Supplementary Materials for "Neural Point Cloud Rendering via Multi-Plane Projection"

Peng Dai[1*] Yinda Zhang[2*] Zhuwen Li[3*] Shuaicheng Liu[1†] Bing Zeng[1]
[1]University of Electronic Science and Technology of China
[2]Google Research [3]Nuro Inc

We provide details about network architecture, evaluation details, other comparisons and results on various datasets.

1. Network Architecture

The network architecture is provided in Table. 1, including kernel size, stride, kernel dilation, activation function, and feature channels. In short, our network is a UNet with short-cut connection.

2. Evaluation Details

Our test set consists of four scenes from ScanNet [3] and two scenes from Matterport3D [2]. The scene IDs chosen from ScanNet are 'scene0000_00', 'scene0010_00', 'scene0016_00', 'scene0024_00', and the first and fourth scenes are also adopted in NPG [1]. The scene IDs chosen from Matterport3D are '17DRP5sb8fy' and '29hnd4uzFmX'.

We calculate PSNR and SSIM in standard way by using the function provided by Matlab.

3. More Results

3.1. Multi-Plane Image and Weight

For the network outputs, we produce multiple planes of images with their blending weights. Fig 1 visualizes one of the output example. As can be seen, the blending weights roughly reflects the depth of the scene since each plane corresponds to a specific depth in camera coordinate. Regarding the color image, all the images seems to maintain the scene layout reasonably well, while for each pixel the plane with the correct depth are likely to provide the right color.

3.2. Compare with image inpainting

An alternative approach for point cloud rendering is to treat it as an image inpainting problem, which renders an incomplete 2D image from the sparse point cloud and fill the missing region. To compare with this baseline, we adopt U-Net as backbone, and replace ordinary convolution operation with gate convolution (GateConv) [6] and sparse convolution (SparseConv) [5], which were proposed in SOTA image inpainting methods to fill up irregular holes. We compare our method to this image inpainting baseline, and the results are shown in Fig 2. We find that directly inpaint sparse point clouds images (Fig 2(a)) generates blurry results (Fig 2(b)(c)) compared to our results (Fig 2(d)). This is presumably because that the color of directly projected points are different with ground truth images due to view-dependent effects (e.g. specular) and different exposure time.

3.3. Textured mesh refinement

Even though choosing point cloud as intermediate is flexible and saves time from mesh reconstruction, we also compare to a mesh rendering based approach. We first render the textured mesh of the scene into the target camera viewpoint (Fig 3(a)) and utilize neural network to refine them. The comparison is shown in Fig 3(b). After refinement, the missing region in textured mesh (re-projected) can be roughly completed, but has less details than our results (Fig 3(c)). Moreover, the PSNR and SSIM of refined images (re-projected textured mesh) are 22.208 and 0.86 respectively in 'scene0010_00', and our results are 24.421 and 0.901.

3.4. Test on Creepy Attic

We test our method on Creepy Attic [4] which consists of kinect captured RGB-D and digital camera captured high quality RGB images. In Hedman et al. [4], camera parameters are generated via structure from motion (SFM), and per-view depth images are obtained using multi-view stereo (MVS). Due to the limited number of high-quality images, our method firstly train on kinect captured images, then fine-tune on digital camera captured RGB images. Fig 4 show some test results (randomly selected poses that never

*Equal contribution
†Corresponding author

Layers	kernel size	stride	dilation	channel_in	channel_out	actv	inputs
conv0	1×1×1	1×1×1	1	11	16	lrelu	neural point features
conv1	3×3×3	1×1×1	1	16	32	lrelu	conv0
maxpool1	2×2×2	2×2×2	-	32	32	-	conv1
conv2	3×3×3	1×1×1	1	32	32	lrelu	maxpool1
maxpool2	2×1×1	2×1×1	-	32	32	-	conv2
conv3	3×3×3	1×1×1	1	32	64	lrelu	maxpool2
maxpool3	2×2×2	2×2×2	-	64	64	-	conv3
conv4	3×3×3	1×1×1	1	64	64	lrelu	maxpool3
maxpool4	2×1×1	2×1×1	-	64	64	-	conv4
conv5	3×3×3	1×1×1	1	64	128	lrelu	maxpool4
maxpool5	2×2×2	2×2×2	-	128	128	-	conv5
conv6	1×3×3	1×1×1	2	128	128	lrelu	maxpool5
up1	2×2×2	2×2×2	1	128	128	-	conv6
conv7	3×3×3	1×1×1	1	256	128	lrelu	up1+conv5
up2	2×1×1	2×1×1	1	128	64	-	conv7
conv8	3×3×3	1×1×1	1	128	64	lrelu	up2+conv4
up3	2×2×2	2×2×2	1	64	64	-	conv8
conv9	3×3×3	1×1×1	1	128	64	lrelu	up3+conv3
up4	2×1×1	2×1×1	1	64	32	-	conv9
conv10	3×3×3	1×1×1	1	64	32	lrelu	up4+conv2
up5	2×2×2	2×2×2	1	32	32	-	conv10
conv11	3×3×3	1×1×1	1	64	32	lrelu	up5+conv1
rgbs	1×1×1	1×1×1	1	32	3	-	conv11
blend weights	1×1×1	1×1×1	1	32	1	softmax	conv11

Table 1. Details of network architecture, where **actv** is the activation function. And **up1-5** are transposed convolutions.

appear during training).

3.5. More Results on ScanNet

In this section, we show more results on ScanNet dataset in Fig 5 and Fig 6. Our method is better than pix2pix in terms of the visual quality. Compared to NPG [1], we perform especially better at the location with sparsity and occlusion.

3.6. More Results on Matterport3D

We also show more results on Matterport3D dataset in Fig 7 and Fig 8. Our results are significantly better than all the other methods. Our camera frustum based point projection is more robust compared to z-buffer based projection [1] when the point cloud is sparse, which is especially true on large scenes from Matterport3D. As a result, our renders are more complete and reflect correct occlusion.

3.7. Video for Temporal Consistency

To demonstrate the temporal coherence, we provide video sequences on the same trajectories from NPG and our method. We compare video temporal consistency synthesized by different methods (e.g. ours, pix2pix, neural point based graphic, direct render.) in different scenes.

References

[1] Kara-Ali Aliev, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2019. 1, 2

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 1

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. CVPR*, pages 5828–5839, 2017. 1

[4] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Trans. Graphics*, 35(6):1–11, 2016. 1

[5] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 1

[6] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proc. CVPR*, pages 4471–4480, 2019. 1

Figure 1. One example of multi-plane RGB images and blending weights. Left to right, up to down.

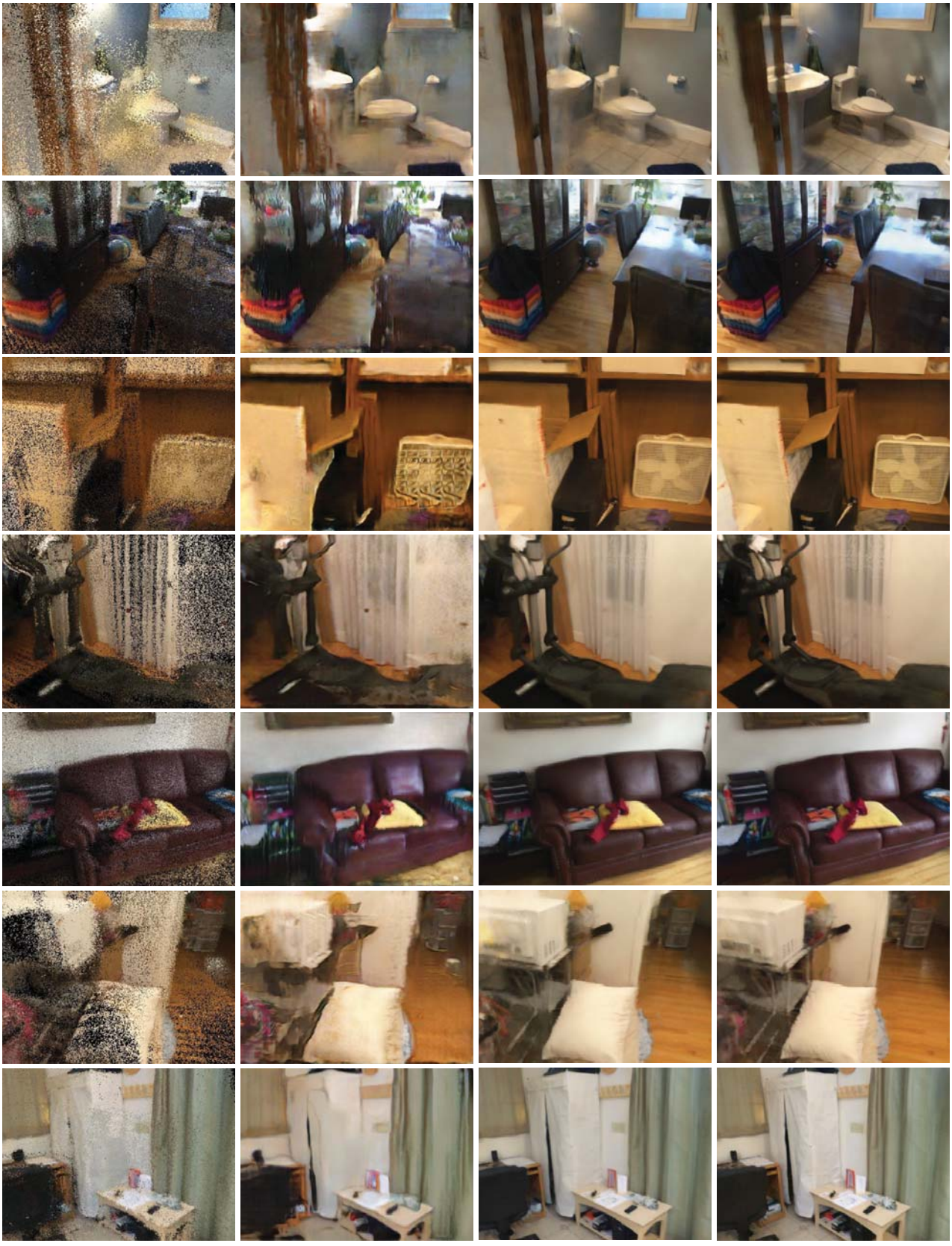
(a) Point clouds (b) GateConv (c) SparseConv (d) Our Results

Figure 2. Compare with image inpainting. (a) Re-projected 2D point clouds images with holes. (b) Results generated by using gate convolution. (c) Results generated by using sparse convolution. (d) Results generated by our method. Directly apply image inpainting will generate blurry results. (Zoom in for details)

(a) Mesh+Texture (b) Mesh+Texture (Refined) (c) Our Results

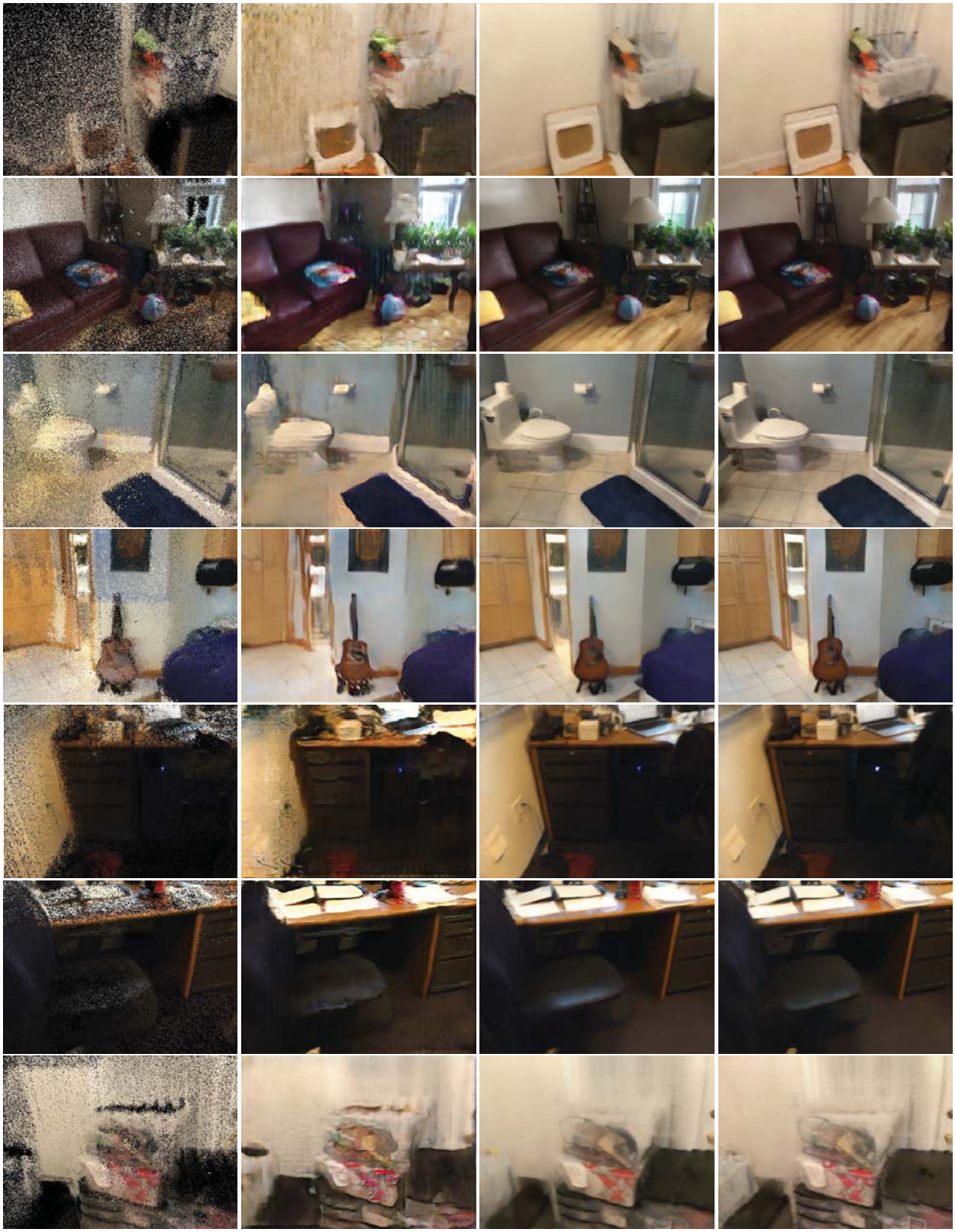
Figure 3. Compare with textured mesh refinement. (a) Re-project textured mesh into 2D image plane. (b) Refined result through neural network. (c) Result generated by our method. The missing region in (a) can be roughly completed through network, but the refined image (b) has less details compared with our results (c). (Zoom in for detail)

Figure 4. Test results(second row) on Creepy Attic with corresponding colored point clouds(first row).



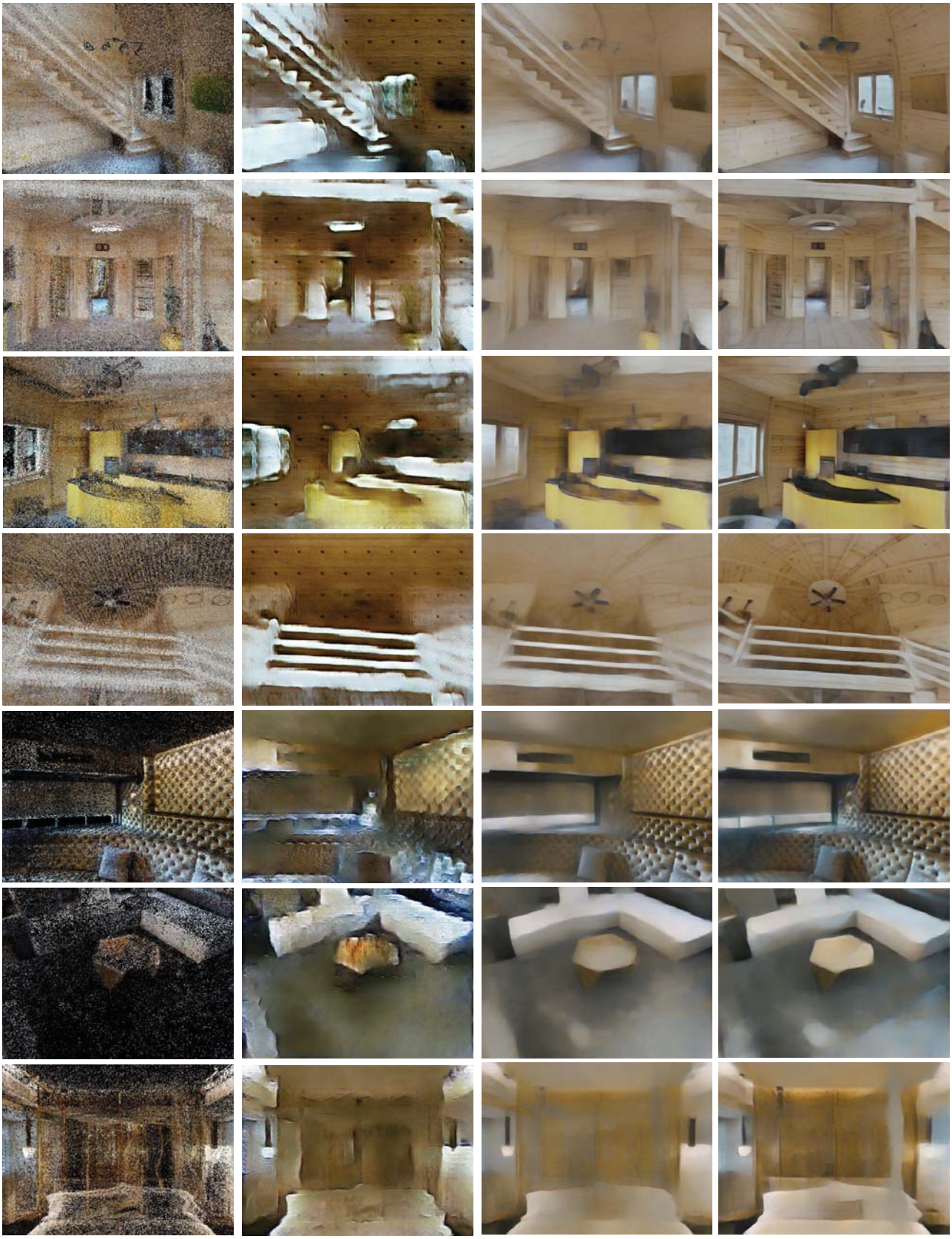
(a) Point clouds (b) pix2pix (c) NPG (d) Our results

Figure 5. More results on ScanNet. Our results are better than other methods, especially at the location with sparsity and occlusion. Zoom in for details.



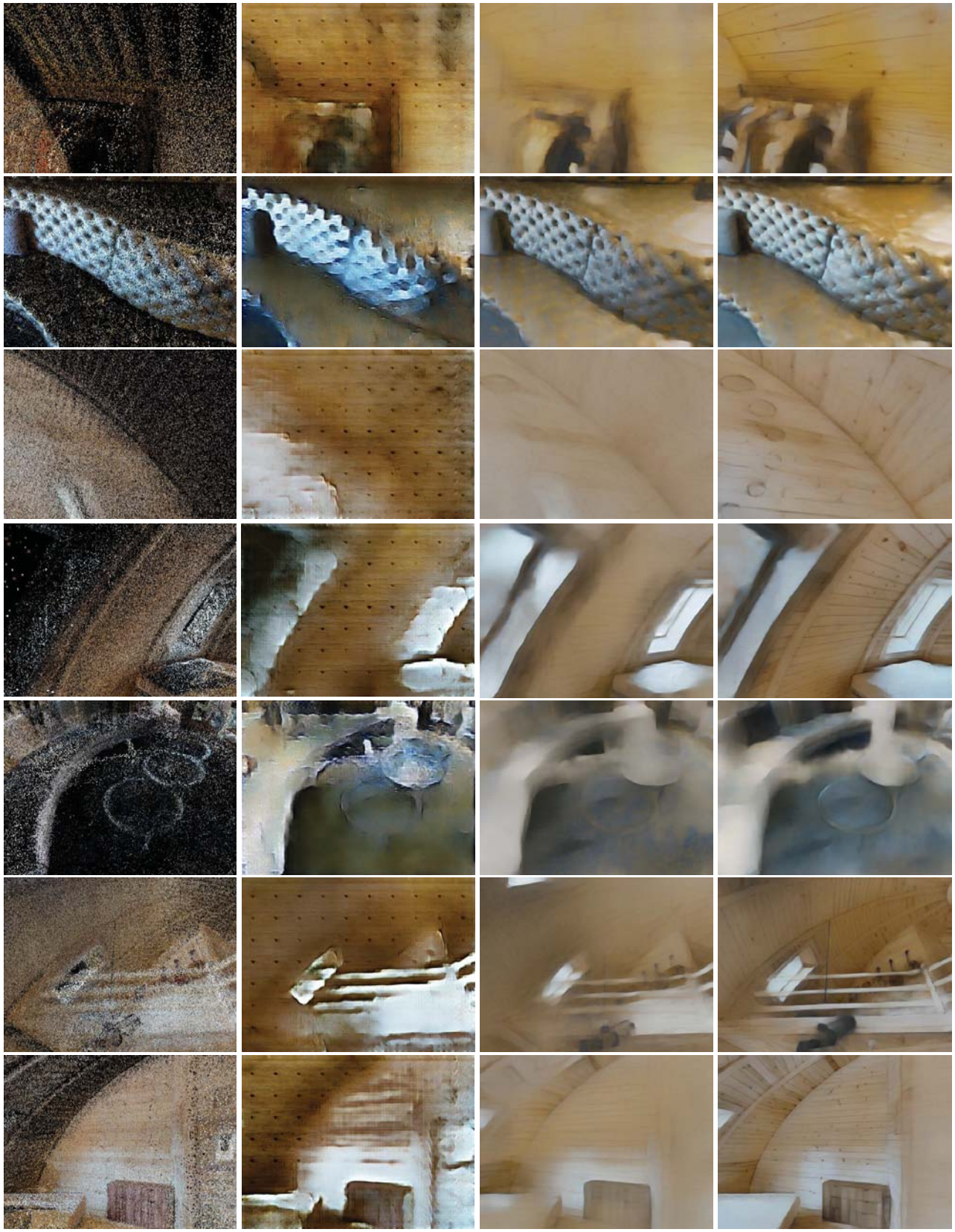
(a) Point clouds (b) pix2pix (c) NPG (d) Our results

Figure 6. More results on ScanNet. Our results are better than other methods, especially at the location with sparsity and occlusion. Zoom in for details.



(a) Point clouds (b) pix2pix (c) NPG (d) Our results

Figure 7. More results on Matterport3D. Our method obtains higher performance than other baselines on this challenging dataset.



(a) Point clouds (b) pix2pix (c) NPG (d) Our results

Figure 8. More results on Matterport3D. Our method obtains higher performance than other baselines on this challenging dataset.