Supplementary Material: Zero-Shot Multi-Object Scene Completion

Shun Iwase[1,2], Katherine Liu[2], Vitor Guizilini[2], Adrien Gaidon[2], Kris Kitani[1,⋆], Rareș Ambruș[2,⋆], and Sergey Zakharov[2,⋆]

[1] Carnegie Mellon University
[2] Toyota Research Institute

1 Implementation Details of Baselines

For occupancy-based networks such as AICNet [6], ConvONet [9], POCO [1], and VoxFormer [7], we use only the averaged BCE loss at LoD-6 (L^6_{occ} in the main paper) for training. For surface-based methods such as MinkowskiNet [2] and OCNN [12], exact the same loss function as our method is used. We use the same hyperparameters for Adam [5] as the proposed method for training.

VoxFormer [7]. We use the implementation from https://github.com/NVlabs/VoxFormer. We make a single modification to adapt to multi-object scene completion. Unlike the original setting, a measured depth map is more accurate than an estimated one. Thus, we directly use the input depth map to extract query tokens in Stage 1. In addition, we leverage trilinear interpolation to reconstruct a surface at LoD-7.

ShapeFormer [14]. We borrow the code from https://github.com/QhelDIV/ShapeFormer. We train VQDIF and ShapeFormer following the paper, and pick the first prediction for evaluation.

MCC [13]. We choose the implementation from https://github.com/facebookresearch/MCC. We train the model with the lower number of sampling points being $1,100$ (twice more than the original implementation) due to their memory-expensive Transformer architecture.

ConvONet [9] We use the implementation from https://github.com/autonomousvision/convolutional_occupancy_networks. We modify the network to accept the encoded feature from an RGB image as well as the point features through concatenation for a fair comparison.

POCO [1] We choose the implementation from https://github.com/valeoai/POCO. As well as ConvONet [9], we modify the network to accept the encoded feature from an RGB image as well as the point features through concatenation for a fair comparison.

⋆ Equal advising.

AICNet [6]. The implementation is borrowed from https://github.com/waterljwant/SSC. Since AICNet [6] takes the same input as our method except a foreground mask. We only make a change in its output channel size from the number of classes to 2 for occupancy prediction.

MinkowskiNet [2]. We adopt the implementation from https://github.com/NVIDIA/MinkowskiEngine. We use the network depth of 5 (LoD-9 to LoD-4) for a fair comparison with the other networks. The occupancy probability of 0.5 is also used for pruninig at each LoD.

OCNN [12] The implementation is taken from https://github.com/octree-nn/ocnn-pytorch. We use the same network architecture and pruning strategy as MinkowskiNet [2] and our method. The sparse tensor structures are the key difference between OCNN [12] and MinkowskiNet [2]. Specifically, MinkowskiNet and OCNN use the hash table and octree respectively.

2 Evaluation Metrics

To compute the metrics, we uniformly sample $100,000$ points on a surface for occupancy-based methods. For surface-based methods, we simply use the point locations predicted as occupied. Here, the predicted and ground-truth points are denoted as $\mathbf{P}_{\mathrm{pd}}$ and $\mathbf{P}_{\mathrm{gt}}$ respectively.

Chamfer distance (CD). The Chamfer distance $\mathrm{CD}(\mathbf{P}_{\mathrm{pd}}, \mathbf{P}_{\mathrm{gt}})$ is expressed as

$$\begin{aligned}\mathrm{CD}(\mathbf{P}_{\mathrm{pd}}, \mathbf{P}_{\mathrm{gt}}) = &\frac{1}{2|\mathbf{P}_{\mathrm{pd}}|} \sum_{\mathbf{x}_{\mathrm{pd}} \in \mathbf{P}_{\mathrm{pd}}} \min_{\mathbf{x}_{\mathrm{gt}} \in \mathbf{P}_{\mathrm{gt}}} ||\mathbf{x}_{\mathrm{pd}} - \mathbf{x}_{\mathrm{gt}}|| \\ &+ \frac{1}{2|\mathbf{P}_{\mathrm{gt}}|} \sum_{\mathbf{x}_{\mathrm{gt}} \in \mathbf{P}_{\mathrm{gt}}} \min_{\mathbf{x}_{\mathrm{pd}} \in \mathbf{P}_{\mathrm{pd}}} ||\mathbf{x}_{\mathrm{gt}} - \mathbf{x}_{\mathrm{pd}}||.\end{aligned} \tag{1}$$

F-1 score. The F-1 score is computed by

$$\begin{aligned}P &= \frac{\left|\left\{\mathbf{x}_{\mathrm{pd}} \in \mathbf{P}_{\mathrm{pd}} \mid \min_{\mathbf{x}_{\mathrm{gt}} \in \mathbf{P}_{\mathrm{gt}}} \|\mathbf{x}_{\mathrm{gt}} - \mathbf{x}_{\mathrm{pd}}\| < \eta\right\}\right|}{|\mathbf{P}_{\mathrm{pd}}|}, \\ R &= \frac{\left|\left\{\mathbf{x}_{\mathrm{gt}} \in \mathbf{P}_{\mathrm{gt}} \mid \min_{\mathbf{x}_{\mathrm{pd}} \in \mathbf{P}_{\mathrm{pd}}} \|\mathbf{x}_{\mathrm{pd}} - \mathbf{x}_{\mathrm{gt}}\| < \eta\right\}\right|}{|\mathbf{P}_{\mathrm{gt}}|},\end{aligned} \tag{2}$$

$$\text{F-1} = \frac{2PR}{P+R}. \tag{3}$$

where we set η to 10 mm for all the experiments.

Normal consistency (NC). Normal consistency measures the alignment of normals between the predicted and ground surfaces.

$$\mathrm{NC}(\mathbf{N}_{\mathrm{pd}}, \mathbf{N}_{\mathrm{gt}}) = \frac{1}{2|\mathbf{N}_{\mathrm{pd}}|} \sum_{\mathbf{n}_{\mathrm{pd}} \in \mathbf{N}_{\mathrm{pd}}} \left(\mathbf{n}_{\mathrm{pd}} \cdot \mathbf{n}^*_{\mathrm{gt}}\right) + \frac{1}{2|\mathbf{N}_{\mathrm{gt}}|} \sum_{\mathbf{n}_{\mathrm{gt}} \in \mathbf{N}_{\mathrm{gt}}} \left(\mathbf{n}_{\mathrm{gt}} \cdot \mathbf{n}^*_{\mathrm{pd}}\right). \tag{4}$$

where $\mathbf{n}^*_{\mathrm{gt}}$ and $\mathbf{n}^*_{\mathrm{pd}}$ refer the nearest normal vectors.

3 Derivation of RoPE [11]

RoPE [11] utilizes a rotation matrix to encode positional information to features. Given normalized 1D axial coordinate $x \in \mathbb{R}$, $R : \mathbb{R} \to \mathbb{R}^{\lfloor D'/3 \rfloor \times \lfloor D'/3 \rfloor}$ is defined as

$$R(x) = \begin{bmatrix} \cos x\theta_1 & -\sin x\theta_1 & \cdots & 0 & 0 \\ \sin x\theta_1 & \cos x\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos x\theta_{k/2} & -\sin x\theta_{k/2} \\ 0 & 0 & \cdots & \sin x\theta_{k/2} & \cos x\theta_{k/2} \end{bmatrix}, \tag{5}$$

where $\theta_i = \left(1 + \frac{\lfloor D'/2 \rfloor - 1}{\lfloor D'/6 \rfloor - 1}\right)(i-1)\,\pi, i \in [1, 2, \cdots, \lfloor D'/6 \rfloor]$.

$$\mathbf{R}_i = \begin{bmatrix} R(p_i^x) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R(p_i^y) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R(p_i^y) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{D' \times D'}, \tag{6}$$

$$\mathbf{f}_i' = \mathbf{R}_i \mathbf{f}_i, \tag{7}$$

where $\mathbf{f}_i \in \mathbb{R}^{D'}$, and $\mathbf{p}_i \in \mathbb{R}^3$ is an i-th octree feature and coordinates.

4 Additional Experiments

4.1 Comparison against single-object methods

We trained our method, MCC [8] and ZeroShape [3] on our synthetic dataset with a single-object setup. For a fair comparison, we use ground-truth camera intrinsics and depth maps for Zeroshape. During evaluation, we complete each object individually and then concatenate all the completed objects in a scene. Table 1 demonstrates that our method with single- and multi-object setups outperforms the others regarding completion quality and runtime. Here, 1 and N in #Obj denote single- and multi-object setups, and CD_{occ} is Chamfer distance of occluded surfaces. The large difference between CD and CD_{occ} of single-object methods clearly show its poor occlusion handling due to the lack of multi-object reasoning.

4.2 Foreground vs Instance Masks

Zero-shot 2D instance segmentation of cluttered scenes is still challenging. For instance, the SoTA foreground detection model (InSPyReNet [4]) gives a 14.9% higher IoU of a foreground mask than Grounded-SAM [10] (G-SAM), the latest zero-shot instance segmentation model, on HOPE dataset (69.3% vs 54.4%). For G-SAM, foreground masks are computed by combining its instance mask predictions, and its input prompt are manually tuned to improve an IoU. Further, Table 1 validates that using foreground masks during inference largely improves the final completion quality.

	#Obj	Mask Source	CD↓	CD_{occ}↓	F1↑	NC↑	Runtime↓
MCC [13]	1	Ground truth	14.39	20.26	0.482	0.694	3.5×10^4
ZeroShape [3]	1	Ground truth	12.19	17.44	0.603	0.703	965.5
Ours	1	G-SAM [10]	14.98	20.73	0.666	0.683	240.3
	N	G-SAM [10]	13.56	16.19	0.700	0.699	83.4
	N	InSPyReNet [4]	12.31	14.47	0.724	0.712	84.0
	1	Ground truth	8.55	12.48	0.758	0.730	248.3
	N	Ground truth	**6.97**	**9.31**	**0.803**	**0.750**	**85.1**

Table 1: Ablations of mask sources and the number of target objects on the HOPE dataset.

References

1. Boulch, A., Marlet, R.: POCO: Point Convolution for Surface Reconstruction. In: CVPR (2022) 1
2. Choy, C., Gwak, J., Savarese, S.: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In: CVPR (2019) 1, 2
3. Huang, Z., Stojanov, S., Thai, A., Jampani, V., Rehg, J.M.: ZeroShape: Regression-based Zero-shot Shape Reconstruction. CVPR (2023) 3, 4
4. Kim, T., Kim, K., Lee, J., Cha, D., Lee, J., Kim, D.: Revisiting Image Pyramid Structure for High Resolution Salient Object Detection. In: Proceedings of the Asian Conference on Computer Vision. pp. 108–124 (2022) 3, 4
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 1
6. Li, J., Han, K., Wang, P., Liu, Y., Yuan, X.: Anisotropic Convolutional Networks for 3D Semantic Scene Completion. In: CVPR (2020) 1, 2
7. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: VoxFormer: Sparse Voxel Transformer for Camera-based 3D Semantic Scene Completion. In: CVPR (2023) 1
8. Lin, Y., Tremblay, J., Tyree, S., Vela, P.A., Birchfield, S.: Multi-view Fusion for Multi-level Robotic Scene Understanding. In: IROS (2021) 3
9. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional Occupancy Networks. In: ECCV (2020) 1
10. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks (2024) 3, 4
11. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: RoFormer: Enhanced Transformer with Rotary Position Embedding. In: ICLR (2020) 3
12. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-CNN: Octree-Based Convolutional Neural Networks for 3D Shape Analysis. SIGGRAPH (2017) 1, 2
13. Wu, C.Y., Johnson, J., Malik, J., Feichtenhofer, C., Gkioxari, G.: Multiview Compressive Coding for 3D Reconstruction. In: CVPR (2023) 1, 4
14. Yan, X., Lin, L., Mitra, N.J., Lischinski, D., Cohen-Or, D., Huang, H.: ShapeFormer: Transformer-based Shape Completion via Sparse Representation. In: CVPR (2022) 1