Suppelemtary Material for MOHO: Learning Single-view Hand-held Object Reconstruction with Multi-view Occlusion-Aware Supervision

Chenyangguang Zhang[1]*, Guanlong Jiao[1]*, Yan Di[2], Gu Wang[1], Ziqin Huang[1],
Ruida Zhang[1], Fabian Manhardt[3], Bowen Fu[1], Federico Tombari[2,3] and Xiangyang Ji[1]
[1]Tsinghua University, [2]Technical University of Munich, [3] Google
{zcyg22@mails., xyji@}tsinghua.edu.cn *†

1. Network Architecture

The MOHO network architecture consists of three modules: color feature extraction module, 3D volume rendering head and 2D amodal mask recovery head. Fig. 1 provides an overview of the color feature extraction module and the 3D volume rendering head.

The color feature extraction module bases on ResNet34 [8]. We extract feature pyramids using this backbone, and utilize a bottleneck convolutional layer to obtain the local color feature with channel size of 256. Meanwhile, we use a global average pooling followed by a bottleneck convolutional layer to obtain the global color feature with the same channel size as the local one. The sum of these two features is back-projected onto the corresponding sampled rays, resulting in the sampled color feature denoted as $\mathcal{F}_c^i$.

For 2D amodal mask recovery head, we utilize a decoder architecture consisting of multi-scale atrous convolution and upsampling network referring to the decoder of DeepLabv3+ [4], which is applied to obtain probabilistic hand coverage maps by processing the image feature pyramids.

For 3D volume rendering head, we use two MLPs to encode SDF value and RGB density respectively similar to NeuS [13]. The geometric field ψ_S is modeled by an 8-layer MLP with hidden size of 512. Softplus with $\beta = 100$ is used as activation function for each hidden layer. A skip connection with a scale of $\sqrt{2}/2$ is used at the fourth layer, in order to concatenating the input and intermediate hidden code. The concatenated point feature Cat $\left(\mathcal{F}_c^i, E_P\left(\mathcal{P}_i\right), \mathcal{F}_s^i, \mathcal{F}_h^i\right)$ is fed to the geometric field, and a linear layer with output size of 257 is applied at the end to yield a SDF value s_i and a 256-dimensional SDF feature vector $\mathcal{F}_{SDF}^i$ for this sampled point. Subsequently, the color field ψ_C is modeled by a 4-layer MLP with ReLU as activation function and hidden size of 512. The input is the ray feature consisting of Cat $\left(\mathcal{F}_c^i, E_D\left(\mathcal{D}_i\right), \mathcal{N}^i, \mathcal{F}_{SDF}^i\right)$, where $\mathcal{N}^i$ denotes the normal vector of the geometric field $\mathcal{N}^i = \nabla\psi_S\left(P_i|\mathcal{F}_{con}^i\right)$. The color field yields 3-dimensional RGB density c_i with the help of a linear layer and a Sigmoid layer. We apply it to render the color of the pixel by Eq. 4 in the main manuscript. The E_P and E_D denote the positional and directional encoding functions respectively. We apply E_P for spatial location P_i with 6 frequencies and E_D for viewing direction D_i with 4 frequencies.

2. Details of Synthetic Data Rendering for SOMVideo

For SOMVideo rendering, we generate each hand-object scene on the basis of the released rendering code of ObMan [7] dataset. Following this setting, we select 8 object categories (bottles, bowls, cans, jars, knifes, cellphones, cameras and remote controls) from ShapeNet [1] dataset, which results in a total of 2772 meshes. The object textures are randomly sampled from the texture maps provided with ShapeNet models, and the body textures are sampled from the full body scans used in SURREAL [11]. The skin tone of the hand is matched to the facial color of the body. The backgrounds are sampled from LSUN [16] and ImageNet [9] following the ObMan setting. To render reference views for our synthetic pre-training, we keep the selected shapes, grasps and body poses unchanged as in the ObMan dataset for their plausibility. Thus, the comparison between our proposed pre-training strategy with the previous 3D-supervised pre-training [14] adopting ObMan dataset is strictly fair. We generate 141,550 scenes in total, which exactly corresponds to the scenes in ObMan's training split. After constructing the hand-object interaction scenes and selecting the reference view, we aim to generate multi-view images capturing such hand-object scenes and occlusion-free supervisions. To yield them, we fix the position of the grasped object and rotate the camera around it. The rotated camera trajectory is a circle around the y-axis, centered at the object and with a fixed radius. The radius is randomly sampled between 50 and 80 cm, kept the same as the implementation of ObMan. The camera rotates 360 degrees in total, and the video clips are obtained

*Authors with equal contributions.
†Codes and datasets: https://github.com/ZhangCYG/MOHO

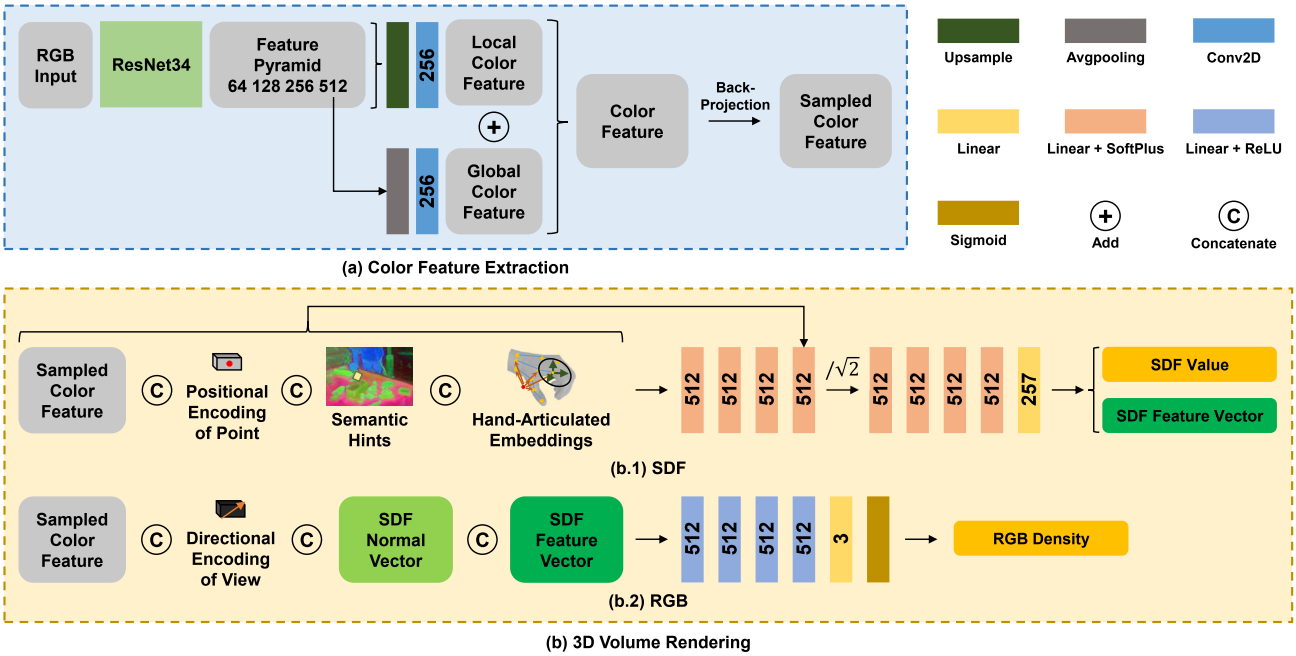

Figure 1. Overview of the MOHO network architecture.

by sampling 10 positions uniformly on the trajectory. We keep the angle of the camera's rotation around the y-axis equal to the angle of the camera's rotation around its origin, in order to force the camera to focus on the object. When rendering the corresponding videos without hand-induced occlusion, we only retain the object without the sampled human body in the scene and set the background to white. Other details are kept exactly the same as the generation process of multi-view hand-object images. Some examples exhibiting our rendered hand-object reference view and occlusion-free supervising views are shown in Fig. 2. The SOMVideo data is released along with our codes.

3. Additional Loss Terms

Two additional losses introduced in Sec. 3.3 of the main manuscript regularizing the predicted surface normals are used for restricting the orientation of visible normals towards the camera ($\mathcal{L}_{n_{ori}}$) [12], and making the predictions smoother ($\mathcal{L}_{n_{smo}}$) [10]:

$$\mathcal{L}_{n_{ori}} = \frac{1}{m}\sum_{i}(min(0, -\hat{n_i} \cdot D_i))^2, \tag{1}$$

$$\mathcal{L}_{n_{smo}} = \frac{1}{K}\sum_{k}(\hat{n_k} - \overline{\hat{n_k}})^2, \tag{2}$$

where K is the capacity of K-nearest-neighbor (KNN) region, set to 16 during implementation; $\hat{n_{k/i}} = \sum_j \omega(j)\nabla\psi_S(P(j))$, corresponding to the sampled ray k or i; $\overline{\hat{n_k}}$ is the average normal vector in the KNN region. The definition of D_i, m, ω, ψ_S and P is kept the same as the main manuscript.

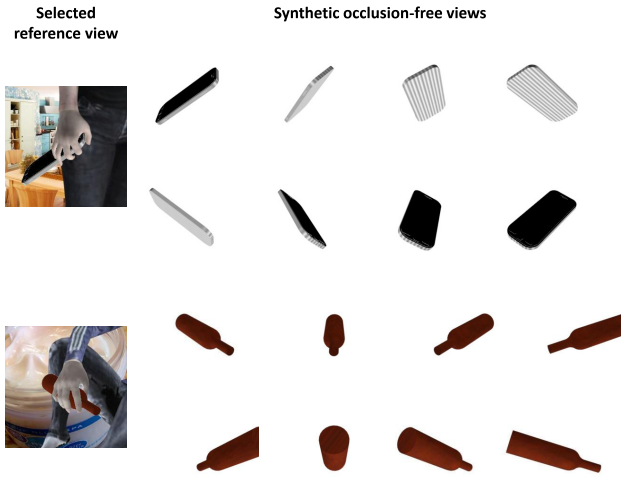

Figure 2. Rendered reference views and occlusion-free views in SOMVideo for our proposed synthetic pre-training.

4. Limitation Analysis

As shown in Fig. 3, although MOHO can reconstruct photorealistic textured mesh of hand-held object from a single view, some holes can be found on the reconstructed surface, as well some inconsistent textures are generated. More

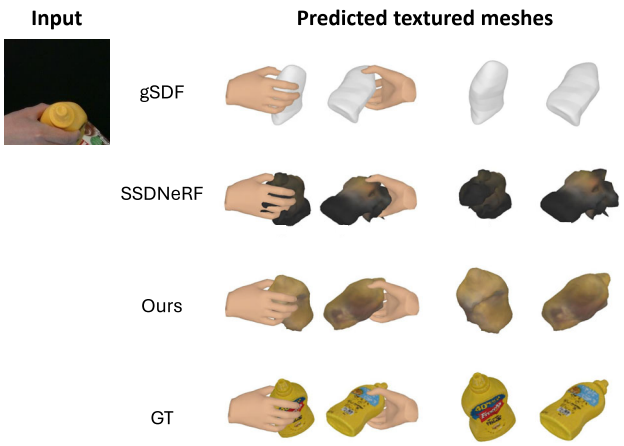

Figure 3. Visualization of failure cases.

Method	HO3D [6]			DexYCB [2]		
	F-5 ↑	F-10 ↑	CD ↓	F-5 ↑	F-10 ↑	CD ↓
IHOI [14]	0.14	0.27	4.36	-	-	-
gSDF [5]	-	-	-	0.15	0.29	1.92
Ours	**0.23**	**0.41**	**1.00**	**0.21**	**0.37**	**1.24**

Table 1. Zero-shot experiments of MOHO against 3D-supervised baselines.

advanced backbones or differentiable rendering techniques could be used for better results. In addition, since current real-world hand-object video datasets are of relatively small scale, the scene, hand and object variety is limited. The generalization ability across large-scale scene, hand and object variety could be improved for MOHO as new powerful datasets are proposed.

5. Efficiency Analysis

To demonstrate the efficiency of MOHO, we compare its running speed to generate the reconstructed object mesh with IHOI, which is the top-performing SDF-based single-view hand-held object reconstruction method. All experiments are conducted on a single NVIDIA A100 GPU with a reference image as the input (the batch size is set to one). MOHO runs at 10 FPS, which is slower than IHOI with 23 FPS, but still achieves comparable efficiency. The decrement of the inference speed mainly comes from the color branch of our network for texture reconstruction.

6. Zero-shot Experiments

Tab. 1 exhibits the zero-shot experiments of MOHO against 3D-supervised baselines. For fair comparison during implementation, both 3D-supervised baselines IHOI and gSDF are pre-trained on ObMan dataset and directly tested on HO3D and DexYCB respectively. MOHO is pre-trained on SOMVideo with exactly the same ObMan shapes. Results show because of the effectiveness of our proposed synthetic pre-training technique for constructing hand-object correlations in both 3D and 2D space, MOHO gains more generalization ability. Concretely, MOHO exceeds IHOI by 64.2% of F-5 on HO3D and leads gSDF by 40.0% of F-5 on DexYCB.

Noise	F-5 ↑	F-10 ↑	CD ↓
Pred	0.60	0.81	0.15
Pred + σ=0.1	0.58	0.78	0.16
Pred + σ=0.5	0.55	0.75	0.18
GT	0.63	0.82	0.14
GT + σ=0.1	0.60	0.79	0.16
GT + σ=0.5	0.57	0.76	0.17

Table 2. Ablation studies for the input predicted hand pose on DexYCB [2].

7. Ablations on the Sensitivity of the Input Hand Pose Predictions

Tab. 2 shows the sensitivity of the input hand pose predictions of MOHO. We add some Gaussian noises with specified variance for this ablation study. Results illustrate that MOHO gains some robustness against wrong and noisy hand pose predictions. Meanwhile, if the quality of input hand poses is improved, MOHO yields more accurate reconstruction results, which also demonstrates the effectiveness of our adopted hand-articulated geometric embeddings.

8. Visual Demonstration of the Occlusion Removal Ability of MOHO

In Fig. 4, we compare the visualization results of novel view synthesis to investigate the occlusion removal ability of MOHO. Specifically, results from SSDNeRF [3], MOHO w/o synthetic pre-training (SYN), and MOHO are exhibited to illustrate the effectiveness of our strategy to resist hand-induced occlusion in real world.

Line 1 indicates that SSDNeRF [3] lacks the ability to remove occlusion, which results in the failure to reconstruct hand-covered regions of the input reference view. The bleach cleanser on the left is reconstructed neglecting the occluded parts (presented as the black fragmentary holes), while the mug on the right is generated with a distorted shape. The main reason is that the incomplete supervision of real-world videos leads the network only to reconstruct visible parts to get local optimum. MOHO w/o SYN can get a little more coherent reconstruction though, the occluded parts are still difficult to complete (the bleach cleanser in the left, line 2). Moreover, the shape distortion is not released utterly due to

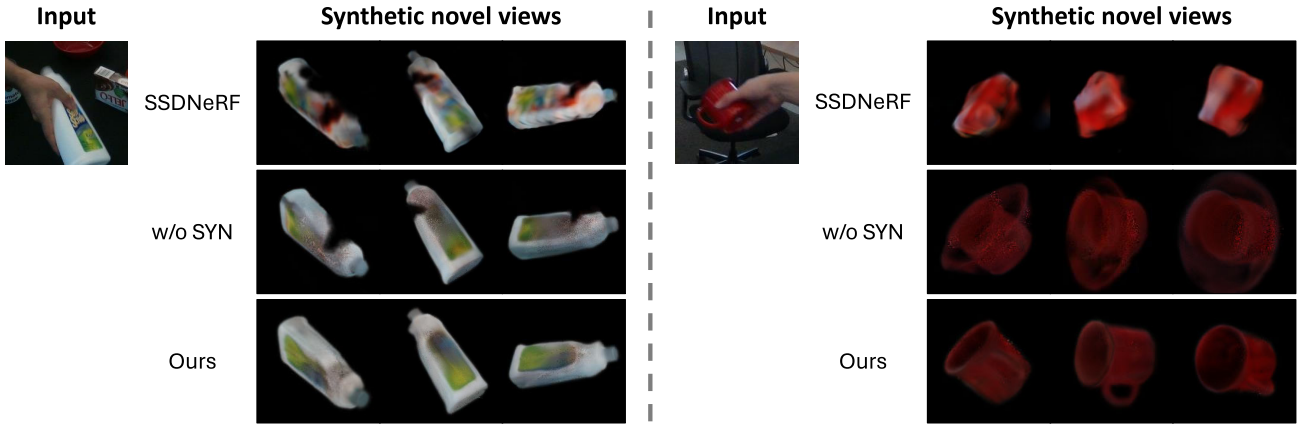

Figure 4. Visual demonstration of the occlusion removal ability.

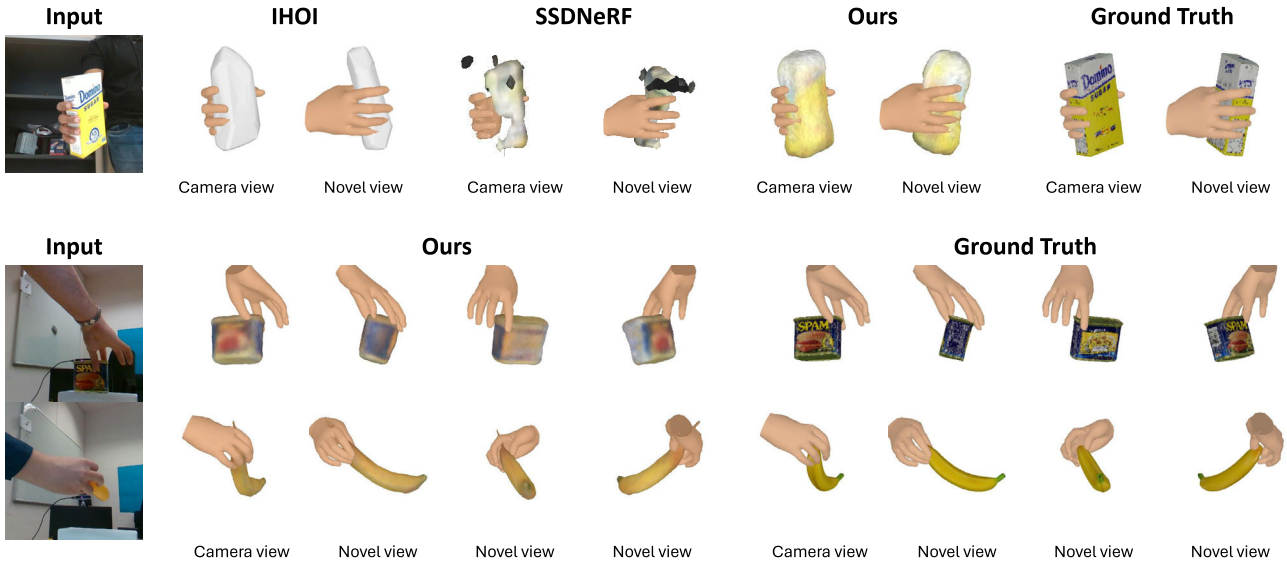

Figure 5. Additional visualization of textured meshes on HO3D [6].

the lack of complete geometric guidance during training (the mug on the right, line 2). In contrast, MOHO with the whole synthetic-to-real framework can solve the problem of hand-induced occlusion greatly due to adequate occlusion-aware knowledge transferring. It generates photorealistic novel views for occluded inputs (Line 3), as well as accurately reconstructs the shape of objects.

9. Additional Qualitative Results

We visualize additional textured meshes predicted by MOHO and some competitors including IHOI [14], gSDF [5] and SSDNeRF [3] in Fig. 5 and Fig. 6 for HO3D [6] and DexYCB [2] respectively. Compared to the baselines, the predicted textured meshes by MOHO are complete and photorealistic, showing that MOHO releases real-world occlusion obviously and performs well in both mesh reconstruction and texture prediction.

10. Qualitative Results of Novel View Synthesis

We visualize novel view synthesis of MOHO and the NeRF-based competitors PixelNeRF [15] and SSDNeRF [3] in Fig. 7 and Fig. 8 for HO3D [6] and DexYCB [2] respectively. Qualitative results on novel view synthesis show due to the imposed partial-to-full cues and the proposed synthetic-to-real framework, MOHO is endowed to handle complex occlusion scenarios in real world and generates more complete, photorealistic, and coherent novel views.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-

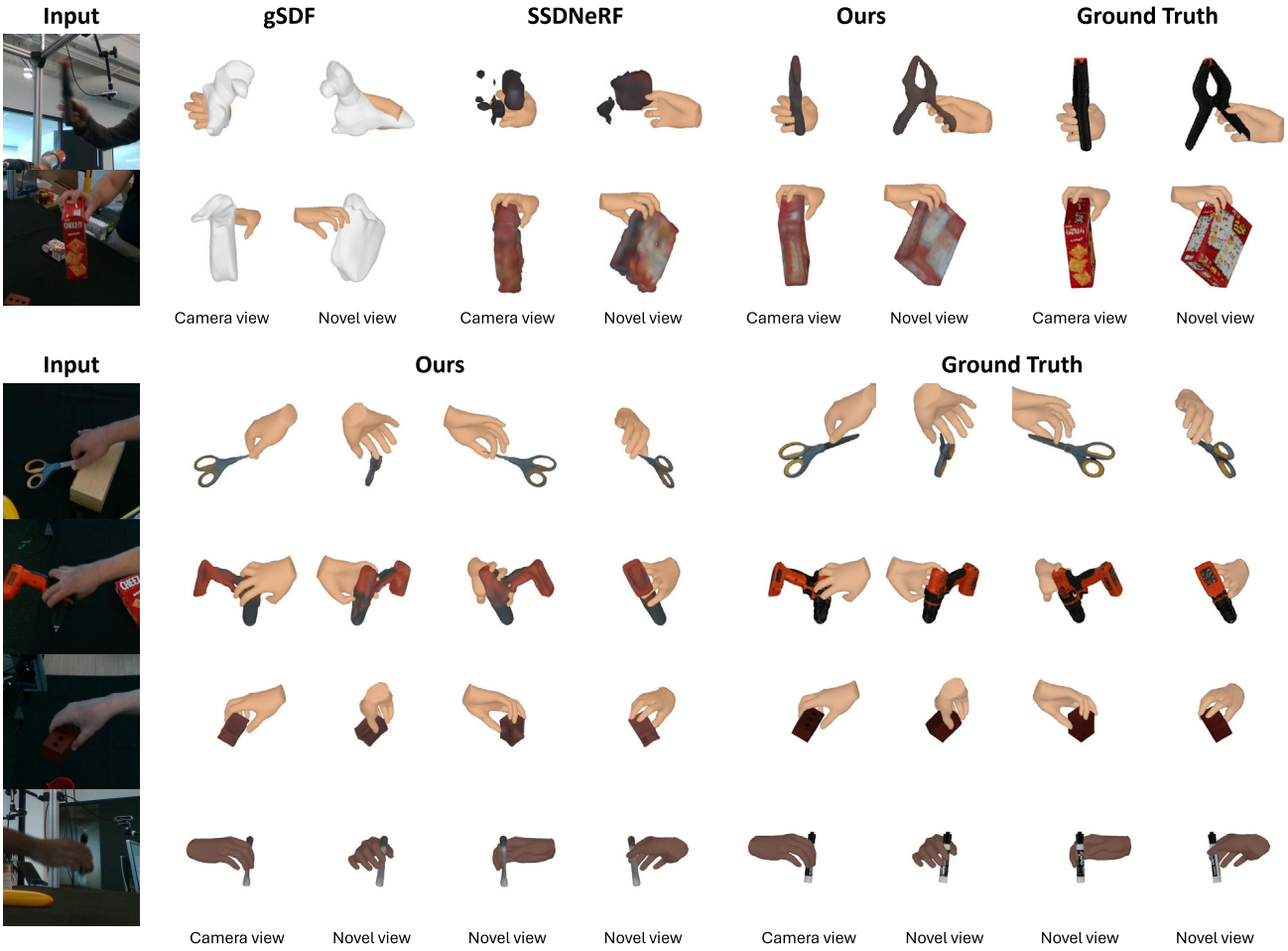

Figure 6. Additional visualization of textured meshes on DexYCB [2].

rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[2] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, pages 9044–9053, 2021. 3, 4, 5, 7

[3] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023. 3, 4

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[5] Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. gsdf: Geometry-driven signed distance functions for 3d hand-object reconstruction. In *CVPR*, pages 12890–12900, 2023. 3, 4

[6] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, pages 3196–3206, 2020. 3, 4, 6

[7] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, pages 11807–11816, 2019. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 1

[10] Rajat Sharma, Tobias Schwandt, Christian Kunert, Steffen Urban, and Wolfgang Broll. Point cloud upsampling and normal estimation using deep learning for robust surface reconstruction. *arXiv preprint arXiv:2102.13391*, 2021. 2

[11] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, pages 109–117, 2017. 1

[12] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields.

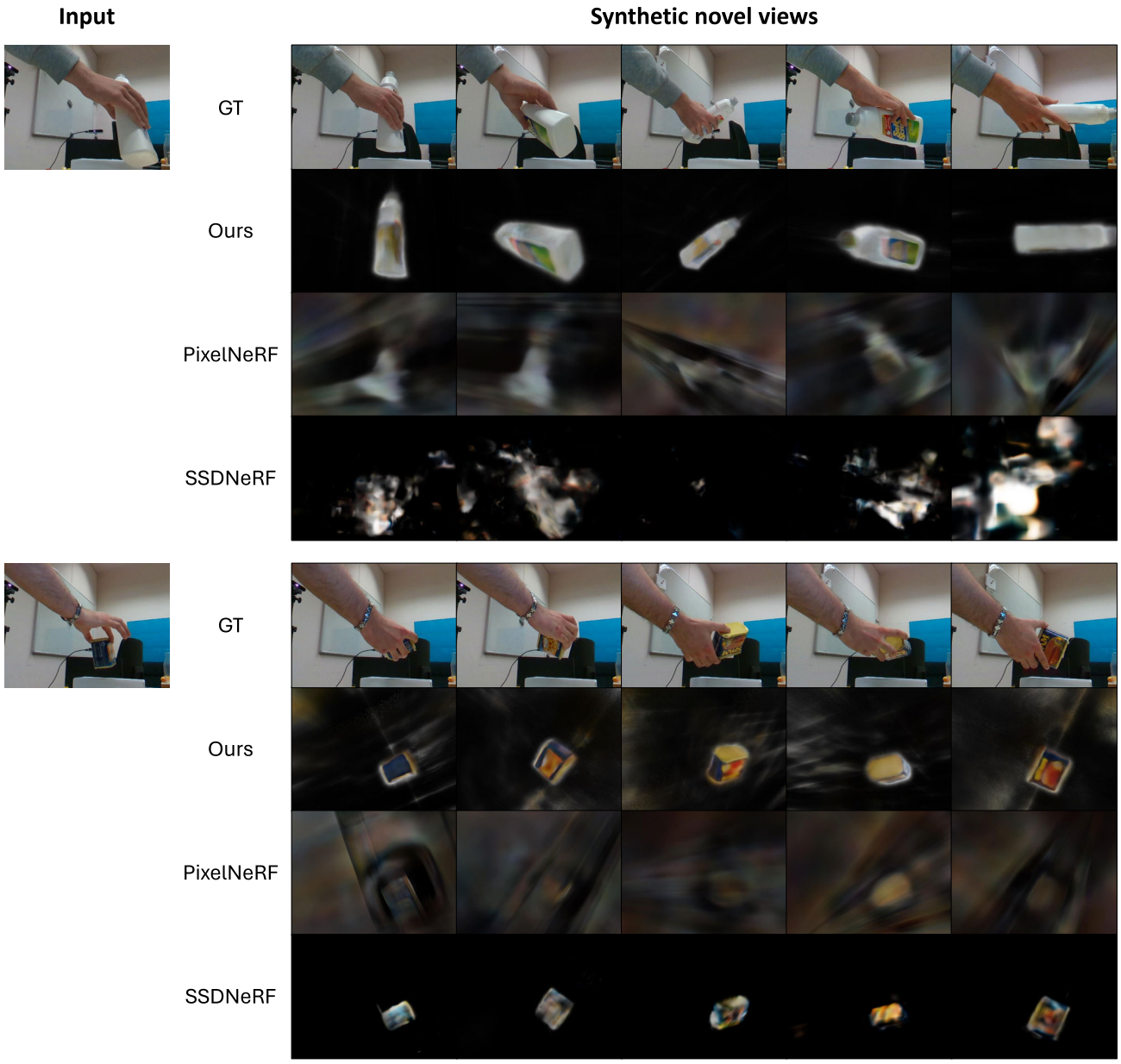

Figure 7. Synthetic novel views on HO3D [6].

In *CVPR*, pages 5481–5490. IEEE, 2022. 2

[13] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 34:27171–27183, 2021. 1

[14] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *CVPR*, pages 3895–3905, 2022. 1, 3, 4

[15] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 4

[16] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 1

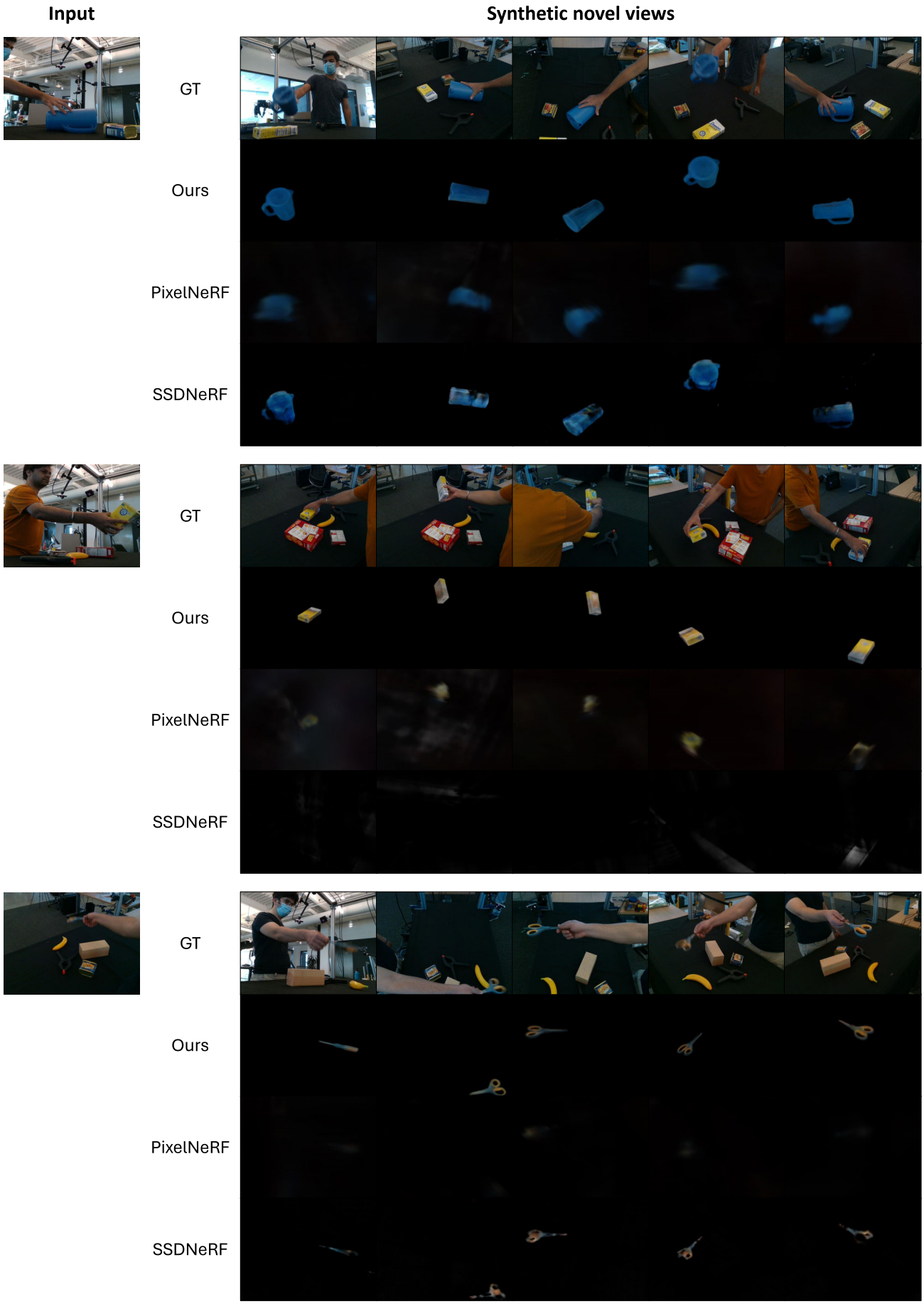

Figure 8. Synthetic novel views on DexYCB [2].