GenNBV: Generalizable Next-Best-View Policy for Active 3D Reconstruction

Supplementary Material

1. Implementation Details

1.1. Occupancy Grid Mapping Algorithm

Before updating the probabilistic occupancy grid F_t^G, Bresenham's line algorithm is implemented to cast the ray path in 3D space between the camera viewpoint and the endpoints among the point cloud back-projected from D_{t+1}. According to the classical occupancy grid mapping algorithm [11], we have the log-odds formulation of occupancy probability:

$$\log Odd(v_i|z_j) = \log Odd(v_i) + \log \frac{p(z_j|v_i=1)}{p(z_j|v_i=0)}, \quad (1)$$

where v_i denotes the occupancy probability of i^{th} voxel in the grid F_t^G, z_j is the measurement event that j^{th} camera ray passes through this voxel. The numerator of the item $\log \frac{p(z_j|v_i=1)}{p(z_j|v_i=0)}$ means that the probability of being passed for a voxel if this voxel is occupied in fact, which shows the confidence of ray casting process. Obviously, the numerator and denominator can be set as an empirical constant. Therefore, we update the log-odds occupancy probability of each voxel in the grid F_t^G by adding a constant one time when a single camera ray passes through this voxel. Note that the probabilistic occupancy grid F^G is continually updated within one episode. Finally, the occupancy status of voxels can be classified into three categories: unknown, occupied, and free, by setting a probability threshold.

We use a constant C' to represent the value of a ratio $|\frac{\log p(z_j|v_i=1)}{\log p(z_j|v_i=0)}|$ in Eq. (1), where the denominator term $|\log p(z_j|v_i=0)|$ is set to 0.01. To demonstrate the robustness and optimality of hyper-parameter in our implementation, we show the experimental results in Tab. 1.

Table 1. Evaluation results for different empirical parameters in the implementation of occupancy grid mapping algorithm. Note that we use a constant C' to represent the value of a ratio $|\log \frac{p(z_j|v_i=1)}{p(z_j|v_i=0)}|$ in Eq. (1)

.

Occupancy Threshold	Mean AUC ↑	Final Coverage Ratio ↑	Accuracy ↓
0.5 (Ours)	**87.39%**	**95.63%**	**0.37**
1.0	84.41%	93.43%	0.41
1.5	84.84%	95.20%	0.39
2.5	54.77%	58.70%	0.85
Value of Constant C'	**Mean AUC ↑**	**Final Coverage Ratio ↑**	**Accuracy ↓**
5	66.96%	71.92%	0.49
10	80.75%	92.16%	0.43
20 (Ours)	**87.39%**	**95.63%**	**0.37**
40	86.97%	94.12%	0.40

1.2. Network and Training

Training. In Isaac Gym [7], we set 256 parallel environments for policy training, where each environment corresponds with one building-level object. Once the coverage ratio reaches a threshold (99% in practice), a collision happens or the episode length reaches the maximum threshold (100 steps), the environment is reset and the building to be reconstructed is replaced. Our NBV policy is optimized through over 32 million iterations and uses approximately 24 hours of training time on an NVIDIA V100 GPU. All networks are randomly initialized and trained end-to-end.

Network. We propose a multi-source representation that has better generalizability. In particular, we first build two mid-level representations: a 3D geometric representation F^G from depth maps and a semantic representation F^S from RGB images. Geometric representation F^G encoded from depth images is with shape of $(N, X, Y, Z, 4)$, where the number of parallel training environments N is 256, grid size (X, Y, Z) is $(20, 20, 20)$ and last dimension represents the 3D world coordinate and occupancy possibility. Semantic representation F_S is stacked grayscale images encoded from multi-view RGB images with the shape of (N, M, H, W), where the number of RGB images M is 5, the size of grayscale images is $(64, 64)$

Specifically, we encode mid-level representations to embeddings: $s_t^S = f^S(F_t^S)$ and $s_t^G = f^G(F_t^G)$. f^S encompasses a 2-layer 2D convolution and $\mathrm{Linear}(\mathrm{Flatten}(x))$ operation, while f^G encompasses a 2-layer 3D convolution and $\mathrm{Linear}(\mathrm{Flatten}(x))$. Subsequently, we combine them with the historical action embedding $s_t^A = \mathrm{Linear}(a_{1:t})$ to generate the final state embedding s_t, as the input to the policy network. This process can be formulated as:

$$s_t = \mathrm{Linear}(s_t^G; s_t^S; s_t^A), \quad (2)$$

where all embeddings are 256 vectors.

Our policy network PPO, implemented by Stable Baseline3 [10], is a 3-layer multi-layer perceptron (MLP). The output of our policy is used to parameterize a distribution over our 5-dimension action space. In this way, the action can be drawn from the stochastic policy $a \sim \pi(\cdot|o_t)$.

1.3. Baseline Policies

The implementation details of baseline policies are described below:

1) **Random Policy**: This policy randomly generates 5-dim vector $(x, y, z, pitch, yaw)$ among the action space as the

next action. The randomly generated positions are constrained so as not to cause collisions. The reported results are evaluated on the test set and averaged over random seeds from 0 to 4.

2) **Random Policy on the Sphere**: This policy randomly generates positions (x, y, z) on a pre-defined hemisphere that exactly covers all objects of the test set. The headings are required to point to the center of the hemisphere. To avoid collisions, we set the radius of the hemisphere to 9 meters, which is greater than the maximum height of the test set object. The reported results are evaluated on the test set and averaged over random seeds from 0 to 4.

3) **Uniform Policy on the Sphere**: All positions are evenly distributed on the previously mentioned hemispheres. Specifically, for Houses3K test set, all sampling points are distributed over 5 heights, each with 6 evenly spaced positions. For OmniObject3D, all sampling points are distributed over 4 heights, each with 5 evenly spaced positions.

4) **Uncertainty-Guided** [4]: This NBV policy iteratively selects the next view from a pre-defined viewpoint set based on the entropy-based uncertainty based on a continually optimized neural radiance field. We use TensoRF [1] as the implementation foundation of neural radiance field. Before implementing uncertainty-driven viewpoint selection, we uniformly sample 100 views on the pre-defined hemisphere as the viewpoint candidate set. Note that we need to sample the candidate set for each object.

5) **ActiveRMap** [14]: The working pipeline is similar to uncertainty-guided policy. In particular, we implement the "discrete (free)" setup of ActiveRMap, which constrains the drone agent on the pre-defined hemisphere. Having considered collision avoidance in the design of viewpoint set, we remove the collision penalty in its optimization objective.

6) **Scan-RL** [9]: This RL-based policy predicts the next viewpoint from a 3-dim hemisphere space, relying on the historical RGB.

7) **Ours with Scan-RL's Representation**: To further compare the former Next-Best-View policy Scan-RL [9] with us, we implement Scan-RL in our experimental setup, with our action and state space. This free-space NBV policy uses Scan-RL's representation only extracted from RGB images instead of our original hybrid representations.

2. Data Preprocessing

3D Mesh. To boost our NBV policy's generalizability on building-scale objects, we rescale the original 3D meshes from Houses3K [8], OmniObject3D [12] and Objaverse [3] to a reasonable building size. For example, the size of building meshes from Houses3K and OmniObject3D are approximately $(15m, 15m, 8m)$.

Ground-truth Point Cloud. We use the Poisson Disk sampling method [13] implemented by Open3D [15] to sample $100,000$ points from 3D meshes. And then we voxelize these point clouds. These voxelized points are viewed as ground-truth point clouds of these meshes.

3. Additional Experiments

Number of Training Objects. Motivated by generalizable RL-based policies [2, 5, 6], we explore the impact of diversity of training data for generalizability. As shown in Table. 2 and Fig. 1, we found that increasing the diversity of training objects indeed leads to better generalizability for 3D reconstruction.

Table 2. Ablation studies of the number of training objects in our framework on unseen OmniObject3D house category.

Number of Training Objects	Mean AUC	Final Coverage Ratio
1	61.76%	67.32%
2	71.03%	82.98%
4	70.72%	89.53%
8	73.71%	90.47%
16	73.73%	88.23%
32	82.39%	92.41%
64	82.09%	93.70%
128	87.21%	96.75%
256	**88.63%**	**97.12%**

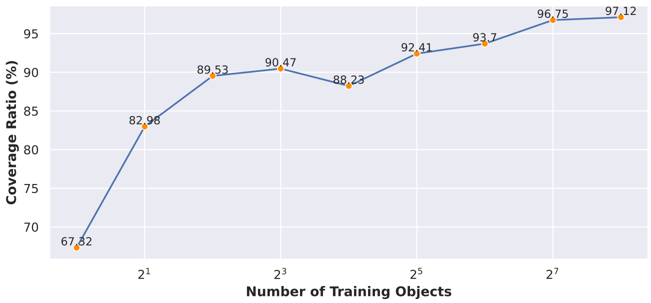
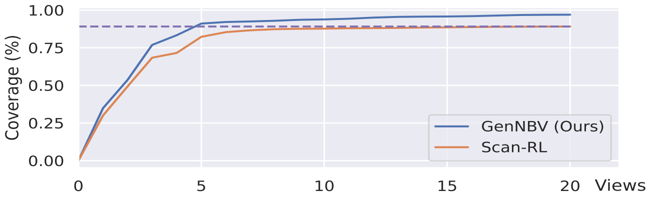


Figure 1. The curve of coverage ratio with the increasing number of training objects on unseen OmniObject3D house category.

The evidence for sufficient and efficient capturing. To evaluate both completeness and efficiency, we use the area under the curve (AUC) of coverage ratio as our main metric, stated in Sec. 4.1. Additionally, the figure below shows the mean AUC curves for OmniObject3D houses. Our GenNBV achieves a better coverage ratio under 20 views (97.12% v.s. 92.53%), and even surpasses the saturated coverage of Scan-RL using merely 5 views.



The effectiveness of 2D semantic representations. Below, we add the experiments on Houses3K test set to illustrate the effectiveness of the proposed semantic representation. Note that Scan-RL's representation uses 6 frames, while our semantic representation only needs 2 frames.

Num. of RGB	NBV Policy	AUC ↑	FCR ↑	Acc. ↓
2	Ours w/ Scan-RL Repre.	76.31%	79.35%	0.50
	Ours (RGB-only)	**81.24%**	**87.90%**	**0.45**
6	Ours w/ Scan-RL Repre.	87.39%	95.63%	**0.38**
	Ours (RGB-only)	**87.95%**	**96.92%**	**0.38**

In addition, we preprocess the RGB images to grayscale images because we empirically find that the grayscale images featuring edges are sufficient to guide the NBV prediction and achieve slightly better performance.

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 2

[2] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020. 2

[3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2

[4] Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077, 2022. 2

[5] Quanyi Li, Zhenghao Peng, Lan Feng, Chenda Duan, Wenjie Mo, Bolei Zhou, et al. ScenarioNet: Open-source platform for large-scale traffic scenario simulation and modeling. In *Advances in Neural Information Processing Systems*, 2022. 2

[6] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022. 2

[7] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance gpu-based physics simulation for robot learning, 2021. 1

[8] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Houses3K dataset. `https://github.com/darylperalta/Houses3K`, 2020. 2

[9] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 558–573. Springer, 2020. 2

[10] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1): 12348–12355, 2021. 1

[11] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 1

[12] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 2

[13] Cem Yuksel. Sample elimination for generating poisson disk sample sets. *Computer Graphics Forum*, 34(2):25–32, 2015. 2

[14] Huangying Zhan, Jiyang Zheng, Yi Xu, Ian Reid, and Hamid Rezatofighi. ActiveRMAP: Radiance field for active mapping and planning. *arXiv preprint arXiv:2211.12656*, 2022. 2

[15] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 2