Supplementary material for Inner-Outer Aware Reconstruction Model for Monocular 3D Scene Reconstruction

Yu-Kun Qiu[1] **Guo-Hao Xu**[1] **Wei-Shi Zheng**[1,2] *
[1] School of Computer Science and Engineering, Sun Yat-sen University, China
[2] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{qiuyk,xugh23}@mail2.sysu.edu.cn
wszheng@ieee.org

A More Details about IOAR

Definition of ground-truth projective occupancy. In this work, we followed VoRTX [1] to use a transformer to fuse the feature volumes from N different views and assign weights based on projective occupancy. For each view n, the projective signed distance function (SDF) S_v^n of voxel v can define as:

$$S_v^n = d_v^n - d_v. \tag{1}$$

Here, d_v^n is the ground truth depth along the camera ray corresponding to voxel v (*i.e.*, the distance from the camera of view n to the first surface along the camera ray corresponding to voxel v) and d_v is the camera-to-voxel depth (*i.e.*, the distance from the camera of view n to the voxel v). Then, the ground-truth projective occupancy O_v^n of voxel v in view n can be defined as:

$$O_v^n = \begin{cases} 1 & |S_v^n| < t, \\ 0 & |S_v^n| \geq t. \end{cases} \tag{2}$$

Here, t is a truncated distance which is usually set to a multiple of the voxel size (In our work, t is set to triple of the voxel size).

Details about the expansion operation. To predict the occupancy/TSDF volume in the medium and fine levels, the occupancy/TSDF features and logits in the previous level are utilized. However, the features and logits from the previous level have different volume sizes with those at the current level. Specifically, the volumes in coarse, medium, and fine levels have the size of $N_x^c \times N_y^c \times N_z^c$, $N_x^m \times N_y^m \times N_z^m$ and $N_x^f \times N_y^r \times N_z^r$. Here, $N_x^f = 2N_x^m = 4N_x^c$, $N_y^f = 2N_y^m = 4N_y^c$ and $N_z^f = 2N_z^m = 4N_z^c$. As a result, the size of medium-level volumes are eight times of the size of coarse-level volumes. Therefore, each voxel in the coarse level volumes is corresponded to eight voxels in the medium level. Specifically, each voxel in the expanded coarse TSDF features $\hat{F}_{TSDF}^c$ can be assigned by:

$$\begin{aligned} &\hat{F}_{TSDF}^c(\hat{x}, \hat{y}, \hat{z}) = F_{TSDF}^c(x, y, z), \\ &where\ \hat{x} = 2x + o, \hat{y} = 2y + o, \hat{z} = 2z + o, o \in \{0, 1\}. \end{aligned} \tag{3}$$

We can also get the expanded occupancy features in a similar way. These expanded features from the previous level have the same size as the features at the current level. Therefore, they can be concatenated with features at the current level to make predictions.

Definition of all metrics in the paper. Definition of all metrics are summarized in Table. A.1.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Table A.1: Definition of all 2D and 3D metrics in the paper. Here, N is the set of all pixels in the depth map, and n is a pixel in the depth map. d_n is the predicted depth value of pixel n and d_n^* is the ground-truth depth value of pixel n. P and P^* are the predicted point cloud and ground-truth point cloud. p is a point in the predicted point cloud and p^* is a point in the ground-truth point cloud.

2D Metrics		3D Metrics	
Metrics	Definition	Metrics	Definition
Abs Rel	$\frac{1}{\lvert N\rvert}\sum_{n\in N}\lvert d_n - d_n^*\rvert / d_n^*$	Acc	$\frac{1}{\lvert P\rvert}\sum_{p\in P}(\min_{p^*\in P^*}\lVert p-p^*\rVert_2)$
Abs Diff	$\frac{1}{\lvert N\rvert}\sum_{n\in N}\lvert d_n - d_n^*\rvert$	Comp	$\frac{1}{\lvert P^*\rvert}\sum_{p^*\in P^*}(\min_{p\in P}\lVert p-p^*\rVert_2)$
Sq Rel	$\frac{1}{\lvert N\rvert}\sum_{n\in N}\lvert d_n - d_n^*\rvert^2 / d_n^*$	Prec	$\frac{1}{\lvert P\rvert}\sum_{p\in P}\mathbb{I}(\min_{p^*\in P^*}\lVert p-p^*\rVert_2 < 0.05)$
RMSE	$\sqrt{\frac{1}{\lvert N\rvert}\sum_{n\in N}\lvert d_n - d_n^*\rvert^2}$	Recal	$\frac{1}{\lvert P^*\rvert}\sum_{p^*\in P^*}\mathbb{I}(\min_{p\in P}\lVert p-p^*\rVert_2 < 0.05)$
delta-1.25	$\frac{1}{\lvert N\rvert}\sum_{n\in N}\max(\frac{d_n}{d_n^*}, \frac{d_n^*}{d_n} < 1.25)$	F-Score	$\frac{2\times Prec\times Recal}{Prec+Recal}$

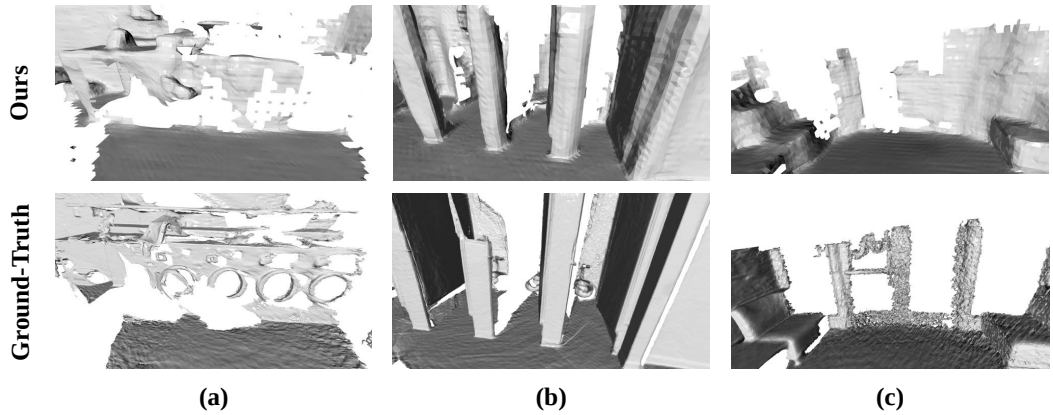

Figure B.1: Visualization of some challenging cases that our model only have low reconstruction quality.

B Limitation

In this section, we discuss the limitation of our model. After analyzing the cases that have low reconstruction quality, we found that our model performs poorly in three challenging cases. These cases are summarized in Figure. B.1. First, the reconstruction quality of our model may be poor if the surface of instances is only partially observed, which is demonstrated in Figure B.1 (a). These washing machines are only partially observed in all video frames and thus, our model can not reconstruct it properly. In fact, even ground-truth 3d meshes have only partial surface of these washing machines as well. Secondly, the reconstruction quality of our model may be low if the instance is heavily occluded. We illustrate one such case in Figure B.1 (b). These toilets are heavily occluded by the door (the ground-truth of these toilets only contains partial surface due to occlusion) and our model can not reconstruct them in this case. Finally, our model may perform poorly if the object is transparent or semi-transparent. We show such a case in Figure B.1 (c). In this case, our model can not reconstruct the glasses on the doors. Solving these challenging cases is a promising future research topic since almost all existing methods suffer from them.

C Error Bar

We have run our model and two variant models 5 times to ensure the reproducibility of our experiments. The resulting Precision, Recall and F-score mean and deviation are shown in Figure C.1, with standard deviation visualized as error bars.

*Corresponding author.

(a) **(b)** **(c)**

Figure C.1: The mean and standard deviation of Precision, Recall and F-Score of our model and two variants. Here, the mean and standard deviation are visualized as blue points and red error bars, respectively.

D Additional Experiments

Weights of different losses. Assigning different weights to different losses may result in better performance. To evaluate the impact of each loss weights, we split the losses into three groups: the occupancy losses, the TSDF loss and the projective occupancy losses. We conduct experiments by setting different weights for these three groups of losses. We denote the weight for occupancy losses as λ_1, the weight for TSDF loss as λ_2, and the weight for projective occupancy losses as λ_3. Thus, the overall loss $\mathcal{L}^W$ can be formulated as:

$$\mathcal{L}^W = \lambda_1(\mathcal{L}^c_{OCC} + \mathcal{L}^m_{OCC}) + \lambda_2\mathcal{L}^f_{TSDF} + \lambda_3(\mathcal{L}^c_P + \mathcal{L}^m_P + \mathcal{L}^f_P) \tag{4}$$

We summarized the experiment results in Table D.1. All variants achieve comparable performance except the one with $\lambda_1 = 0.5$. In summary, assigning different weights to these losses can lead to slight performance variance, especially decreasing the weight of occupancy losses.

Table D.1: Evaluation of assigning different weights to different losses.

λ_1	λ_2	λ_3	Acc	Comp	Prec	Recall	F-score
0.5	1	1	0.045	0.095	0.733	0.580	0.646
1	0.5	1	0.046	0.087	0.740	0.600	0.661
1	1	0.5	0.044	0.090	0.744	0.597	0.661
1	1	1	0.043	0.090	0.748	0.597	0.663

Applying occupancy losses and TSDF losses to all resolutions. The occupancy prediction aims to reduce memory cost by reducing the voxel number in the next stage. Since we filter out inner-surface and outer-surface voxels at the end of coarse and medium levels, we only supervise the occupancy prediction in these two resolutions. Similarly, we only supervise the TSDF prediction in the fine resolution because we aim to reconstruct the surface in the fine resolution. Readers may be curious about what if we apply occupancy losses and TSDF losses to all resolutions. We conducted experiments to answer this question. We denote IOAR^{OCC} as the IOAR variant applied occupancy losses to all resolutions, IOAR^{TSDF} as the IOAR variant applied TSDF losses to all resolutions, and IOAR^{ALL} as the IOAR variant applied both occupancy and TSDF losses to all resolutions. The experiment results are reported in Table D.2. All variants achieve comparable performance except the variant applied TSDF losses to all resolutions drops a bit on Recall.

Removing the medium level. An intuitive way to reduce the computational cost is removing the medium level, which results in a two-level vairant of IOAR. Will performance drops critically in this case? To answer this question, we implemented the two-level variant (denoted as IOAR^{TL}) and reported the result in Table D.3. As we can see, by removing the medium level, the performance drops a lot in all metrics. Therefore, simply removing the medium level is not a good choice for reducing the computational cost.

Table D.2: Evaluation of applying occupancy and TSDF losses to all resolutions.

Model	Acc	Comp	Prec	Recall	F-score
IOAR^{OCC}	0.044	0.093	0.750	0.592	0.660
IOAR^{TSDF}	0.042	0.100	0.751	0.579	0.652
IOAR^{ALL}	0.044	0.089	0.743	0.593	0.658
IOAR	0.043	0.090	0.748	0.597	0.663

Table D.3: Evaluation of IOAR with only two levels.

Model	Acc	Comp	Prec	Recall	F-score
IOAR^{TL}	0.058	0.096	0.670	0.553	0.604
IOAR	0.043	0.090	0.748	0.597	0.663

Accuracy of discriminating voxels. We conducted experiments to analyze the accuracy of discriminating voxels at coarse and medium levels. The results are reported in Table D.4 and D.5. At both levels, only a few inner-surface voxels are misclassified as outer-surface voxels and only a few outer-surface voxels are misclassified as inner-surface voxels. This supports our assumption that outer voxels are quite different from inner voxels since the model can easily distinguish them in most cases. In addition, we can observe that a lot of inner and outer voxels are predicted as surface voxels at the medium level. After the filtering process at the coarse level, most remaining voxels are close to the surface, so their features are similar to surface voxels. As a result, our model misclassified them as surface voxels.

Table D.4: Accuracy of IOAR discriminating voxels at coarse level.

Corase	Inner Voxels	Outer Voxels	Surface Voxels
Predicts as Inner Voxels	71.83%	4.46%	4.17%
Predicts as Outer Voxels	2.67%	60.17%	0.77%
Predicts as Surface Voxels	25.50%	35.37%	95.06%

E Broader Impacts

We think there exist potential misuse cases of the monocular 3D reconstruction model. Since it is easy to record a monocular video and the camera extrinsic of each frame can be estimated by existing SLAM[2] system or SfM [3] system, a malicious user may use it to reconstruct the 3D structure of private area. To try our best to prevent it from happening, we design a safeguard license which demands our users to promise not to misuse the model.

F Safeguards

To prevent our model from misusing, we design the safeguard license as follows. To use our pretrained model, a user is required to agree to the following terms and conditions:

1. The user should use the model only for non-commercial purposes (e.g., research and education).
2. The user accepts full responsibility for his or her use of the model.
3. The user should require his or her research associates and colleagues to agree to these terms and conditions before providing access to the model.
4. We reserve the right to terminate the user's access to the pretrained models at any time.

Table D.5: Accuracy of IOAR discriminating voxels at medium level.

Medium	Inner Voxels	Outer Voxels	Surface Voxels
Predicts as Inner Voxels	47.30%	4.73%	6.14%
Predicts as Outer Voxels	2.48%	48.40%	4.82%
Predicts as Surface Voxels	50.22%	46.87%	89.04%

References

[1] N. Stier, A. Rich, P. Sen, and T. Höllerer, "Vortx: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion," in *International Conference on 3D Vision*, pp. 320–330, IEEE, 2021.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transcation Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, IEEE, 2016.