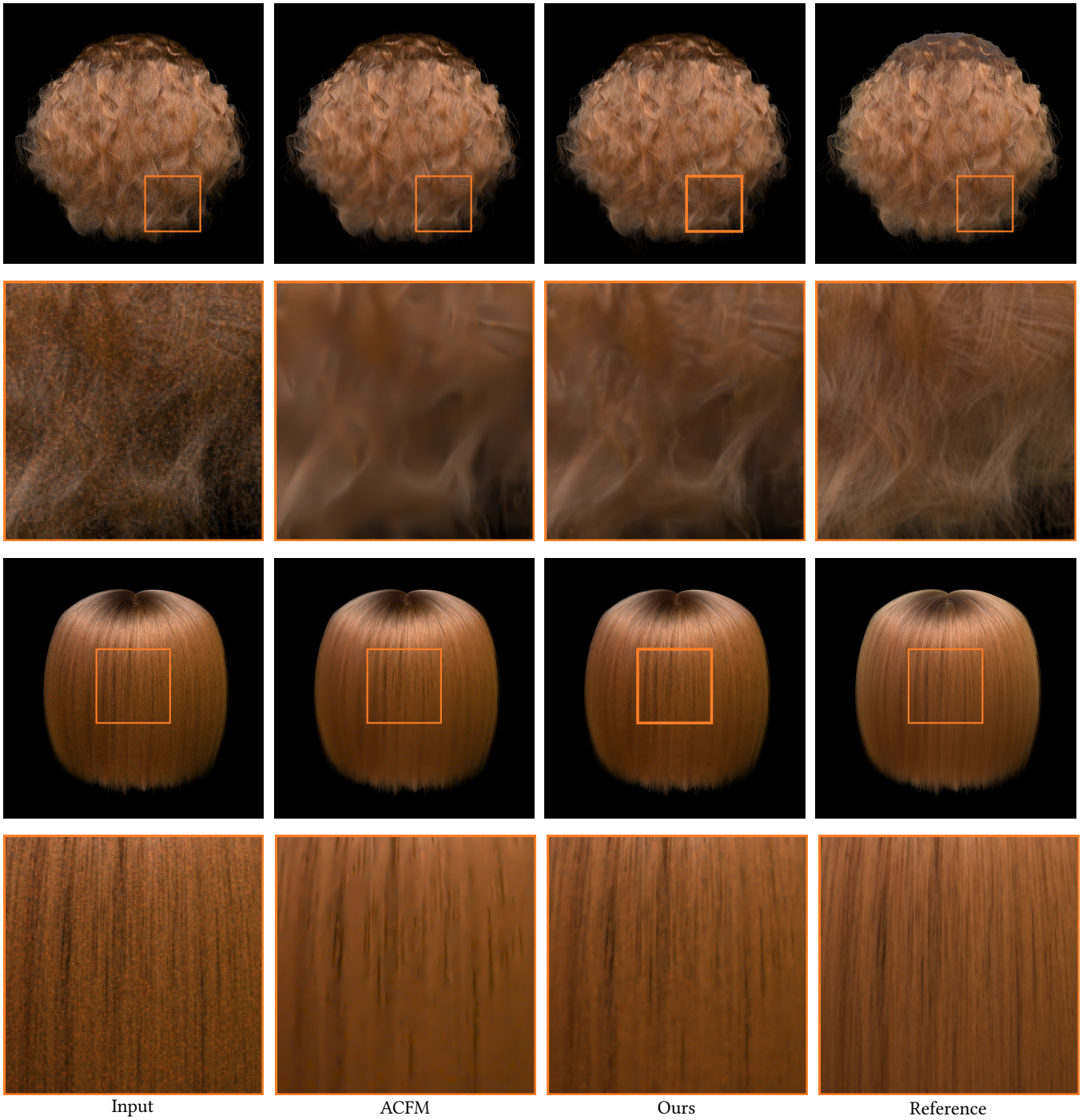


Fig. 11. Denoising results of scenes "HairCurl" and "Furball" with 32 spp noisy input images. Our method can restore more tiny structures than ACFM.

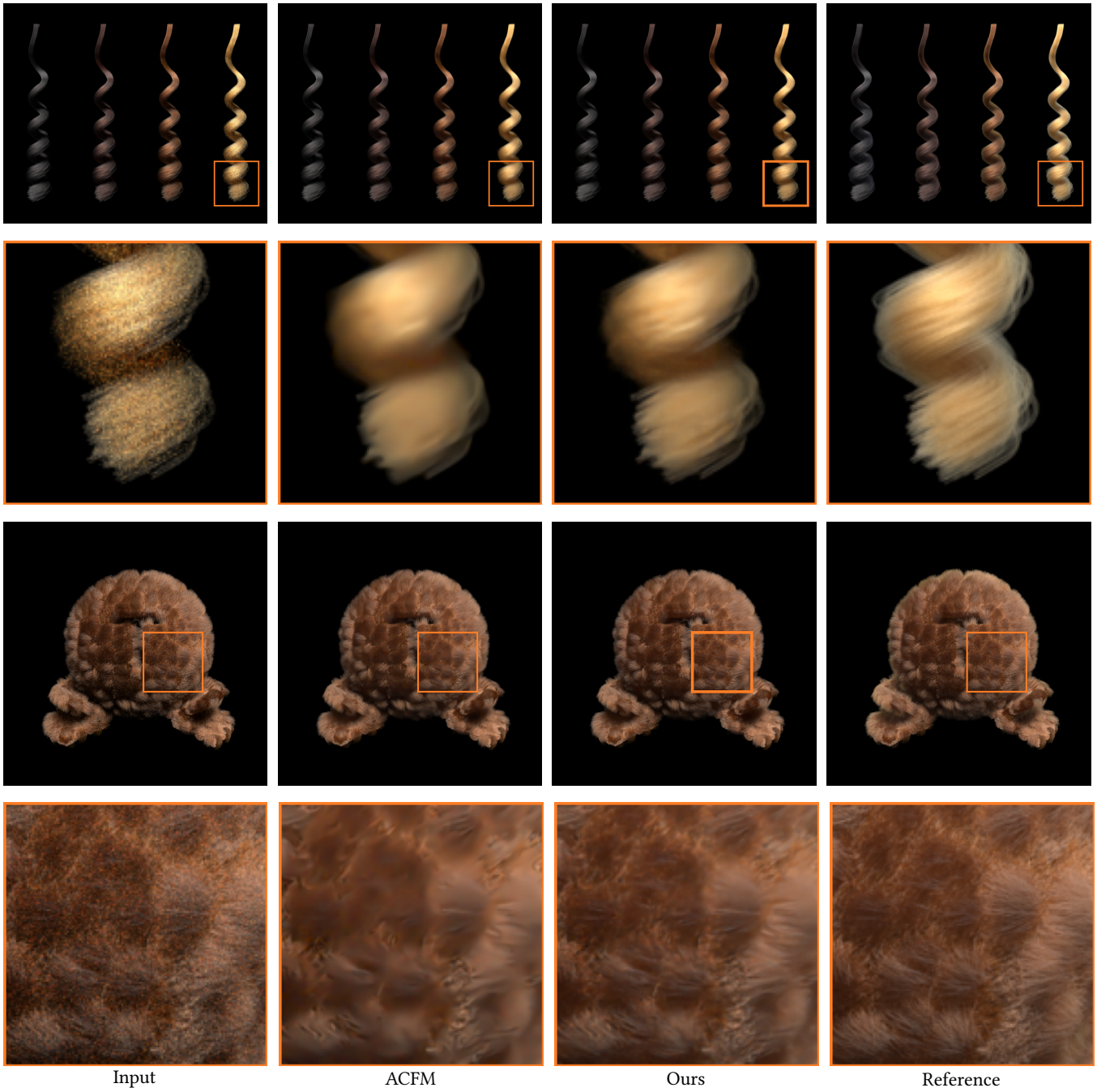


Fig. 12. Denoising results of scenes "CurlyHair" and "StraightHair" with 32 spp noisy input images. Our method can restore more tiny structures than ACFM.

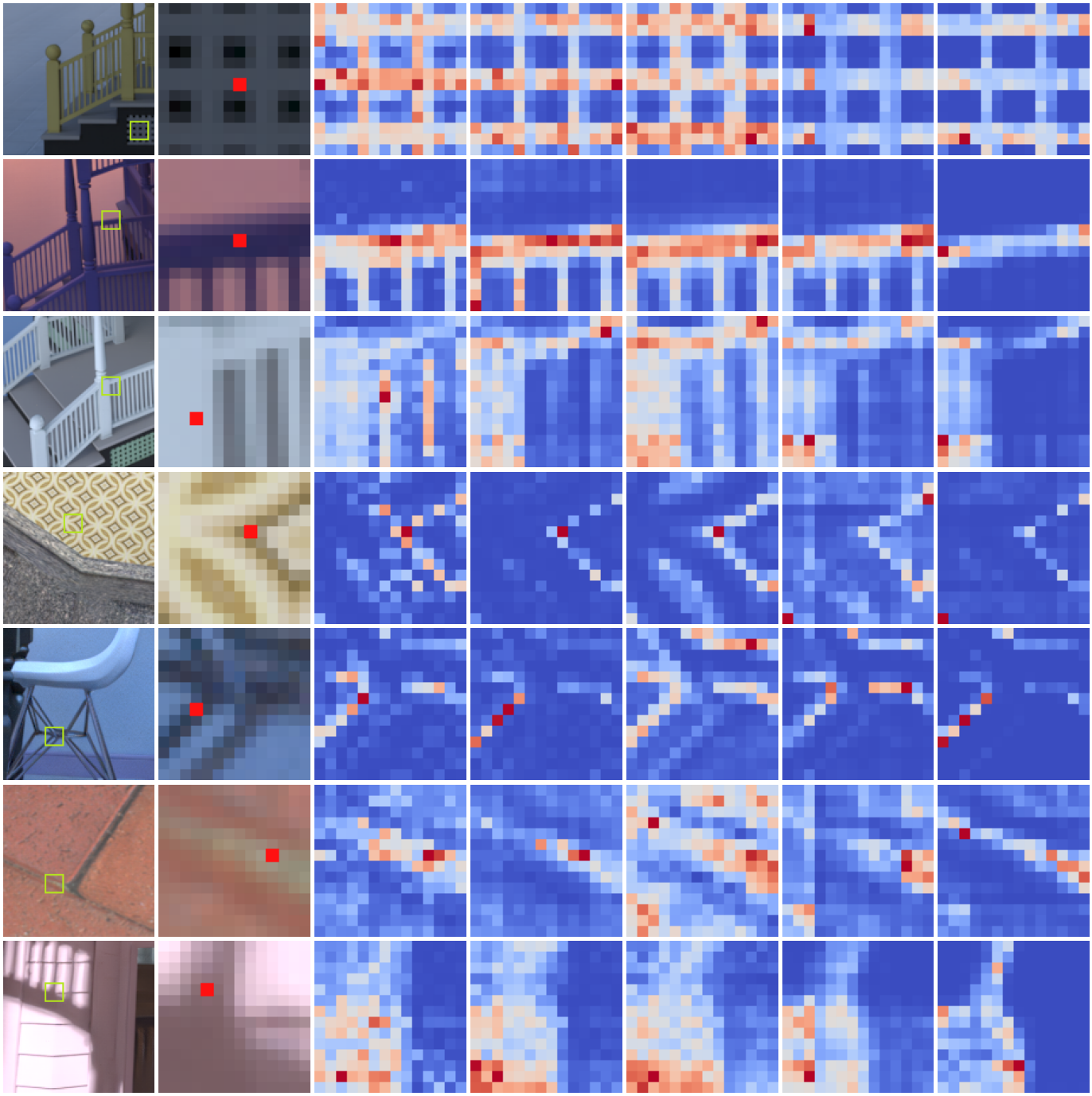


Fig. 13. From left to right: the denoised images by our model with yellow boxes specifying selected patches; zoomed-in patches with red points representing query pixels for which attention maps to be visualized; self-attention maps of all five Transformer blocks, in which a deeper red color means a higher attention score while a deeper blue color represents a lower attention score.

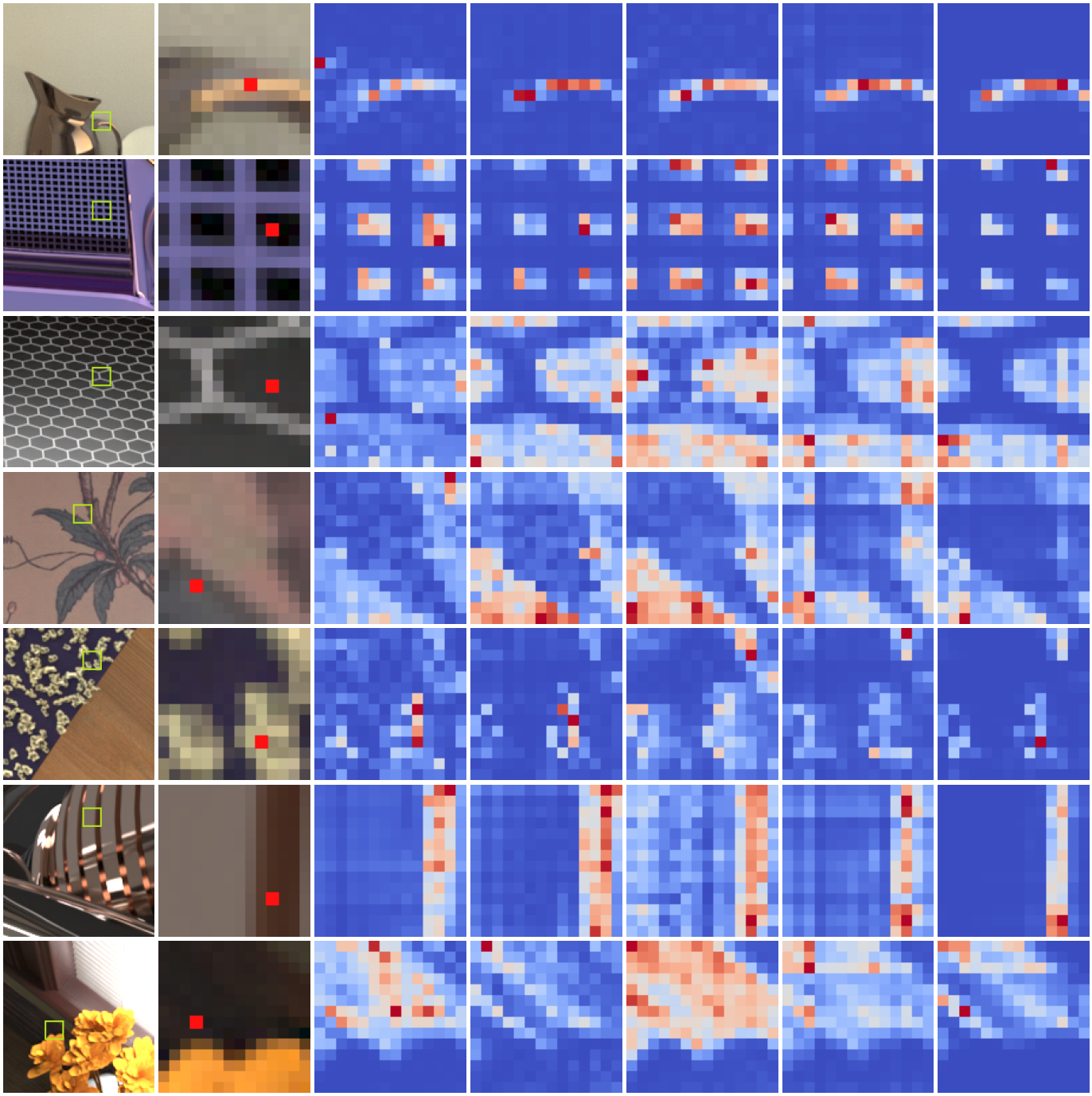


Fig. 14. From left to right: the denoised images by our model with yellow boxes specifying selected patches; zoomed-in patches with red points representing query pixels for which attention maps to be visualized; self-attention maps of all five Transformer blocks, in which a deeper red color means a higher attention score while a deeper blue color represents a lower attention score.