The Supplementary of "Improving Neural Surface Reconstruction with Feature Priors from Multi-View Images"

Xinlin Ren[1], Chenjie Cao[1,2], Yanwei Fu[1], and Xiangyang Xue[1]

[1] Fudan University
[2] Alibaba Group, DAMO Academy
{xlren20,cjcao20,yanweifu,xyxue}@fudan.edu.cn

1 Details of Patch-wise Consistent Loss

Normalization Cross Correlation (NCC). We can use NCC to measure the difference between patches of source view and reference view at the feature level:

$$NCC(\mathbf{F}_r(\hat{s}), \mathbf{F}_s(\hat{s}')) = \text{mean}\left(\frac{\text{Cov}(\mathbf{F}_r(\hat{s}), \mathbf{F}_s(\hat{s}'))}{\sqrt{\text{Var}(\mathbf{F}_r(\hat{s})\text{Var}(\mathbf{F}_s(\hat{s}')))}}\right), \tag{1}$$

Where Cov and Var represent feature covariance and variance, respectively. $\hat{s}$ and $\hat{s}'$ denote the centers of reference and projected source patches[3], respectively. $\mathbf{F}_r$ and $\mathbf{F}_s$ are the features of reference and source view. Finally, we average all channel-wise feature correlations to the feature-level NCC.

Structural Similarity (SSIM) evaluates the local structural information in the images by comparing small patches of pixels. SSIM can also be applied to assess the difference between patches of the source view and reference view at the feature level:

$$SSIM(\mathbf{F}_r(\hat{s}), \mathbf{F}_s(\hat{s}')) = \text{mean}\left(\frac{(2\mu_{\mathbf{F}_r}\mu_{\mathbf{F}_s} + c_1)(2\sigma_{\mathbf{F}_r\mathbf{F}_s} + c_2)}{(\mu^2_{\mathbf{F}_r} + \mu^2_{\mathbf{F}_s} + c_1)(\sigma^2_{\mathbf{F}_r} + \sigma^2_{\mathbf{F}_s} + c_2)}\right), \tag{2}$$

where $\mu_{\mathbf{F}_r}$, $\mu_{\mathbf{F}_s}$ denote the mean of $\mathbf{F}_r(\hat{s})$ and $\mathbf{F}_s(\hat{s}')$. $\sigma_{\mathbf{F}_r}$, $\sigma_{\mathbf{F}_s}$ are the variance of these features, respectively. $\sigma_{\mathbf{F}_r\mathbf{F}_s}$ represents the covariance of $\mathbf{F}_r(\hat{s})$ and $\mathbf{F}_s(\hat{s}')$. The constants c_1 and c_2 are set to 0.01 and 0.03, respectively, as suggested in [8]. We average all channel-wise feature similarities as the SSIM.

Patch Similarity (PS). Similar to the pixel similarity, we use the average of pixel similarity between corresponding elements in the two patches to represent the similarity of two patches as:

$$PS(\mathbf{F}_r(\hat{s}), \mathbf{F}_s(\hat{s}')) = \text{mean}\left(\frac{\mathbf{F}_r(\hat{s})\mathbf{F}_s(\hat{s}')}{|\mathbf{F}_r(\hat{s})|\ |\mathbf{F}_s(\hat{s}')|}\right). \tag{3}$$

[3] For simplicity, we use $\hat{s}$ and $\hat{s}'$ to present reference and projected source patches centered at $\hat{s}$, $\hat{s}'$, *i.e.*, these patches are warped by the zero-cross distance of the related center points.

Patch-wise Consistent Loss. For the patch-wise consistent loss, we dynamically optimize each patch on 4 different source views with the lowest losses which are selected from 10 candidates. This strategy is a simple yet effective way to mitigate occlusion as:

$$\mathcal{L}_{patch} = \frac{\sum_{x=1}^{N} \sum_{s=1}^{4} 1 - D_{patch}(\mathbf{F}_r(x), \mathbf{F}_s(x'))}{4N}, \tag{4}$$

where x and x' denote the pixels and their corresponding pixels in the reference and source patches centered at $\hat{s}$ and $\hat{s}'$ respectively; and N is the number of pixels in these patches. Besides, D_{patch} can be represented as NCC, SSIM, and PS. Additionally, we use 11×11 patch size to compute the consistent loss.

Table 1: Quantitative results on the subset of DTU dataset. Features from pre-trained models are based on the **lowest** resolution.

Model	Resolution	case24	case37	case106	Mean
MAE [9]	$1024 \times 32 \times 48$	1.21	1.32	1.37	1.30
ConvMAE [6]	$768 \times 32 \times 48$	0.98	2.05	0.86	1.29
CroCo [19]	$768 \times 32 \times 48$	1.19	2.05	0.82	1.35
Twins [4]	$1024 \times 18 \times 24$	1.00	2.40	0.87	1.42
ConvNeXt [15]	$2048 \times 18 \times 24$	1.22	2.30	1.48	1.66
SegFormer [20]	$512 \times 18 \times 24$	0.99	2.44	0.97	1.46
GLPN [11]	$512 \times 18 \times 24$	0.97	1.31	0.88	1.05
MiDaS [16]	$2048 \times 16 \times 24$	1.21	2.05	1.42	1.56
IGEV [21]	$160 \times 18 \times 24$	1.19	2.35	1.18	1.57
QuadTree [17]	$256 \times 72 \times 96$	0.74	1.39	0.67	0.93
CascadeMVS [7]	$32 \times 256 \times 384$	0.79	1.24	1.00	1.01
MVSFormer [2]	$32 \times 256 \times 384$	**0.65**	**1.00**	**0.63**	**0.76**
Baseline	/	1.37	1.21	0.66	1.08

2 Implementation Details

For a fair comparison, we adopt the same architecture as the current works [3, 5, 18]. The geometry network consists of an 8-layer MLP with 256 hidden units, which is initialized using the geometric initialization proposed in [1]. And it outputs SDF and a 256-dimension feature vector which will be concatenated with normal, view direction and 3D point to regress color. The radiance net has a 4-layer MLP with 256 hidden units, and positional encoding with 6 and 4 frequencies is applied to encode the 3D position and view direction, respectively. Additionally, we sample 512 rays per batch which follow the hierarchical sampling strategy in NeuS [18] where each ray has 64 coarse and 64 fine sampled points. Besides, the background image is modeled using 32 points outside the unit sphere, following the approach of Nerf++ [22]. We train the network for 300k iterations costing 10 hours on a single A6000.

Table 2: Quantitative results on the subset of DTU dataset. Features from pre-trained models are based on the **highest** resolution.

Model	Resolution	case24	case37	case106	Mean
ConvMAE [6]	$256 \times 128 \times 192$	0.99	1.68	0.81	1.16
Twins [4]	$128 \times 128 \times 192$	0.89	1.52	0.87	1.09
ConvNeXt [15]	$256 \times 128 \times 192$	0.73	1.45	0.79	0.99
SegFormer [20]	$64 \times 128 \times 192$	0.83	1.25	0.88	0.98
GLPN [11]	$64 \times 128 \times 192$	0.90	1.32	0.81	1.01
IGEV [21]	$96 \times 128 \times 192$	0.88	1.23	0.76	0.95
RAFT-Stereo [14]	$256 \times 128 \times 192$	0.59	0.92	0.66	0.72
QuadTree [17]	$128 \times 256 \times 384$	0.48	0.85	0.55	0.62
CascadeMVS [7]	$8 \times 512 \times 768$	0.46	0.75	0.55	0.59
MVSFormer [2]	$8 \times 512 \times 768$	**0.40**	**0.65**	**0.49**	**0.51**
Baseline	/	1.37	1.21	0.66	1.08

3 Detailed Quantitative Results for Different Pretext Tasks

We evaluate models of various pretext tasks by applying pixel similarity as the consistency loss on a subset of the DTU dataset including scan24, scan37, and scan106. These objects are relatively challenging due to difficult lighting conditions and occlusions. Tab. 1 and Tab. 2 show the lowest and highest features of various models, respectively.

4 Additional Results of Feature Priors on Grid-Based Representation

We incorporate feature priors into grid-based Neuralangelo [13] on DTU [10] and Tanks&Temples [12] as in Tab. 3 and Fig. 1.

Table 3: Quantitative results of Neuralangelo with feature priors on DTU dataset.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
Neuralangelo	0.50	1.03	0.83	0.47	2.93	0.61	2.49	2.72	3.31	2.87	1.69	2.98	0.77	4.16	2.80	2.01
+ MVS Feat	0.52	**0.82**	**0.46**	**0.38**	**1.60**	0.63	**1.15**	**1.56**	**1.89**	**0.85**	**0.70**	**0.61**	**0.30**	**1.23**	**0.85**	**0.90**

4.1 Quantitative results of Feature Priors on DTU dataset

We train Neuralangelo with MVSFormer feature priors on the DTU dataset for 300k iterations. As indicated in Tab. 3, the integration of MVSFormer features results in a reduction in errors.

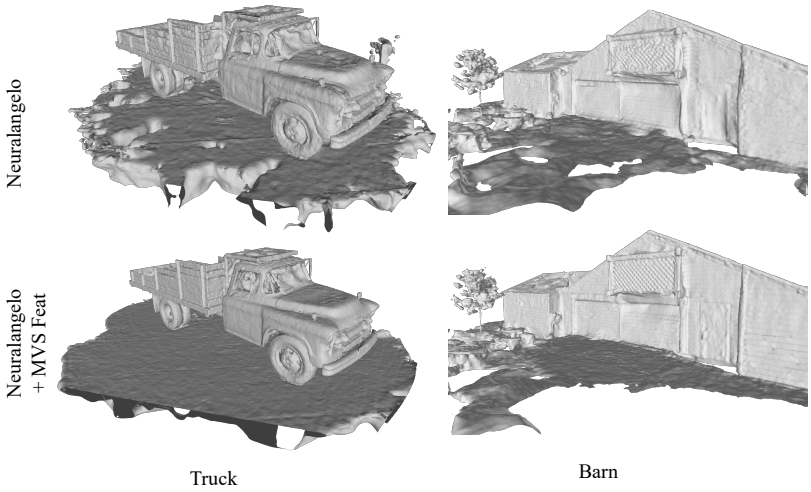

Fig. 1: Qualitative results of Neuralangelo with feature priors on Tanks&Temples dataset.

4.2 Qualitative Results of Feature Priors on Outdoor Scenes

We train Neuralangelo with MVSFormer feature priors on the Tanks&Temples dataset for 300k iterations. From Fig. 1, Neuralangelo with feature priors results in a smoother surface and more accurate geometry.

4.3 Efficiency of Feature Priors

We present the outcomes of Neuralangelo incorporating feature priors on Scan37 across various training iterations in Fig. 2. Remarkably, the results achieved by Neuralangelo with feature priors at 150k iterations surpass the accuracy of Neuralangelo without feature priors even after 300k iterations. This underscores the significant impact of feature priors in enhancing network convergence.

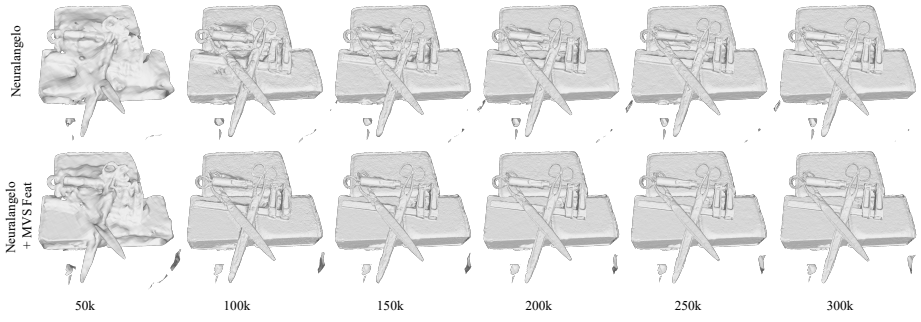

Fig. 2: Qualitative results with feature priors Scan37 at different training iterations.

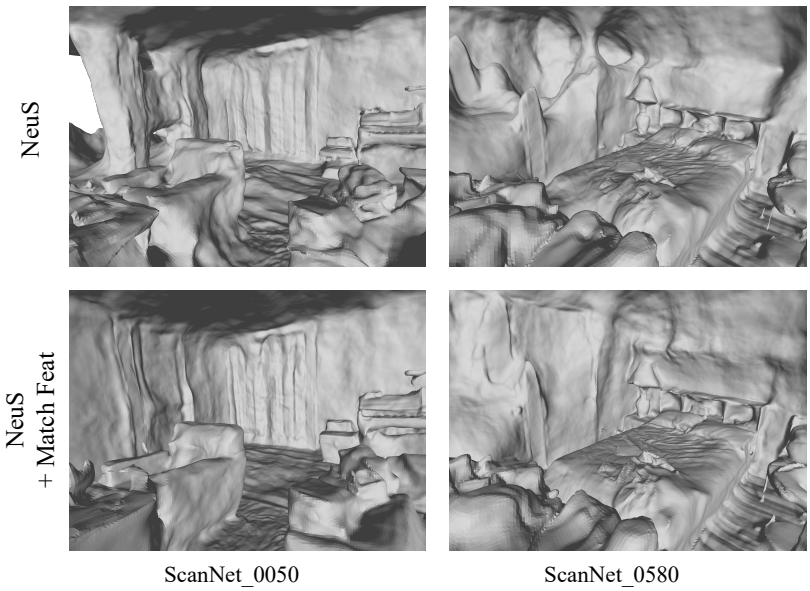

Fig. 3: Qualitative results of NeuS with feature priors on ScanNet dataset.

5 Results of Feature Priors on Indoor Scenes

We provide more qualitative results on ScanNet of NeuS [18] with QuadTree [17] Feature in Fig. 3. Notably, NeuS with match features can reconstruct more accurate geometry than NeuS, verifying that our method can be also generalized to indoor scenes.

6 More Analysis

6.1 GPU Memory Consumption

We show the GPU memory cost of our Match-NeuS and MVS-NeuS in Tab. 4. To improve efficiency, we save all off-the-shelf features in the GPU at once, avoiding costly GPU and CPU memory swappings. Compared with the baseline, this strategy costs extra GPU memory consumption. But we should clarify that these memory costs are optional to accelerate the training phase without frequently loading features into the GPU devices.

6.2 The Effect of Patch Size

In our paper, we use 11×11 patch size as default to compute the consistent loss. Here we show the performance of NCC with different patch sizes on Scan37 of the DTU dataset. The result in Tab. 5 shows that using NCC with patch sizes that are too small results in poor performance. Conversely, using too large patch sizes may lead to slightly inferior results. The best performance is achieved with

Table 4: GPU memory consumption on DTU dataset. † denotes saving all features in the GPU beforehand.

Model	Pre-Trained	Consistent Loss	GPU Memory (MB)	GPU Memory† (MB)
Baseline	/	/	7033	7033
Match-NeuS	QuadTree [17]	Pixel Similarity	8415	18999
		Patch Similarity	13823	24401
		Patch NCC	14097	24683
		Patch SSIM	14373	24955
MVS-NeuS	MVSFormer [2]	Pixel Similarity	7443	10089
		Patch Similarity	8126	10724
		Patch NCC	8303	10951
		Patch SSIM	8327	10949

7×7 and 11×11 patches. It is reasonable that a small patch size suffers from the local minimum while too large patch sizes can't capture detailed geometry accurately.

Table 5: Quantitative results of NCC with different patch sizes on scan37 of DTU Dataset.

Model	Pre-Trained	Patch Size	Mean
Baseline	/	/	1.21
Match-NeuS	QuadTree [17]	3×3	0.906
		7×7	0.684
		11×11	0.671
		15×15	0.706
		19×19	0.713
Match-NeuS	MVSFormer [2]	3×3	0.828
		7×7	**0.602**
		11×11	0.631
		15×15	0.653
		19×19	0.678

6.3 NCC with Multi-Scale Features

We show additional quantitative results of NCC with multi-scale features on scan37 of the DTU dataset in Tab. 7. Using multi-scale features leads to worse results than only using the highest-resolution features.

6.4 Combination of MVS and Image Matching features

We show additional quantitative results of combining MVS and image matching features as mixed features in Tab. 6. The method employing patch NCC with these mixed features is denoted as MVS-Match-NeuS. From Tab. 6, we observe

that the integration of MVS and image matching features does not consistently enhance performance. Moreover, it incurs a higher memory cost to store all these features in the GPU as illustrated in Sec. 6.1.

Table 6: Results of CD (mm) compared on DTU dataset (lower is better). The best results are in **bold**.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
MVS-NeuS	**0.37**	**0.63**	**0.31**	**0.32**	**0.79**	**0.51**	0.50	**1.12**	**0.87**	0.62	0.46	**0.64**	**0.29**	**0.38**	**0.41**	**0.548**
Match-NeuS	0.39	0.67	0.32	0.33	**0.79**	**0.51**	0.49	1.21	**0.87**	0.63	0.48	0.71	**0.29**	**0.38**	**0.41**	0.565
MVS-Match-NeuS	**0.37**	**0.63**	0.33	**0.32**	0.80	0.52	**0.46**	1.16	0.90	**0.60**	**0.45**	0.74	**0.29**	0.40	0.42	0.560

Table 7: Quantitative results of NCC with multi-scale features on scan37 of DTU Dataset.

Model	Pre-Trained	Multi-Scale Features	Mean
Baseline	/	/	1.21
Match-NeuS	QuadTree [17]	×	0.671
		✓	0.753
MVS-NeuS	MVSFormer [2]	×	**0.631**
		✓	0.676

7 Additional Qualitative Results

We show additional qualitative results on the DTU dataset in Fig. 4, Fig. 5.

8 Limitation

As shown in Tab. 4, we need to save all off-the-shelf features in the GPU at once, avoiding costly GPU and CPU memory swappings. This becomes particularly crucial when utilizing NCC with high-resolution features, such as QuadTree, which have large dimensions and require more GPU memory. Another limitation is that our work is based on NeuS [18] and it is difficult to model transparent surfaces.

9 Board Impact

Our work focuses on leveraging the wealth of 2D image data to learn and reconstruct the 3D geometry of the world. Our approach enables high-quality 3D

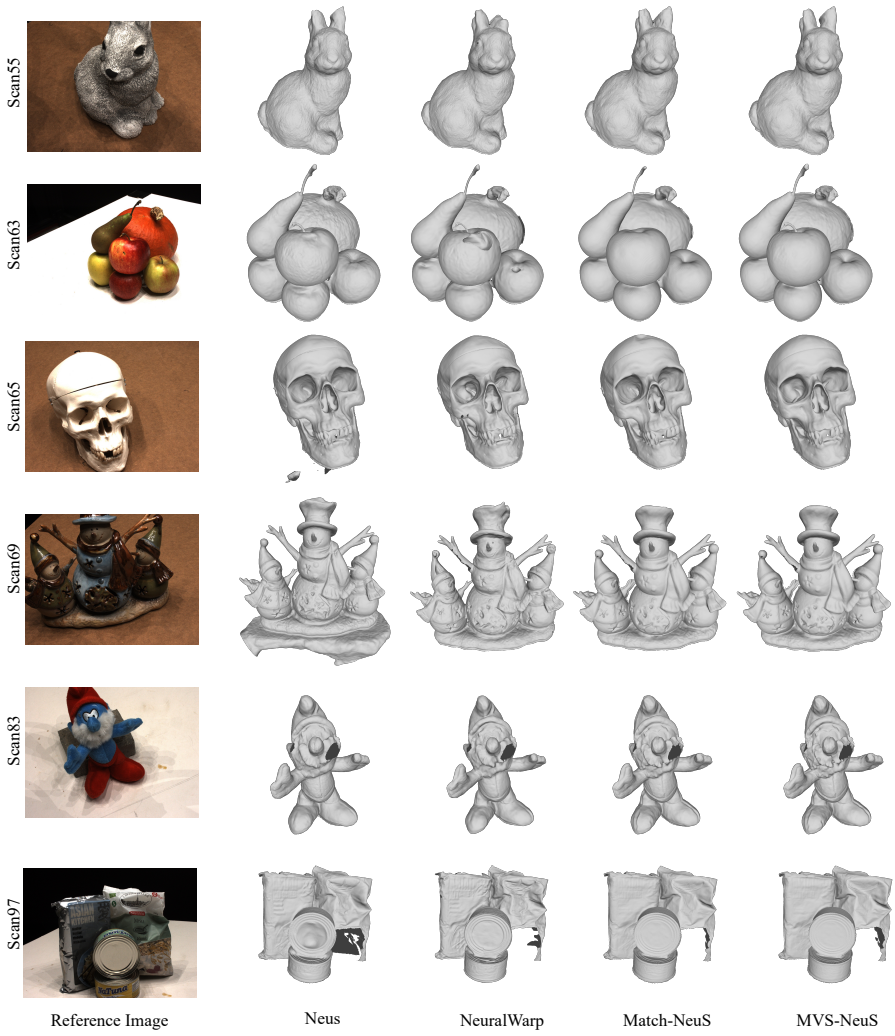

Fig. 4: More qualitative results on the DTU dataset.

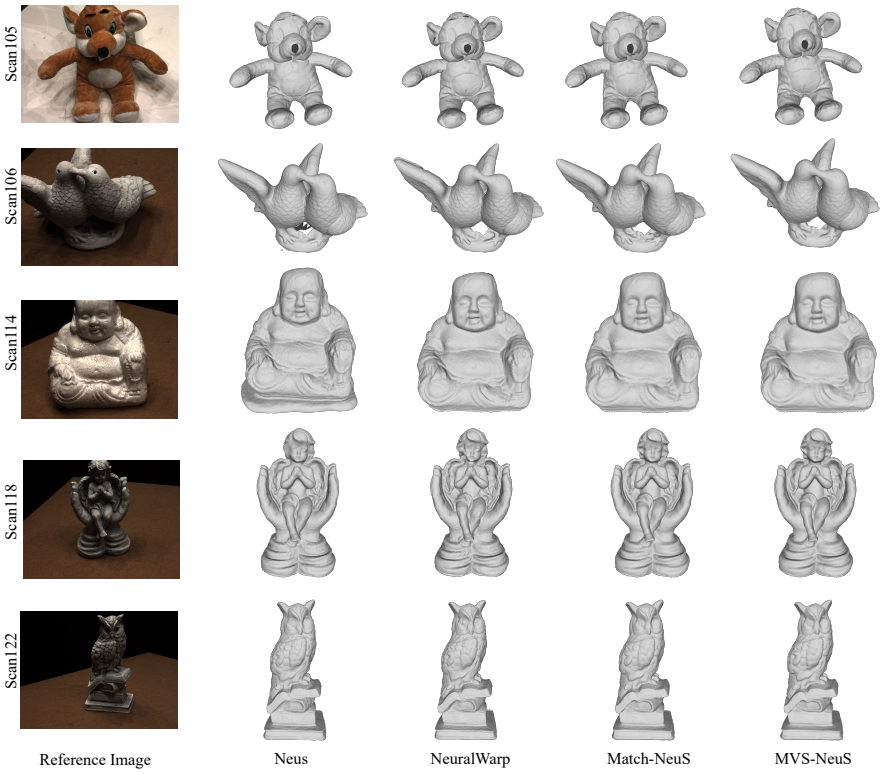

Fig. 5: More qualitative results on the DTU dataset.

reconstruction of objects using only standard images. The applications of our algorithm are diverse and can be utilized in various fields where 3D information is essential but only 2D images are available. This has the potential to benefit areas such as manufacturing, virtual reality, surveillance, healthcare, and many others.

References

1. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)
2. Cao, C., Ren, X., Fu, Y.: Mvsformer: Multi-view stereo by learning robust image features and temperature-based depth. Transactions of Machine Learning Research
3. Chen, D., Zhang, P., Feldmann, I., Schreer, O., Eisert, P.: Recovering fine details for neural implicit surface reconstruction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4330–4339 (2023)
4. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. Advances in Neural Information Processing Systems **34**, 9355–9366 (2021)
5. Fu, Q., Xu, Q., Ong, Y.S., Tao, W.: Geo-neus: geometry-consistent neural implicit surfaces learning for multi-view reconstruction. arXiv preprint arXiv:2205.15848 (2022)
6. Gao, P., Ma, T., Li, H., Dai, J., Qiao, Y.: Convmae: Masked convolution meets masked autoencoders. arXiv preprint arXiv:2205.03892 (2022)
7. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2495–2504 (2020)
8. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5511–5520 (2022)
9. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
10. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 406–413 (2014)
11. Kim, D., Ka, W., Ahn, P., Joo, D., Chun, S., Kim, J.: Global-local path networks for monocular depth estimation with vertical cutdepth. arXiv preprint arXiv:2201.07436 (2022)
12. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
13. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023)

14. Lipson, L., Teed, Z., Deng, J.: Raft-stereo: Multilevel recurrent field transforms for stereo matching. In: 2021 International Conference on 3D Vision (3DV). pp. 218–227. IEEE (2021)
15. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
16. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence **44**(3), 1623–1637 (2020)
17. Tang, S., Zhang, J., Zhu, S., Tan, P.: Quadtree attention for vision transformers. ICLR (2022)
18. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
19. Weinzaepfel, P., Leroy, V., Lucas, T., Brégier, R., Cabon, Y., Arora, V., Antsfeld, L., Chidlovskii, B., Csurka, G., Revaud, J.: Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. arXiv preprint arXiv:2210.10716 (2022)
20. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021)
21. Xu, G., Wang, X., Ding, X., Yang, X.: Iterative geometry encoding volume for stereo matching. arXiv preprint arXiv:2303.06615 (2023)
22. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)