Object-Compositional Neural Implicit Surfaces

Supplementary Material

Qianyi Wu[1], Xian Liu[2], Yuedong Chen[1], Kejie Li[3], Chuanxia Zheng[1], Jianfei Cai[1], and Jianmin Zheng[4]

[1] Monash University
[2] The Chinese University of Hong Kong
[3] University of Oxford
[4] Nanyang Technological University

In this supplementary material, we will describe the dataset information, more details about the comparison setting, and some analysis of our method.

A Dataset

We use the dataset as in [4] for a fair comparison. Here we give a brief introduction about these two datasets.

ToyDesk Dataset The ToyDesk dataset contains two image sets with 96 and 151 posed images and the corresponding instance segmentation. They capture the scene and use SfM [2] and 3D reconstruction techniques [1,3] to recover the meshes with camera poses. And for train/test set split, they randomly sample 80% frames for training and use the rest for testing. We also use their train/testing data split as they give in the GitHub issue[5].

ScanNet Dataset In our experiment, we choose 'scene0024_00', 'scene0038_00', 'scene0113_00' and 'scene0192_00' in ScanNet as used in ObjectNeRF [4] for fair comparison. For the experiment conducted in these data, we resize the image resolution to 320×240 in order to match image resolution in SemanticNeRF [6] and avoid the OOM issue. To match the training setting of SemanticNeRF, we use the category semantic label of ScanNet for network training and the mIOU metric evaluation of each method.

B Comparison Setting Details

We introduce the details in the comparison setting. Firstly, we will introduce the pipeline we used in calculating the semantic map of ObjectNeRF [4] since it does not explicitly produce such a result. The main principle we use in computing the semantic map of ObjectNeRF is the Z-buffer algorithm. We use the object branch in ObjectNeRF to compute the depth of each object using the following equation:

$$\hat{D}_i(\mathbf{r}) = \int_{v_n}^{v_f} T_i(v)\sigma_i(\mathbf{r}(v))v dv, \tag{1}$$

[5] https://github.com/zju3dv/object_nerf/issues/2

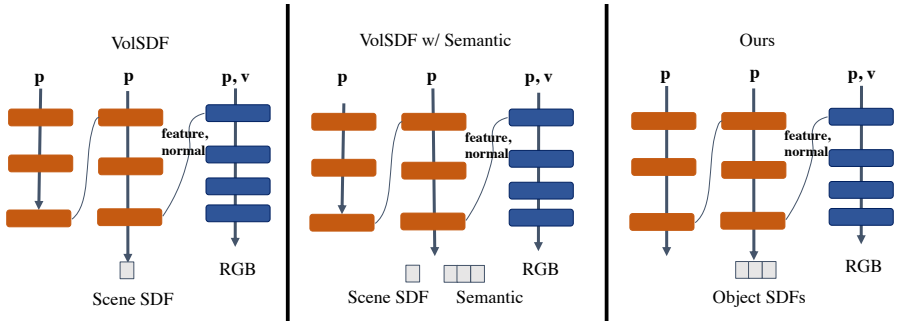

Fig. 1. Network structure of model design ablation., we show the network structure design in the ablation study, including "VolSDF", "VolSDF w/ Semantic", and Ours. The inputs for all methods are point position $\mathbf{p}$ and view direction $\mathbf{v}$. The difference lies in the output branch of the first MLP (orange part).

where the T_i, σ_i are the object transparency and object density from i-th object from object branch, and v is the value of depth along the ray $\mathbf{r}$. After computing the i-th object's depth of the ray $\hat{D}_i(\mathbf{r})$, we use the object with minimum depth value in ray $\mathbf{r}$ as the semantic label in this pixel.

We also provide the opacity computation in the experiments. The opacity is a complement probability of $T(\mathbf{r})$, which can be used to understand whether this ray be occluded in the final. The value of opacity lies in $[0, 1]$. It can be calculated as:

$$\hat{O}(\mathbf{r}) = \int_{v_n}^{v_f} T(v)\sigma(\mathbf{r}(v))dv. \tag{2}$$

We adopt this value in computing the opacity map to judge the quality of rendering a single object. If the opacity of a ray is 0, we paint it as black in the rendered image and paint it as white if it reaches 1.

C Ablation study

We provide more details about the ablation study related to model design. The model structure of different variants can be found in Fig. 1. The difference lies in the output of the first MLP (orange part). VolSDF [5] predict the scene SDF and "VolSDF w/ Semantic" predict the scene SDF with an additional semantic prediction. However, in our framework, we directly predict the SDF of different objects and transfer them to scene SDF and semantic with a transformation function.

We provide the details in "VolSDF w/ Semantic" to obtain each object representation by extracting the object with a threshold. Suppose we expect to obtain the i-th object, we will get the semantic label $\mathbf{s}$ and volume density σ of each point. Then we apply SoftMax operation to normalize the semantic label $\mathbf{s}$ and judge whether the i-th semantic label large than the given threshold τ, *i.e.*,

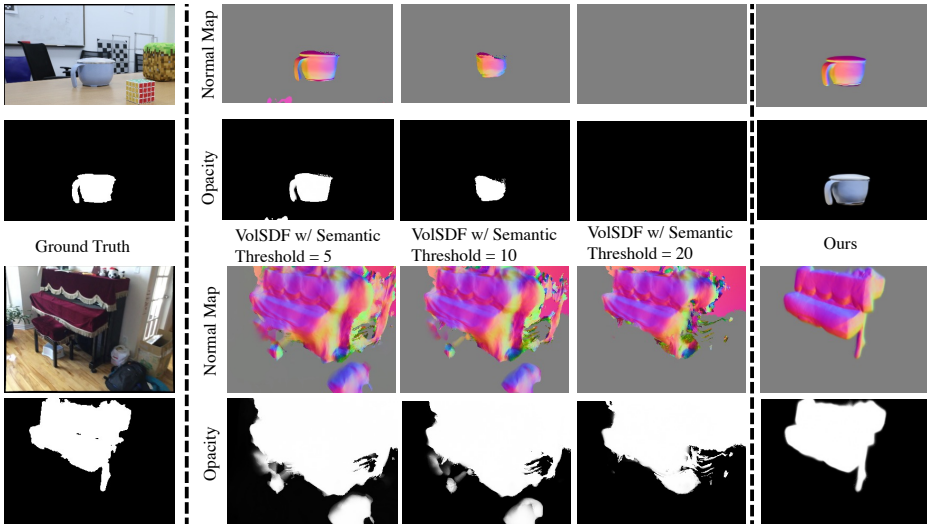

Fig. 2. Apply threshold in original semantic prediction, we show the result that applying a threshold to extract an object from the original semantic prediction of "VolSDF w/ Semantic". From left to right, we show the ground truth image and instance mask, the results of "VolSDF w/ Semantic" and the results of ours

$\text{SoftMax}(\mathbf{s})_i > \tau$. If the semantic label meets the requirement, we will adopt the density in this place for rendering the final result.

In the supplementary, we also show more results in extracting the instance in original semantic value rather than normalized semantic value using SoftMax. The result is given in Fig. 2. We use thresholds $5, 10$, and 20 to extract the object. And we also notice that when the threshold is 10, the extracted teapot is getting ruined but the piano in the bottom is far away from the ground truth. For different instances, we cannot use the same threshold to extract the object precisely for the variant "VolSDF w/ Semantic". It again demonstrates the robustness of our proposed framework in representing objects inside the scene.

D Analysis of Our framework

There still exist some limitations of our framework. As our method regarding the background as an individual object, we can also visualize the reconstructed result of the background. We give an example from ToyDesk. As shown in Fig. 3, we notice that there are some holes in the desk region. The reason behind it is the lack of sufficient observation information in the invisible part. A possible solution to solve it is incorporating some physics constraint or causality guidance to constraint the reconstruction quality of the invisible region. We also show the rendered result of the desk, and we can observe that the texture in the invisible

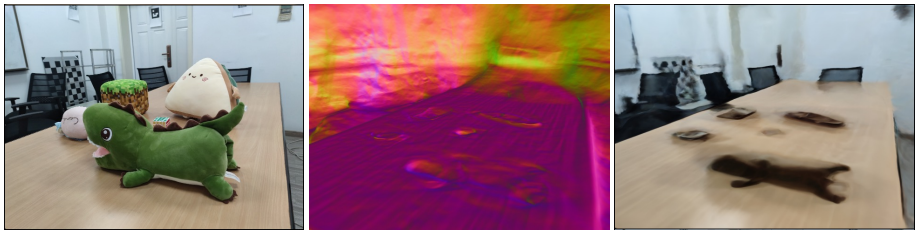

Fig. 3. Analysis of our framework, we show the result of rendered desk result from the Toydesk dataset. From left to right, we show the ground truth image, the rendered normal map of desk (background) and the rendered image of desk. Due to the lack of observation in the bottom region of each toy, our framework cannot guarantee the reconstruction result in the invisible regions.

region also contains some artifacts. It also resulted from a lack of observation in the invisible region. Solving the reconstruction and texture issue in the invisible region is crucial for a further application like realistic scene editing. We will explore this problem in future work.

References

1. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
2. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
3. Xu, Q., Tao, W.: Multi-scale geometric consistency guided multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5483–5492 (2019)
4. Yang, B., Zhang, Y., Xu, Y., Li, Y., Zhou, H., Bao, H., Zhang, G., Cui, Z.: Learning object-compositional neural radiance field for editable scene rendering. In: International Conference on Computer Vision (ICCV) (October 2021)
5. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. arXiv preprint arXiv:2106.12052 (2021)
6. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.: In-place scene labelling and understanding with implicit scene representation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)